ANALYSING USABILITY AND SECURITY ISSUES IN DESIGN AND DEVELOPMENT OF INFORMATION SYSTEMS

JOHNNES EBOT-ARREYMBI

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS OF THE UNIVERSITY OF GREENWICH FOR THE DEGREE OF MASTER OF PHILOSOPHY

February 2012

DECLARATION

I certify that this work has not been accepted in substance for any degree, and is not concurrently being submitted for any degree other than that of Doctor of Philosophy being studied at the University of Greenwich. I also declare that this work is the result of my own investigations except where otherwise identified by references and that I have not plagiarised the work of others".

J. Arreymbi

*Signature _____ Date: __25/07/11_*

Supervisor Signature ---------- Date: ----------

ACKNOWLEDGEMENT

I extend my sincere gratitude and appreciation to the many people who made this research possible. Special thanks to my supervisors Professor Liz Bacon, Professor Lachlan MacKinnon and Professor Mohammad Dastbaz for their constant support and dedication throughout my research, all to whom, I am highly indebted.

I am greatly indebted to my wife and kids for their patience, understanding and enduring support during the course of my hectic research work and the many days and nights away from them; may God grant us all our wishes abundantly.

I will also like to acknowledge with much appreciation, the many people who have helped, supported and encouraged me directly or indirectly throughout the course of this research. And not forgetting the crucial roles played by many organisations, friends, and colleagues in making this work possible; especially, Royal Academy of Engineering, KTN, and SITC.

ABSTRACT

Recent technological advancements and the global economic challenges have meant that, individuals and businesses are constantly seeking new ways to exploit Information Systems (IS) and in manners that not only enhance user experiences and/or improve business processes and productivity, but also protect the individual's privacy and business assets for competitive advantage. Therefore, Information Systems need to be designed and developed to meet these challenges and/or other objectives. This thesis will delve primarily into the history of IS as a basis for establishing where the problem(s) lie or emanate from. It will focus on critically analysing existing Information Systems, and investigating the conflicting issues of usability and security, from an Information Systems Design and Development perspective by analysing various approaches. An in-depth review of literature and critical analysis of requirements necessary for the design and development of a usable and secure Information System will be carried out and will form the intellectual framework for this research. The premise therefore, is to look for a balanced approach or appropriate trade-off framework for designing usable-secure systems. The research will conclude with a discussion on how an envisaged conceptual framework or model can be developed based on certain influential factors, and how the framework can be experimentally evaluated, and to suggest areas for further improvement or future research.

DEDICATION

This work is dedicated to my wife and kids who have endured so much with me during this period, and to my mother, who has always been there for the family but has recently suffered and is recovering from Stroke in the last couple of years.

'Knowledge allows you to *plan, planning helps you* to *prepare, preparation gives you the competitive edge and the edge well, grants you* a *better chance* of *success.'*

J. Arreymbi

Table of Content

List of Figures

List of Tables

Acronyms

ABA	American Bar Association
AEGIS	Appropriate and Effective Guidance for Information Security
APS	Alternate Power Source
AR	Assigned/Active Role
ATM	Automatic Teller Machine
B-2-B	Business to Business
B-2-C	Business to Customers
BBC	British Broadcasting Corporation
BIM	Biba Integrity Model
BLP	Bell-La-Padula Model
BPR	Business Process Reengineering
BRS	Business Requirement Specification
BRS	Business Request Specification
BSO	Business System Options
CATWOE	Customer, Actors, Transformation, Weltamschauung, Owners, Environment
CDI	Constrained Data Item
CDMA	Code Division Multiple Access is a 2G technology developed by Qualcomm that is transitioning to 3G
CIA	Confidentiality, Integrity and Availability
CLD	Composite Logical Design
CNSS	Committee on National Security Systems
COBIT	Control Objectives for Information and Related Technologies,
CoU	Context of Use
CS	Computer Science

CSA	Cognisant Security Agency
CWIM	Clark- Wilson Integrity Model
CWM	Chinese Wall Model
DFD	Data Flow Diagram
Doc	Security Documentation
DSDM	Dynamic System Development Methodology
DSS	Decision Support System
EDGE	Enhanced Data GSM Environment is a 3G digital network
EDI	Electronic Data Interchange
EIS	Executive Information System
Epos	Electronic Point of Sale
ERD	Entity Relational diagram
ETHICS	Effective Technical and Human Implementation of Computer-based Systems
GPRS	General Packet Radio Service is a 2.5G network that supports data packets
GSM	Global System for Mobiles
GSM	Global System for Mobile Communications is a 2G digital cell phone technology
GUI	Graphical User Interface
HAS	Human Activity Systems
HCI	Human Computer Interaction
HSBC	Hong Kong & Shanghai Banking Corporation
HTML	Hypertext Markup Language
I & A	Identification and Authentication
ICA	Internal Control Agent
ICO	Information Commissioners' Office

ICO	Information Communication Officer
ID	Identity
IEEE	Institute of Electrical and Electronic Engineers
IS	Information System
ISAC	Information Systems Work and Analysis of Changes
ISD	Instructional Systems Design
ISDM	Information System Development Methodology
ISO	International Standards Organisation
ISS	Intangible Security Scenario
ISSM	Information System Security Manager
ISSO	Information System Security Officer
IT	Information Technology
IVP	Integrity Verification Procedure
JAD	Joint Application Development
KAD	Knowledge Analysis and Documentation System (or Knowledge Analysis and Design Support)
LAN	Local Area Network
LDM	Logical Data Model
LDM	Logical Data Method
LDS	Logical Data Structuring
LFD	Logical flow diagram
LMD	Logical Modelling Diagram
LMD	Logical Method Diagram
MAC	Mandatory Access Control
MIS	Management Information system
MLS	Multilevel Security

MoSCow	Must have, Should have, Could have, Won't have
Multiview	An approach to systems analysis and design and to the evaluation of potential solutions to information processing problems
n.d	Not dated (no date)
NCC	National Computing centre
NHS	National Health Service
NIMSAD	Normative Information Model-based Systems Analysis and Design
NIST	National Institute of Standards & Technology
NSA	National Security Agency
NSTISSC	National Security Telecommunications and Information Systems Security Committee
OAS	Office Automation Systems
OMG	Object MGT Group
OOA	Object Oriented Analysis Approach
OOAD	Object Oriented Analysis and Design
OOADM	Object Oriented Analysis and Design Method
OOD	Object Oriented Design
OOP	Object Oriented Programming
OpCit	Optional Citation
P	Permission
PA	Permission Assignment
PC	Personal Computer
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PL	Protection Level
PRINCE2	Project in Constraint Environment 2
QA	Quality Assurance

R	Role
RA	Authorised Role
RAD	Rapid Application Development
RBAC	Role Based Access Control
RescrcCtrl	Resource Control
RIT	Partially ordered role hierarchy
ROCE	Return on Capital Employed
ROI	Return on Investment
S	Subject
SA	Subject Assignment
SATNAV	Satellite Navigation
SCS	Security Control System
SDLC	Software Development Life Cycle
SDM	Software Design Methodology
SE	Software Engineering
SessCtrl	Session Control
SET	Social, Economic and Technical
SMS	Short Message Service
SOA	Service Oriented Architecture
SOD	Segregation of Duties
SQL	Structured Query Language
SQUARE	Security Quality Requirements Engineering
SR	System Recovery
SRS	Software Requirement Specification
SSADM	Structured systems analysis and design method
SSM	Soft System Methodology

SSP	System Security Plan
STRADIS	STRuctured Analysis, Design and Implementation of information Systems.
SUS	Security Usability Symmetry
SysAssur	System Assurance
TA	Transaction Authorised for Role
TDMA	Time Division Multiple Access is 2G technology
TL	Transaction Log
TP	Transformation Procedure
TPS	Transaction Processing System
TSO	Technical System Option
TSS	Tangible Security Scenario
UCCR	UK Cyber Crime Report
UCD	User Centred Design
UDI	Unconstrained Data Items
UELC	Usability Engineering Life Cycle
UI	User Interface
UK	United Kingdom
UML	Unified Modelling Language
UMTS	Universal Mobile Telephone Service
UPS	Uninterruptible Power Supply
User ID	User Identity
VPN	Virtual Private Network
WAN	Wide area Network
WWW	World Wide Web
YSM	Yourdon Structured method

Glossary

Acceptability	Worthy of being accepted; adequate to satisfy a need, requirement or standard satisfactorily.
Access	Ability, permission to approach, right, to enter, use, admittance granted or Access Control –to and from a system; The ability and opportunity to gain knowledge of classified information.
Accessibility	Ease of access, convenience, case of misunderstanding, openness
Adaptability	Compliance/ Malleability, ability to change or adjust to changing situations
Agile Method	A software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. It is a conceptual framework that promotes foreseen interactions throughout the development cycle.
Antivirus	Virus protection in a Software; is protective software designed to defend your computer against malicious software. Malicious software or "malware" includes: viruses, Trojans, key loggers, hijackers, diallers, and other code that vandalizes or steals your computer contents.
Asset	Anything of tangible or intangible that is of value
Attack	To set upon in a forceful, hostile, or aggressive way with or without a weapon; to work on with purpose and vigour
Attractiveness	Capability of being liked, has an appeal to it and is interesting
Authentication	The process of identifying an individual usually based on a username and password. Authentication merely ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual.
Authorisation	In security systems, authorization is the process of giving individuals access to system objects based on their identity.
Availability	Enables timely and reliable access to information and use of information; a loss of availability is the disruption of access to or use of the information.

Backup	Activity of copying files/database to be preserved in case of system failure. Backup - restoration of data
Bluetooth	Bluetooth is a telecommunications industry specification that describes how mobile phones, computers, and personal digital assistants (PDA) can be easily interconnected using a short-range wireless connection.
Cognitive aspects	Emphasises not merely how individuals receive material to be learned and "construct" it inside their heads, but how they and the system designers construct it between them through their dialogue
Concept	An aspect of thought; a general idea derived or inferred from specific instances or occurrences
Conceptual Model	A conceptual model represents 'concepts' (entities) and relationships between them
Confidentiality	Confidentiality is when disclosure or exposure to unauthorised individuals or systems is prevented; Confidentiality enables users with rights and privileges to have access to information.
Control	Exercise restraint or direction over; to exercise dominating or authoritative influence over or adjust to requirements.
Countermeasure	A measure or action taken to offset another one.
Cybercrime	Offences that are committed against individuals or groups of individuals with a criminal motive to intentionally harm the reputation of the victim or cause physical or mental harm to the victim directly or indirectly, using modern telecommunication networks such as Internet (Chat rooms, emails, notice boards and groups) and mobile phones (SMS/MMS)
Data Transmission	Transfer of digital data over a communication (point-to-point or point-to-multipoint) channel
Design	A strategic approach or roadmap to achieve a unique expectation, clearly defining the specifications, plans, and parameters etc. Of the system
e- Commerce	Electronic commerce
Exploit	An act or deed, or using something to one's advantage; taking advantage of a security vulnerability
Exposure	An act or instance of subjecting, influencing or leaving open to external factors or risk, exposing.

Feasibility	The determination as to whether the assigned tasks could be accomplished by using available resources; Capable of being accomplished or brought about; possible; Viability or probability
Firewall	A defence mechanism to protect against intrusion attacks
Flexibility	Elasticity/ Suppleness, easily changeable
Framework	A set of assumptions, concepts, values, and practices that constitutes a way of viewing reality; A hypothetical description of a complex entity or process
Fraud	Unauthorised use or change of System data
Functionability	The quality or state of being functional; especially the set of functions or capabilities associated with software or hardware
Hacking	To gain unauthorised access to a system, either intentionally, forcefully or otherwise
Hard Aspects	Rigid, tough/ inflexible aspects
Holistic approach	General or full view or encompassing approach
Human Factors	Ergonomics (or human factors) is the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data and methods to design in order to optimize human well-being and overall system performance.
Identity Fraud	Identity Fraud is a crime in which an imposter obtains key pieces of personal information, such as Social Security or driver's license numbers etc, in order to impersonate the victim and commit fraud in the process. The information can be used to obtain credit, merchandise, and services in the name of the victim, or to provide the thief with false credentials.
Impersonation	Pretending to be somebody else
Incremental approach	Stage by stage additions
Information Sensitivity Matrix	A matrix that shows the various levels of security concerns and the information sensitivity qualifiers
Information Systems (IS)	Information system (IS) is an integrated set of components for collecting, storing and processing data; for delivering information, knowledge and digital products. It is any combination of IT and

	people's activities that support operations, management and decision making.
Information System Development Methodology (ISDM)	Information system development methodology (ISDM) defines a process to be used in the development and deployment of information and other systems. ISDM defines a controlled project management process which moves from early identification of a business problem or opportunity through to its delivery into operation; it also covers the processes for maintaining the solution in operation and its final removal.
Infrared	Relating to the range of invisible radiation wavelengths from about 750 manometers, just longer than red in the visible spectrum, to 1 millimetre, on the border of the microwave region.
Integrity	Enables a system to perform intended function free from deliberate or unauthorized manipulation of the system
Interaction	A mutual or reciprocal action; ways in which elementary particles and bodies can influence each other.
Iterative Approach	A technique of developing and delivering incremental components of business functionality, product development or process design.
JAD Methodology	JAD (Joint Application Development) is a methodology that involves the client or end user in the design and development of an application, through a succession of collaborative workshops called JAD sessions
Learnability	Ability to learn/ operation control or effort for operations
Logon	Gaining Access to a System Log
Maintainability	The ability of an item, under stated conditions of use, to be retained in or restored to a state in which it can perform its required functions, when maintenance is performed under stated conditions and using prescribed procedures and resources
Memorability	The quality of being worth remembering or noted. Can be measured by how easy it is to remember something after a substantial time-lapse between visits.
Method	Process or Scheme, procedure or technique or way of doing something especially in line with a plan
Methodology	Study of methods; a body of practices, procedures and rules used by those who work in a discipline

Model	A preliminary work or construction that serves as a plan from which a final product is to be made; A schematic description of a system, theory, or phenomenon that accounts for its known or inferred properties and may be used for further study of its characteristics
Multimedia	A seamless integration of various media (audio, text, video, graphics, etc.) using a computer
Object	A tangible, visible, self-contained and reusable entity
Operability	Being such that use or operation is possible; Possible to put into practice; practicable; Operation control/ effort for operation
Pragmatic approach	Practical, realistic approach to performing a task
Protection Profile	Documentation used as part of a certification process
Prototyping	Prototyping is the process of building a model of a system.
Psychological aspects	The psychological dimension of the making and messages of a system can provide opportunities for release of emotions, perception, expression, cognition, motivation and definition of self.
Reductions approach	Downsizing, minimalist approach
Reliability	The ability of a system or component to perform its required functions under stated conditions for a specified period of time
Requirement	The characteristics or features of a desired system, what the intended system must do and related to identified business needs
Research	Scholarly or scientific investigation or enquiry. A detailed study of a subject in order to discover (new) information or reach a (new) understanding.
Rich Picture	Part of the Soft systems methodology, Rich Pictures provide a mechanism for learning about complex or ill-defined problems by drawing detailed ("rich") representations of them.
Risk	The possibility of suffering harm, loss or danger; an element of uncertainty.
Root definition	A Root Definition is a structured description of a system. It is a clear statement of activities which take place (or might take place)

	in the organisation being studied.
Safeguard	Protection against the unknown, a device designed to defend or prevent accidents; measures to ensure safety
Satisfaction	Contentment with the system, fit for purpose and happy with use of the system
Security	Freedom from risk, danger or safety; Safety measures, protection or defence approach, or precautions or control mechanisms against any danger to the system
Security Blueprint	Operations security planning guidance or documentation of security objectives; overall security plan to ensure the proper operations of the system
Security Model	A scheme for specifying and enforcing security policies; founded on a formal model of access rights.
Security Posture	Aspects of business approach to security; level of assurance that adequate technical security controls have been implemented to meet the information protection needs.
Security Profile	Authentication protocol that is used as part of a certification process. Giving access to an account or area depending on the users profile
Soft Aspects	Aspects which involve psychological, social, and cultural elements; Flexible, elastic/ fluid aspects
Stakeholder	A person, group, organization, member or system, with an interest in a project, and who affects or can be affected by an organization's actions
System Analyst	Person responsible for the analysis, design, development and modification of an information
Test	A means of trial, the means by which the quality, genuineness, presence of anything is determined. Security Testing – testing of system features for functionality
Threat Agents	Method used in breaching the security of a system, facility, operation by exploiting a vulnerability
Threats	An expression of an intention to do harm, inflict pain, injury, punishment or impending danger to a system
Time-Box	Time management technique, a time box allots a fixed period of time for an activity; Constraints of time/ Set time periods/dates

	with little or no flexibility
Understandability	Recognising logical concept; the quality of comprehensible language or thought; ability to grasp meaning and understand or follow a process or instructions
Usability	The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use; Usability means making products and systems easier to use, and matching them more closely to user needs and requirements.
User	Anybody who uses a system
Validation	Testing for compliance; confirming by examination and provision of objective evidence that the specifications conforms to standards or user needs etc; to establish soundness and/or make the system legally valid.
V-model	A software development process with a V-shaped graphical representation of the systems development lifecycle, describing the activities to be performed and the results that has to be produced during the product development.
Vulnerability	A flaw or weakness in the system security procedures, design, implementation or internal controls that could be exploited and lead to a security breach or violation of the system's security policy
Vulnerability	Weakness/ susceptibility in a system
Waterfall Model	A sequential design process often used in software development processes, in which progress is seen as flowing steadily downwards through the phases.

Chapter 1 Introduction to the Thesis

1.0 Introduction

This chapter provide the background and basis for the research. It sets out the research questions, aims and objectives, including the methodology and rationale, and indicates how the report will be structured.

1.1 Background and Raison d'être

Increasingly nowadays, users and businesses, including Government authorities have tended to demand more from designers, developers and vendors of information systems; with regards to access, ease of use and better protection of their vital information and the systems infrastructures. Therefore, the need to conform to these increasingly stringent usability and security demands in Information Systems (IS), together with quality standards in communication means that, the more traditional strategies for developing and dealing with these systems (i.e. storing, accessing, manipulation and transmission of information processes) are no longer acceptable; due partly to advancement, proliferation and ubiquitous nature of the technologies. Also, (surprisingly) it is down due to the user techno-savvy attitudes, the systems' needs, and the very dynamic environments in which they operate. All these issues have compounded the problems of the technologies. Users are frequently put-off operating technologies that either have difficult (not-user-friendly) interfaces or have many layers of security protocols. Therefore, what is probably needed is an adaptive technology, that is based on users experience and providing adaptive and/or contextual solutions to users problems. For example, the security and usability of information systems have always posed conflicting challenges that need addressing right from the systems design and development; even more especially in this age of digitalisation, convenience, information superhighway and ubiquitous computing. The complex and demanding nature of information systems, coupled with the insatiable nature of users and computer power, all these have made the tasks more daunting than ever before. It is well known fact that, as in all systems, with increase usability, security is reduced or compromised and vice versa. Hence, the question arises; how do we solve the issues of trade-off or balancing security and usability in IS? And how do we make this conflicting but necessary marriage of technologies, that of convenience rather than inconvenience?

Historically, IS and business operate within a broader economic and social context. Many analysts have suggested that information systems (IS) are social systems or at least systems with social implications, with an extensive use of technology (Benson and Standing, 2002). However, most IS have been designed and developed as separate entities with very little user consideration and/or involvement in the approaches; and more especially in relation to the combine issues of usability and security requirements. Most often, in systems design and development, these issues are tackled in isolation and/or in varied proportions/ratios; either because of the organisation / business requirements or environmental demands that serves a particular purpose or has a different objectives. It is very rare to see a good balance of usability and security in IS designs, because the issues (usability and security) have very diverse and conflicting interests at any point in time; therefore always a trade-off between them. The trade-off has arisen because of the fact that, whenever very stringent security protocol are in place within any systems, users for one reason or the other find it very difficult to use the system and in some circumstances may circumvent the security or abandon it altogether. Likewise, if the system is made very user friendly, and easy, there is a high chance that the security protocols are weak or non-existence and therefore very vulnerable to attack by unauthorised persons. So therefore, we must look for better ways of aligning security and usability seamlessly. That is, adequately protecting the system but still making it usable or user friendly - that is usable-security (Dewitt and Kuljis, 2006, Sasse, 2011).

According to Sasse, (2011, pp 6), to achieve this goal, systems developers and security community must look at systems security from a human-centred perspective; designed to fit human capabilities and limitations, without generating unreasonable demands and workloads. Such concerns have meant that, increased efforts have been aimed at providing better quality interactive interfaces, reducing the length of process time on the one hand, and on to more efficient means of providing security, quality and speed on the other hand. For example, technologies today have provided digital multimedia information - power cycles – sound, text, animation, imagery, sequence, pattern and paradigm that have facilitated and improve usability and security (interactivity, processing speed, clarity, compression, encryption techniques and quality in transmission, communications and availability etc.), with an efficiency that far outweigh the conventional analogue cycles. Such improvements have brought about the emergence and proliferation of digitalised electronic information systems, most of them internet enabled (for example an e-Commerce system), which has been

estimated to grow annually by 10% to the tune of approx. $250 billion by 2014 in US alone; up from $155 billion in 2009 (Forrester Research, 2010). The estimate for online and web-influenced transactions in the US will top almost $1,409 billion by 2014. Forrester Research Inc. forecast the Worldwide e-Commerce growth, for both B-2-B and B-2-C transactions online to be approximately $24 trillion by 2014 (Forrester Research, 2010). Therefore, with such enormous growth and financial benefits; and the potential associated risk to both individuals and businesses, it is very important that such vital systems should be made very user friendly to encourage more users, and also securely protected to maintain integrity, privacy and availability of the information for business continuity.

Meanwhile, advances in the design, development, proliferation and use of internet-enabled mobile and wireless communication technologies (web browser, GSM, GPRS, EDGE, CDMA, TDMA, UMTS, Bluetooth, infrared, multimedia and compression techniques etc. on SmartPhones, iPads, kindles notepads, iPhones and Android systems etc.) have taken the issues of usability and security to another level in IS, which requires prompt action. However, creating functional and operational environments for such mobile systems is by no means a straightforward task. Although some of the mobile devices features have improved recently, there still are very interesting inherent design usability and security problems. For instance and according to Arreymbi and Dastbaz (2002), the memory size, battery life, software compatibility, processing power, protocols, and always-on connectivity etc. of these systems, means that normal design approaches need to be re-examined in order to come up with appropriate design model(s), that provide for adequate security and usability of the information systems. But first we must try to understand what information systems are and the problems associated with them?

1.2 Rationale

Information System design and development have always posed interesting challenges, especially when it comes to issues of usability and security. Potentially, it is due to the rapid advances in technologies, the requirements for manipulating the systems and the increasingly techno-savvy and demanding nature of users. In fact, there is always conflicting interest between usability and security in any system. For example, if a system is designed to be usable and fit for purpose for the user; security then becomes an issue and vice versa; one is very often always compromised for the other. In the last five years, there have been several

security and usability flaws in the development and use of information systems, and which have led to many breaches and/or misuse of systems. For example, in 2010, the HSBC banking system was compromised, due to poor security implementation and data was lost. "HSBC admits huge Swiss bank data theft: about 24,000 clients of HSBC's private banking operation in Switzerland had personal details stolen by a former employee, the company has admitted." (BBC News, 11 March, 2010). Also, the Heathrow passenger biometric and x-ray systems recently raised many users concerns about usability and privacy/security of the systems. Again, there were many usability and security concerns during the disastrous and dreadful launch of the Heathrow Terminal 5 passenger check-in and baggage handling systems failures. Other examples include; "Call centre 'scam' details sought: India's IT industry has urged Britain's Channel 4 television to co-operate with the authorities after a sting alleging data theft from Indian call centres." (BBC News, 4 October 2006). "Unacceptable level of data loss: The number of incidents of loss or theft of personal data has risen to an "unacceptable" level in recent times, the privacy watchdog has warned. The Information Commissioner's Office (ICO) said NHS hospitals holding private medical records were among the worst offenders. In total, 434 organisations reported data security breaches in the last 12 months of 2009, up from 277 the year before." (BBC, 11 November, 2009).

All these and many others have brought to attention the challenges and the many questions being asked about, how design and implementation of the information systems have affected many people (in terms of use and protection), and businesses (in terms of business processes and business continuity). Therefore, in the light of this, how do we design and implement systems appropriately to address the balance or make a sensible compromise that will make information systems user friendly enough and also provide for adequate security? The decision for any such IS project will most probably be based on the likely costs and potential benefits to the organisation. According to Boddy et al., (2009), the costs and benefits of IS projects are notoriously difficult to determine.; and which is why in most recent cases for example, the Libra projects to provide a UK national courts system and the London Borough of Haringey 2003 'Tech Refresh' project. The costs in both cases more than doubled, going from an initial £146m to £390m and from £9m to £24m respectively; and the benefits have either not been realised as in the former, or as in the later, difficult to predict/quantify. The costs in most instances can be summarised to include; costs of acquiring the technology (hardware and software), costs of implementation, ownership, change and infrastructure;

associated impact on stakeholders and other long term costs implications. The benefits can equally be summarised to include; tangibles (direct cost savings, quality improvements, increase productivity and revenue, business survival) and intangibles (staff morale, communication, customer satisfaction and management, reputation, reduced downtime, value chain management, flexibility, etc.) (Boddy et al., 2009).

This research will examine these issues in the light of IS and ISDM and look at better ways of addressing the problems of usability and security. In presenting this challenge, a model/framework on IS development will be proposed, and it is hoped that will potentially resolve the issue or bring about a balance approach in handling the usability and security problems of IS to provide for a system that is usable and secure (secure usability or usable security).

1.3 Research Questions

The underlying factors in making information systems (IS) effective in today's environment are to make them a) usable and b) secure. Therefore, the ultimate desire or challenge for Information Systems (IS) designers is to find a balance or appropriate trade-off between these two very necessarily desirable but conflicting issues - usability and security - in IS design and development. The question therefore:

Is there currently a methodology or framework that successfully addresses the problems of security and usability?

In an attempt to resolve the above issues, other concerns come to light such as:

1. How do design and development practitioners achieve a 'balance,' (compromise) in an attempt to make IS usable and secure?
2. What is required to make this issue less of a trade-off? (i.e. not sacrificing one for the other)
3. How can designers make sure that, users find system security not something that is daunting and/or hindering them from performing or achieving set goals?
4. What are the necessary requirements to design and develop a usable and secure system?
5. Will a proposed new framework or model be adequate enough to contribute in solving the problems of usability and security in IS design and development?

If this research achieves the methodology that addresses these issues, then it can best seek the method to design and develop or set out criteria for such a system.

In attempting to address the questions, this research will delve primarily the history of Information Systems (IS) as a basis for establishing what the problem(s) are; where they emanate from; and how they can be resolved. It will focus on looking at what constitute an information system, then critically analysing existing information systems design and development methodologies in relation to how they are used to address the issues of security and usability; and to determine better ways of approaching the current escalating issues. Also, and to highlight the issues, we will critically analyse two approaches such as Appropriate and Effective Guidance for Information Security (AEGIS) (Flechais et al., 2007) and Security Quality Requirements Engineering (SQUARE) (Mead et al., 2008) which purport to address the issues of security and usability. Then, we propose a model in an attempt to overcome the current information systems difficulties as already highlighted. Finally, the research will summarise in-depth the issues discussed and then draw on some conclusions and make recommendations for future research.

1.4 Aims and Objectives

1.4.1 Aims of the Research

This research aims to investigate the usability and security issues in the design and development of Information Systems (IS).

1.4.2 Research Objectives

The objectives here are to:

- Review historical background of information systems as a premise for design
- Critically analyse some existing IS design and development approaches to see how they address or handles issues of usability and security.
- Identify the canonical set of issues that can form the basis for design of a potential framework or model for developing a secure and usable IS
- Critically analyse existing IS design approaches and identify the best candidate approach in dealing with security and usability.

- Draw some conclusions on research findings and make recommendations for further work, also incorporating consideration on whether a novel approach or framework can contribute to solving the issues of usability and security in design and development of IS. This can be further achieved and/or supported through future works and impending experimental evaluation of the framework.

1.5 Research Methodology and Process

This research is carried out to investigate the usability and security concerns of information systems. The research methodology involves, according to Collis and Hussey (2003); Buzan and Buzan (2006) the use of analytical approach, mainly a qualitative analysis (non-empirical data), and review of literature on Information systems. In this Thesis we will examine the current body of knowledge to highlight the current issues pertaining to usability and security in the context of information system design and development. The overall approach to the research process - from the theoretical underpinning to the analysis of data - the research work involved methodological triangulation, and undertook a comprehensive review of all the literature sources used in the course of the Thesis. The aim of the literature search was to identify as many items of secondary data as possible which were relevant to the research such as books, articles in journals, magazines and newspapers, conference papers, reports, archives, published statistics and records. We critically analysed literature on existing IS development methodologies to identify the limitations and knowledge gap with regards to usability and security in IS development. Here we present the strengths and weaknesses of concepts and theories to help justify the purpose of the research, and to present the grounds of its contribution to knowledge. In the discussion, we analyse two research approaches to see how they dealt with the problems of usability and security and looked at areas for future research. Finally, and based on the critical evaluation of the literature and thorough examination of ISDMs, we proposed a novel approach or framework to address the knowledge gap in this area. The proposed framework is designed based on problem solving perspective of methodologies (Jayaratna (1994). According to Jayaratna (1994) and Akhgar (2003), models are embedded in methodologies and their role, type and form help to determine what aspect of reality are captured and understood; and its complexity increase as we learn more about the underlying problem domain.

The research therefore contains three interrelated parts: Part 1 is critical literature evaluation as supported by Orlikowski and Baroudi, (1991); and Zabriskie & Huellmantel, (1994), of information systems and IS development methodologies. It highlights the strengths, weaknesses and concepts, and the theoretical gaps presented as rationale for undertaking the research, and to improve things in future. An in-depth analysis and evaluation of the IS methodologies with regards to usability and security protocols will be undertaken to assess the problems faced. The second part analyses existing ISDMs, selected on the basis of their suitability in representing a particular IS development paradigm, considering the mechanisms and processes by which they address usability and security issues.

The third part presents a discussion of the approach used, together with some conclusions drawn and recommendations for future works, including a consideration of the canonical set of issues in relation to the design of information systems that are both secure and usable, which can form the basis for a future design of a novel ISDM.

1.6 Novelty of the work

The research will identify through intensive review and critical analyses of literature, new areas which highlights the challenges and limitations of designing usable and secure information systems. It will look at the barriers and other issues influencing systems failures, to determine ways of achieving a good design and development balance between usability and security in information systems. In proposing a framework, it is assumed this will form a body knowledge that will contribute to improvement in IS design and development.

1.7 Dissemination and outputs

It is anticipated that a broader, but related discussion of the issues explored in the thesis and the findings, from an academic perspective will be published through a number of refereed journals, conferences and/or public presentations.

1.8 Thesis Structure

Chapter 1 provides a brief introduction setting the scene for the thesis, setting out the aims and objectives, methodology used and rationale for the research.

Chapter 2 presents some general background that critically discusses Information Systems (IS) as an academic discipline from current body of knowledge and exploring IS in the individual and organisational contexts of security and usability to provide a conceptual understanding of the issues involved. It will look at existing IS methodologies and paradigms from problem solving perspective and highlighting their limitations in attempting to address the issues as raised. provides some analyses of a selection of methodologies which are representative of the paradigmatic type, and makes comparison of how they are used for information systems development; showing the strengths and weaknesses; and providing the rationale for the selection, in an attempt to address the usability and security issues, and how the knowledge gather can be used to addresses the current problems of IS.

Chapter 3 examines issues of usability and security in relation to information systems. It discusses more generally issues of the behaviour of human actors relative to security protocols and systems.

Chapter 4 looks at life cycle models (SDLC) and provides some analyses of them in relation to the issues raised. It draws from the analysis to build a critical consideration of the key factors required to be introduced at each stage of the lifecycle - analysis, design and development of IS, to support usability and security considerations within it, such as development lifecycle, user engagement and the security versus usability trade-offs.

Chapter 5 focuses on identifying the canonical set of issues for IS development in attempting to establish the trade-off between usability and security. It describes each stage of the lifecycle and highlights the issues that appear in the existing models and the features they provide to address usability and security aspects.

Chapter 6 provides a summary of the findings and analyses the research approach. Also, it draws some conclusions and makes recommendations as to the way forward, including areas for future work. This chapter will consider the value of existing ISDM approaches to security and usability, identifying the best fit from existing models, and will consider the use of the canonical set of issues as the basis for the design of a novel future ISDM. It also highlights limitations to the research and future work.

1.9 Summary

In this chapter, we have set out the premise of the research, by providing some historical background on IS and how it has changed the way we live and do business. Here we developed the research question(s) to highlight of the impending issues regarding IS and IS development. The aims and objectives, rationale and the research methodology have been set out to justify the raison d'être for carrying out the research. The end of this chapter provides the Thesis structure to indicate how and what the research will be about.

Chapter 2 Research Overview

2.0 Introduction

This chapter deals with the origins of information systems; what constitute IS, and the different disciplines that make up what we know to be information systems. Various IS definitions are given to clarify the underlying concept in design and development methodologies in use today.

2.1 Background to Information Systems

According to Buckland (1998), information system (IS) has its humble beginning from "Documentation". Meanwhile Benson and Standing (2002) are saying information systems have been around for at least 6000 years. However, the application of computing technology to information processing only occurred at the end of the 1950s and later in the 60s when it was applied to commercial uses, in trying to solve complicated problems and calculations in a matter of minutes. Buckland in his research, presented the idea that Information has a relationship with history; because, History he says "is concerned with analysing, weighing, and interpreting the available evidence, especially documentary evidence. Information systems are concerned with the selection, representation, and preservation of available evidence, especially documents. Therefore, it means that, "no documents, no history", citing the historian Fustel de Coulanges (Buckland 1998). He also went on to state that, "the creation, survival, and accessibility of documents is an accident-prone matter," because, as he says, "in the way the "content" is collected and presented". Many analysts see information systems as a multi and interrelated discipline. The Venn diagram illustrates the multi and interdisciplinary nature of information systems. It shows how the history of IS coincides with the history of Computer Science (Shallit, 1995), in a relationship that began long ago and before modern computer science discipline of the 20th Century.

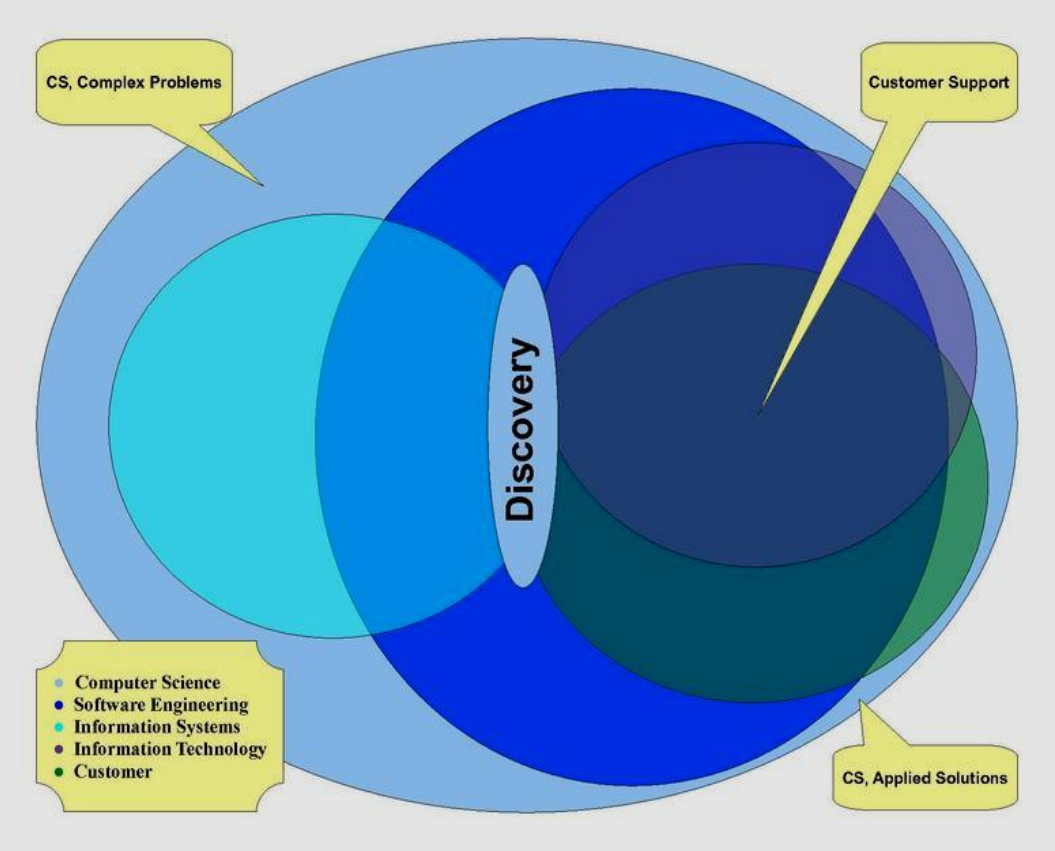


Figure 2.1 The CS Venn diagram

The CS, SE, IS, IT, & Customer Venn Diagram, where functionality spans left and design spans right stemming from discovery. (UTC, 1 4 February 2010)

According to Benson and Standing (2002), the early information systems gave rise to the development of written languages - accountancy, taxation and banking. Kelly et al, (1999) believe that many legacy systems are still in operation today and are used for circulation of information and ideas. The systems are constantly updated to promote ethnographic approaches, which ensures data integrity, and to improve the social effectiveness and efficiency of the whole process. Generally, information systems are focused upon processing information within organisations, especially within business enterprises, and the benefits are transmitted and shared with modern society (Jessup and Valacich, 2008). Computers have in the past been used to access, processed, and filter large amounts of data. It also allowed mundane tasks to be automated but the system did not produce knowledge as is required today for competitive advantage. It only speeded up processes. The fact that the basis of information system was in Science and engineering brought with it a mind-set that was

rigorous, logical and grounded in mathematical notations. This notion, over the years had had very huge influence on development of the discipline (Akhgar, 2003; Kelly et al., 1999; Benson and Standing, 2002; Jessup, 2008).

Information System (IS) as a discipline is rapidly evolving, due partly to the rapidly developing technologies, which is making it not to be stable. Also, with the increasing demands of real world environments (including the many challenges such as security and usability), it means that the perfect system will never be designed and developed to meet the demands; rather, the search is now directed towards finding the best possible compromise.

2.2 Defining Information Systems

Information systems (IS) and its domain come with many different broad and/or narrow definitions; some of which perhaps are related to many other scientific disciplines. The term "Information Systems" in itself has different meanings to different people both in academia and industry. Some analysts now believe that information system has a social dimension (Akhgar, 2003; Kelly et al, 1999), as many of the problems encountered have been identified as being human-related. Therefore, what is an "Information System?" may in turn provide varied answers that need to be understood. Information System as an academic discipline has its roots in computer science (Ahituv and Neumann, 1990; Akhgar, 2003; Polack 2009) and has grown dramatically over the past decades to include other disciplines. Benson and Standing (2002) are of the opinion that, IS does not exist in a vacuum. Therefore defining IS involves looking at the components and key concepts, describing the paradigm and evolution of the IS discipline and its relationship with other disciplines; and examining information systems in practice.

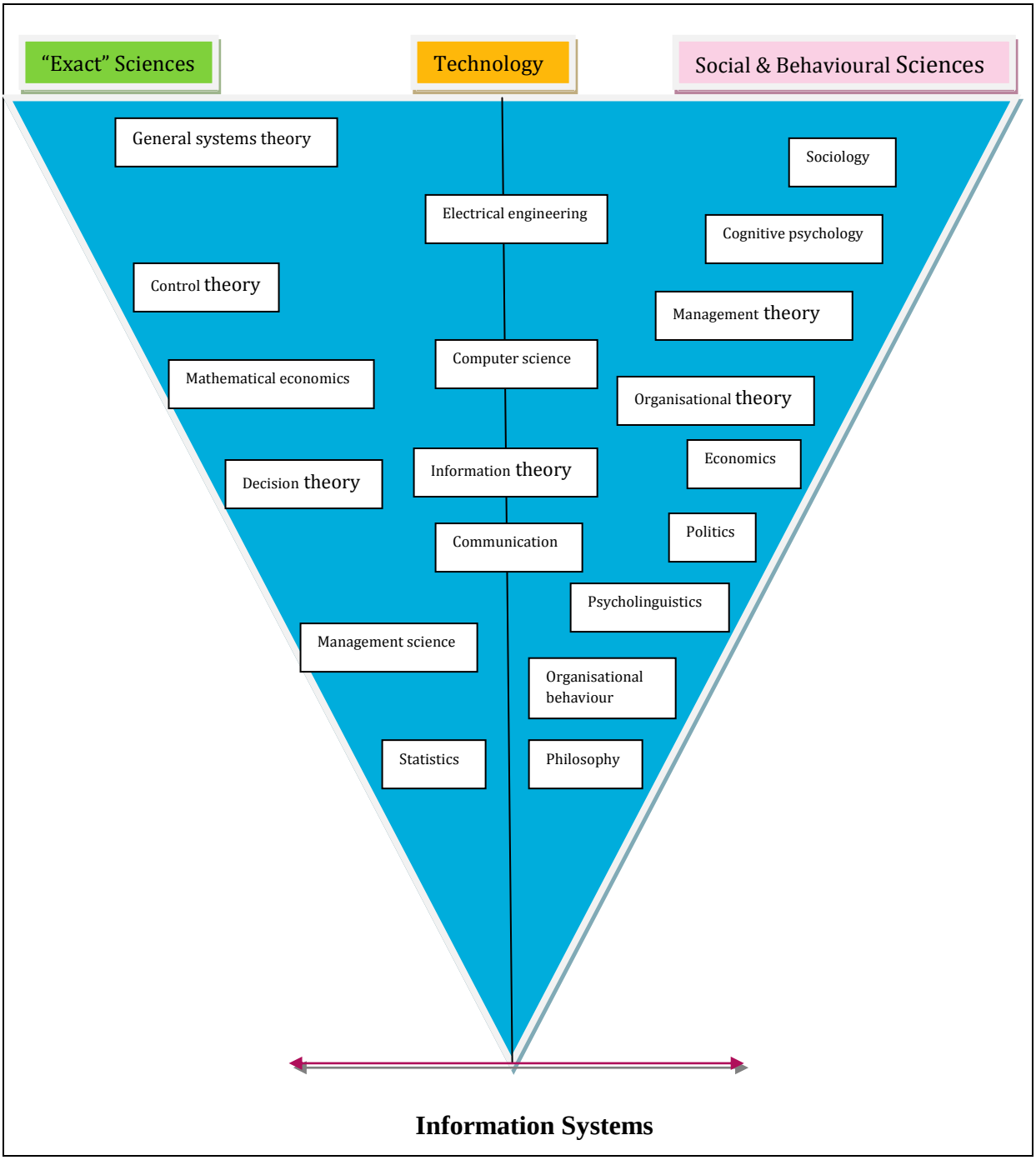


Figure 2.2 Foundations of Information Systems (Culled from Akhgar, 2003)

In order to understand this, one need to understand where information is coming from and in what context. Information comes from processed data, and an interpretation of the information forms knowledge. There is a very linear relationship between data, information and knowledge. One is derived from the other; in that, you need data to get information and from information, knowledge can be acquired based on experience and interpretation to assist decision making.

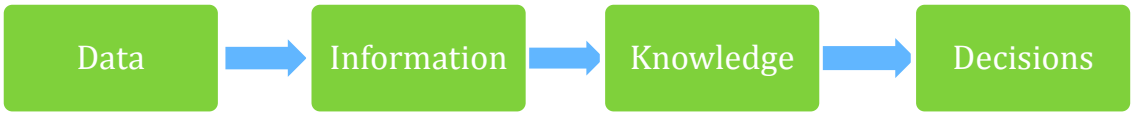


Figure 2.3: Relationship between data, information and knowledge

The words, "information System" can be separated in parts to see what they each mean. "Information" as we know is derived from data (Bell and Wood-Harper, 1998).

- Data are raw inputs of information systems, e.g. customer, order and payment details gathered over a period.
- Information is the output of processed or value-added data (i.e. Data are input, manipulated or processed in some way to give information as output) to highlight trends or features.
- Knowledge comes from understanding what the information means or implies; and according to Benson and Standing, (2002), it is a human thing, – which is very subjective and based on experience.

Meanwhile, a "system" according to Dr Alfred Howarth, is 'a collection of parts that work together to achieve some purpose'. This definition is however coming from a computing perspective, which is human-made and physical in nature. A distinction can be made purely from an abstract point or philosophical paradigm; which sees a system as 'a set of related ideas or constructs which are organised in some way' (Benson and Standing, 2002). Some systems might be a combination of physical, natural, abstract, and human-made subsystems. Some generalisations about systems include:

- A system has a purpose or function
- A system has a context or environment in which it has applicability or operates in
- A system has a boundary which marks the limits of its environment
- The removal of a single component of a system will cause that system to fail.
- A component may belong to more than one system
- A system usually has inputs and outputs
- Complex systems usually consist of subsystems which in turn may have subsystems of their own

This list demonstrates the interconnectivity of systems, subsystems and components that makes life so difficult for system developers. Where to draw the boundary of a system is one of the most difficult questions that have to be considered in designing any systems. It is for this reason that, systems thinking, methods, tools and techniques have been developed to facilitate this process. In general therefore, one can say that information systems is all about generating and managing information for a purpose. The purpose is to acquire good knowledge for competitive advantage and therefore must be protected. However, generating and managing the information requires that the system be made easy to access and easy to use. Drucker (1993) equates knowledge with power, which makes it difficult to share, especially amongst employees. That is why many businesses are now faced with the uphill task of trying to encourage employees to share knowledge rather than keep it to themselves. Over the last two decades, the knowledge economy has grown in importance, and the focus has shifted now from information to knowledge with an attached value to the user or owner. With all these in mind, components of the system - the people, processes, information and knowledge must be securely protected but readily available, easy to access, delivered and used at optimum speed and time optimum to achieve the desired objectives. Therefore, the definition of Information system has been enlarged to generally include the social dimension; with components and how they interact in their environment, and consist of:

- People
- Data/information
- Processes/procedures
- Software
- Hardware
- Communications

2.3 Types of information systems in organisations:

1. Transaction Processing System (TPS): A TPS collects and stores information about transactions in the business. It can be connected to the other systems, so that it informs the employees about the inventory stocks. It can be used to generate reports for high level managers and also be used for generating day to day transactions for low level managers. For example, the till machines used in McDonald's or any other store can be called as a Transaction Processing System.

2. Decision Support Systems (DSS): A Decision support system helps the managers in making decisions. It analyses the large volumes of data and gives the managers the required data or information using which he can make decisions. An Sql Server Analysis 2005 software application can be used to generate high quality reports from the large databases based on a Sql query and the generated report can be used by the managers in making decisions. For example, a report of the sales fluctuations in different months can be used by the managers to decide the needed level of stocks in the stores.
3. Management Information System (MIS): A MIS usually used for providing the managers with the current and the past operational data. The input to these system is manly a TPS. Using the data from the TPS, these systems can generate two types of reports; Summary report, which accumulates the data from the several transactions and presents it in a condensed form and a exception report, which outlines any deviations between actual output and expected output.
4. Executive Information Systems (EIS): These are used to provide the top level executives in the organisation with the data in a condensed form allowing them to drill down to the low level of data. These systems use data from both MIS and TPS as their inputs and they are very expensive and require good staff to operate them.
5. Office Automation Systems (OAS): OAS provides individuals with effective ways to process personal and organisational data, perform calculations, and create documents. (for example, Word Processing, spreadsheets, file managers, personal calendars, presentation packages). They are used for increasing personal productivity and reducing "paper warfare". OAS software tools are often integrated (for example, Word processor can import a graph from a spreadsheet) and designed for easy operation.

2.4 Overview of Information Systems Development Methodologies

This section examines Information Systems Development Methodologies (ISDMs) in relation to the issues of usability and security, and will attempt to identify the limitations of the existing methodologies and technologies in addressing them. The Information Systems (IS) development strategies are presently not only poor, but also incorporate haphazard approaches (for example, trial ‘n’ error) which may not be capable of meeting the complex

requirements of modern information systems, such as effective usability and robust security features. Therefore, there is a need to establish whether or not structured, formalised ISDMs or a flexible/adaptable unstructured methodology can best be used to develop IS that will cope with the heavy demands of usability and security. The section explores the IS development methodologies as a premise to highlight whether or not they have met or can meet the security and usability requirements with or without modifications, And also how the existing ISDMs have encompassed the soft aspects of usability and hard aspects security within the architectures and in what combination or proportion. It will attempt to identify which methodologies best address the security and usability needs, and in what proportions if a trade-off? It will attempt to demonstrate that a system designed for high usability requires a totally different approach, and involves a compromise or trade-off on security and vice versa. So, how can a design approach address this challenge? The analysis of ISDMs carried throughout this thesis, it is hoped, will assist in the body of knowledge and proposal of a new ISDM specifically designed to address what user concerns are, with regards to usable and secured IS from a design and development perspective. The advantages and disadvantages of using an ISDM must be examined in order to assist in determining their usefulness in designing usable secured applications. Developers use ISDMs in different ways; many use one ISDM, while others use a combination of ISDMs. Ultimately, systems development methodologies are supposed to benefit the developer. Some researchers believe that there has been a lack of detailed research into the use of systems development methodologies and have recommended that there should be a 'clearer understanding of the realities of software development' (CSTB, 1990). This seems to be true in the case of developing usable security for mobile information systems, as it is a very dynamic and relatively new area in the world of development methodologies. The internet and the WWW have made the task very daunting as users demand more interactivity and privacy on their system and/or protection of vital personal data whilst on the move.

Therefore there are urgent needs to make these systems very usable and secure for both businesses and the ordinary users. The apparent lack of methodologies to specifically address these areas of need has exacerbated the problems.

Users and businesses are increasingly becoming mobile and more demand for ubiquitous technologies or services such as 'Cloud,' have increased with complexity in such a way that, the old methods of developing and deploying IS can no longer be applied and may need a great deal of modification and/or contextualization to cope with vast amounts of user or

business information requirements – such as easy access to data, effectiveness of use and privacy and protection, since data can be accessed by anyone, anytime, anyhow and from anywhere in the world, using various devices, platforms and/or protocols.

2.5 Defining a Methodology

The term methodology can be described as method of completing a task or, the processes undertaken in order to achieve an end target or goal. It has its origins in the Greek language and meaning "the study of methods". The Oxford dictionary defines a methodology as the "study of systematic methods of scientific research". And according to Jayaratna (1996), in the context of both Information systems and Information technology, the term methodology has the same meaning as "method" and they are often used interchangeably within the Information systems domain. Methodologies are often developed through a combination of theory and practice. There is no single definition of a methodology and many suggested definitions often do not even resemble each other. An information systems methodology can be referred to as "a methodical approach to information systems planning, analysis, design, construction and evolution" (Olle, 1991). Information systems themselves are basically a means of informing people or a source of necessary data. In other words, information systems contain information regarding organisations and their environments. According to Checkland (1981), a methodology is something which "lacks the precision of a technique but will be firmer guide to action than philosophy. Where a technique tells you how and a philosophy tells you what, a methodology will contain elements of both." However, researchers Jayaratna (1996) and Avison and Wood-Harper (1990) argue that Checkland's definition was limiting in that it focused on context rather than both context and content. In the endeavour, Avison and Wood-Harper (1990) put forth their definition of a methodology as: "a coherent collection of concepts, beliefs, values, and principles supported to help problem-solving groups to perceive, generate, assess and carryout, in a non-random way, changes to an information situation." In trying to elaborate on the definition, Avison and Fitzgerald (2006) describe methodologies as 'recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management and training for developers of Information Systems'.

On a very basic level, a methodology can be described as having three main components:

(a) A breakdown of guidelines referring to systems development on what to do and when to do it.
(b) The techniques and actual methods of how to do it.
(c) And information and recommendations on how to manage the quality of results.

However, Jayaratna (1994), looks at defining methodology from a holistic and goal driven point of view by stating that a methodology is; "an explicit way structuring one's thinking and actions. Methodologies contain model(s) and reflect particular perspectives of "reality" based on a set of philosophical paradigms. A methodology should tell you "what" steps to take and "how" to perform those steps but most importantly the reasons "why" those steps should be taken, in that particular order". Methodologies have an important role to play in producing information systems of all kinds. It can be difficult to select one methodology, as there are so many to choose from. There are even methodologies or frameworks, for example the Normative Information Model-based Systems Analysis and Design (NIMSAD) framework (Jayaratna, 1994; 1996), which are used for evaluating other methodologies. The NIMSAD is a generic framework which can even be used for evaluating ISDMs. The framework suggests that effective application of a method depends on three elements: the method itself, the person who applies the method and the context in which the method is applied (Jayaratna, 1994). And from analysis of NIMSAD framework, it could be deduced that, it treats the evaluation of a method as a dynamic activity that is carried out before, during and after the application of the method.

Avison and Fitzgerald, (1988) described a methodology as a "collection of procedures, techniques, tools and documentation aids which will help the systems developers in their efforts to implement a new information system". According to Crinnion, (1995) one of the most pressing problems faced by systems development managers and systems analysts is 'how to provide the full range of methods and facilities necessary for the analysis, design and construction of business information systems'. This is also true in cases of mobile information systems and m/e-commerce systems development. Therefore, any lack of methodologies to cater for this type of development will expose inadequacies of the system.

In this section we provide brief analysis of the selected existing methodologies used for the design and development of Information systems. The fact that this research is looking at ways of making information systems very usable and secure; the selection of these methodologies

will be an attempt to highlight the good, the bad and ugliness of each of them. For the purposes of this research, we have focused and briefly analysed three existing and very widely used methodologies - Structured Systems Analysis and Design Method (SSADM), Checkland's Soft Systems Methodology (SSM) (Checkland, 1991) and Dynamic Systems Development Methodology (DSDM) (DSDM Consortium, 1995). These methodologies have been selected as being representatives of particular paradigmatic approaches to IS design. SSADM represents data and process oriented methods; SSM represents user oriented methods and DSDM represents output or system oriented methods. The selection of SSADM can be looked at from a security point of view; the fact that it maintains a very disciplined structured approach, and is a rigourous, tried and tested and 'hard' methodology (Avison and Fitzgerald, 1995; 2006; SmartDraw, 2011), and have tendencies towards the reductionist model {Bell and Wood-Harper, 1998), makes it comparatively better on the hard aspects of security, where the problem is in the mind of the expert, who also defines the part boundary and operates in a very controlled environment. SSM on the other hand, uses a less structured approach (Patching, 1990), which is systemic as shown in figure 2.4 (Bell and Wood-Harper, 1998; Checkland and Scholes, 1990) and highly user (stakeholder) focused and uses rich picture to help users understand the organisational situation, (soft aspects), therefore better on usability aspects. The DSDM approach is flexible in that, it attempts to combine some features of both the SSADM and SSM methodologies within its processes to deliver a system that is structured enough to cover some hard aspects of security and involves users at all stages of the iterative process to improve usability of the system. DSDM uses an incremental prototyping approach where the system to be developed is divided into components that can be developed separately (Avison and Fitzgerald, 2006). There are many other methodologies to choose from; however limitations of time and research resources do not facilitate their detail mention here.

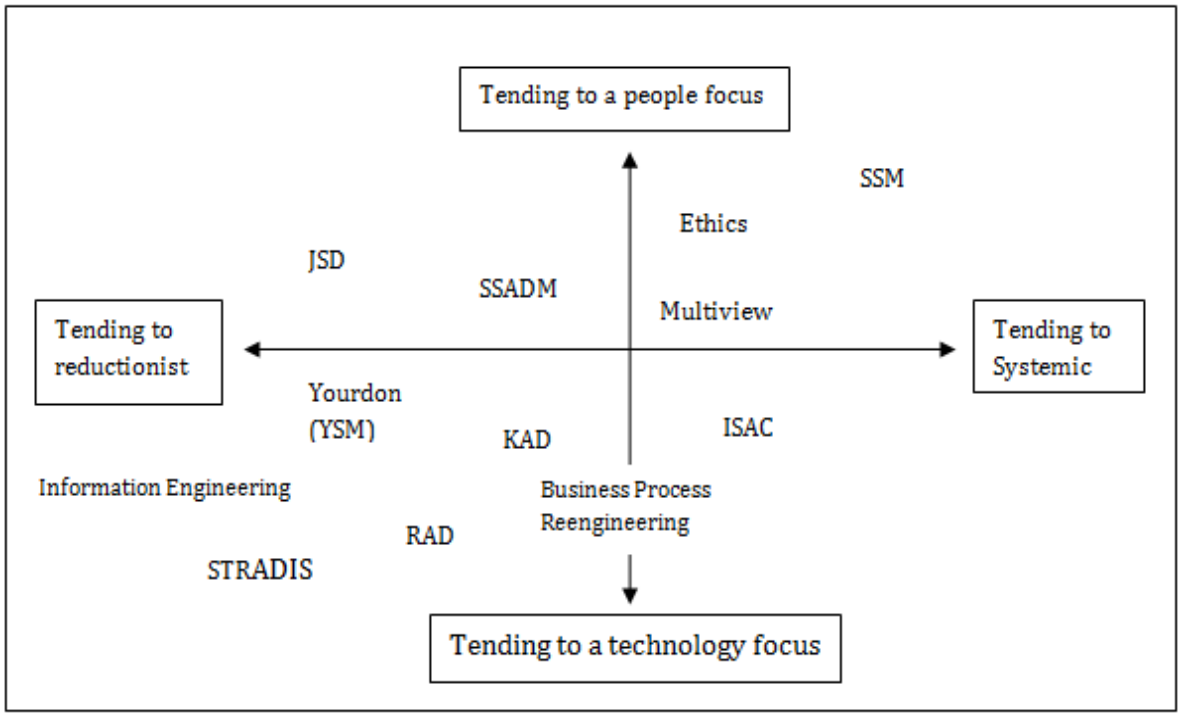


Figure 2.4 Methodologies - Systemic and Reductionist, People and Technology

(Culled from Bell and Wood-Harper, 1998)

Each of the selected methodologies will be dealt with in more details.

2.6 The IS Methodologies in brief

There are many different IS methodologies that have been proposed since the 1970s (Hawryszkiewycz, 1998), with many different ways to design a computer information system. The majority tend to come from the different procedures used in the various software development enclaves and the varying appropriateness of necessary paradigms from application to application. Overall, there have basically been three approaches in the area of information system development. They include; process-oriented, data-oriented and object-oriented approaches. The process oriented approach was the earliest approach used. Later advances in technologies - both software and hardware – saw dramatic shifts from this archaic approach to data-oriented; and now increasingly, to object-oriented approach. Generally, information systems require a methodology (whether process, data or object oriented) to take their development forward from the initial users requirements to an implemented documented functioning system which satisfies the end-users in its entirety -

functionality, interface (Akhgar, 1997a; Akhgar, 1997b; Beynon-Davies, 1998) and security (Arreymbi, 2007).

As earlier stated and for the purposes of this research, the discussion of existing methodologies will focus mainly on the three selected methodologies - SSADM, SSM and DSDM. The reason for their selection is based on some factors such as Costs and Benefits associated with the methodology. For example, SSADM has been reported to have lower lifetime costs and is Information System (database) specific. But benefits mainly large scale projects. DSDM on its part is a very iterative process and user expectation must be carefully handled. DSDM also benefits small projects and user interface development. Meanwhile, SSM is very user focused and benefits both large and small projects.

More of the selection criteria come on basis of how the each tackle the issues of usability and security. The presumptions are that; SSADM is believed to be structured, very comprehensive, and very strong on the hard aspects than soft aspects (Avison and Fitzgerald, 1995; Bell and Wood-Harper, 1998), therefore good on security; meanwhile SSM is very user focused thus relative strong on soft aspects (Patching, 1990) and weak on the hard aspects, therefore good on usability. DSDM on its part involves a combination of the processes, structured and unstructured, therefore very flexible, with the potential to better deliver the stated objectives of designing a usable and secure system. It has the agility, responsiveness and controls for the rapid development of usable and secure system to tight time scales. It is worth noting that there are many other systems methodologies that exist; and every approach depends on the target area of application. However, the methodologies that have been chosen here, are three of the widely used in systems design and development. The methodologies will later be introduced and providing some explanations of how they are used, when they are used and why they are used in systems development. It can be argued (Olle, 1991) that all methodologies have the same target; however they vary simply in the means by which they reach this target. In the discussion, a brief outline of the steps involved in the existing methodologies, such as those mentioned above will be provided; and looked at in relation to their ability to respond to user needs of security and usability in developing information systems. This is an attempt to see which of them, if any, is more suitable for this purpose, or if a new methodology is required and/or will be more advantageous in solving the problems.

2.6.1 SSADM

Structured Systems Analysis and Design Methodology or SSADM according to Avison and Fitzgerald (1999), SmartDraw, (2011); is an updated version of the Waterfall model, and is a hard methodology that has been promoted by UK government since the 1980's. SSADM uses a top-down approach that starts with defining the information system strategy and then developing a feasibility study module (Akhgar, 1998). It covers aspects of system life-cycle from the feasibility study stage to the production of a physical design. It has often been called a data-driven methodology and employs a combination of processes such as data flow diagramming, entity-relationship modelling, data normalisation and life history analysis. Another version of SSADM known as SSADM4+ was released in 1995 and came about because of competition in the use of other ISDMs. Crinnion (1995) is of the opinion that, most, if not all 'well established' methodologies such as SSADM which are currently on the market are being reviewed and updated. Perhaps a version of SSADM created specifically or modified specifically to suit information systems (e.g. e-commerce) development will be very useful. If this were possible, then the existing processes within SSADM can be altered and used as a good foundation to create an e-commerce methodology. At present businesses are always on the lookout for faster, flexible and more cost effective ways of developing information systems. Businesses such as Smartdraw support the use of SSADM which elaborates on development of object-oriented applications with graphical user interface.

SSADM is comprised of a set of specific rules and guidelines for developers to follow (Stowell, 1995; Akhgar and Siddiqi, 2004). It is used mostly for medium to large projects, but another version called Micro-SSADM, can be used for smaller projects. In much detail, the SSADM process sets out a waterfall view of systems development, in which there are a series of stages or steps, each of which leads to the next. This is in contrast to the RAD - rapid application development - method, which pre-supposes a need to conduct steps in parallel.

The SSADM stages are as follows:

- Feasibility
- Investigation of the current environment
- Business systems options
- Definition of requirements
- Technical system options
- Logical design

- Physical design

SSADM has six main processes which are divided into sub-processes. Each process has a specific and desired end result. The six processes can be divided into two main processes; - three of systems analysis and three of systems design.

(a) Process one in SSADM is the Analysis of the current system - This involves a detailed investigation of the business in its existing state and the creation of a current data flow (Avison and Fitzgerald, 1999). Its current data processing methods and volume are checked. There should also be a feasibility study and report, and a definition of the problem within the current system. This is the first stage in which the requirements of the system must be established. This involves the formulation of a set of requirements for the systems development project and the creation of a plan as to how the requirements will be met. The business systems options report forms the final step in this process. The planning process in SSADM constitutes requirements specification.

(b) Requirements specification - This process defines the requirements of the system in full based on the investigations of the first process. Since most businesses have many employees and many different tasks to complete, there are often many requirements. Requirements are often prioritised however where possible it is essential that all of them are met. At this stage the requirements are often specified in a narrative form however data flow diagrams can also be used. At this stage, the usability and security requirements can be specified and addressed. The aims and objectives of the new system must also be listed at this stage, indicating who and where are the targets and what the system should entail etc.

(c) User selection of service levels or logical system specification - During this process of the systems development, the users are presented with various options for producing the new system. It can often include a technical explanation of what the new system will do, such as centralised data processing or batch processing. It is very important to include the end users in development as much as possible, after all they will use the system and they must approve of all aspects of design before the system can be used in everyday business practices. At this stage the end users may choose from a selection of options presented to them by the developer. They are provided with possible scenarios as to what each option will entail such as costs, staffing needs. It can often be described as a feasibility study containing a problem definition, which defines the systems current problems and future

needs and also the project definition, which defines the means of eliminating current problems. The amount of detail at this stage depends on the size of the project at hand. This is the stage where most aspects of usability and some of security can be addressed, as end user participation is encouraged. This is where aspects of how system will be engineered are looked at e.g. what programming language will be used, such as Java or HTML. In this process all the information required to be displayed is engineered and a technical document can be produced to show what the new system will eventually look like. It is good for all employees to know what the system contains and how to navigate it. Again, it depends on who is to use the system and what for. The end users of a most information systems are more likely to be the customers, rather than the employees of a business. Therefore it is for this reason why the HCI (Human Computer Interface) should be user friendly to meet the customers.

(d) Detailed data design - At this stage in SSADM the data and its relationships is defined. This involves data analysis, which looks at third normal form relations, also called document-driven data analysis. It is a bottom up approach. A sub phase in this stage is called logical data structuring or LDS. This includes entity modelling or an analysis of the relationships between entities. This is a top down means of producing a data structure. This process takes time, as there is much data involved, however it will save time in the long term if it is done correctly. A composite logical design or CLD is produced and this becomes the first part of the database design. This is the stage where system security such as access control can be implemented in the system structures. It maps the logical and physical data structure together. The next stage in SSADM is detailed procedure design.

(e) Detailed procedure design - At this stage things get more technical. The end users should now be clearer on what the end system will be like and they should be happy for it to proceed. Again this is where usability and security features are highly envisaged. The end functions of the new system are catalogued to ensure that all requirements are met and often a prototype system is produced for approval. Any problems with the prototype system at this stage should be resolved. The prototype may be on paper or on computer depending on the requests of the user or the recommendations of the developer. As with any information system, the end user needs to look at a prototype as in SSADM, to ensure that they are happy with what it will look like and how it will function.

(f) Physical data/design control - The prototype from the last stage in SSADM may be developed further. The development of the system is planned and the means of testing the system will be defined. Program specifications and operating procedures are produced

(Avison and Fitzgerald, 1999). This methodology is more suited for database design and very good on the implementation of security within the system. Documentation is refined and will include a test plan, implementation and operations plan and also a user guide to the system. The system evaluation can be helpful for both customers and systems developers. It enables customers to identify any problems they have with the end result in order to solve them and it helps systems developers evaluate the success of the work.

The above stages in SSADM contain many sub-stages which have not all been described in detail at this point. The details in which they are used will depend on system requirements and the size of the project at hand. Meanwhile, because of the structured nature of this process, a system built using SSADM can be considered to incorporate adequate security features as its focus or objective but with minimal or poor usability features.

The SmartDraw Software Company illustrated the SSADM stages as in figure 2.5 (SmartDraw, 2011)

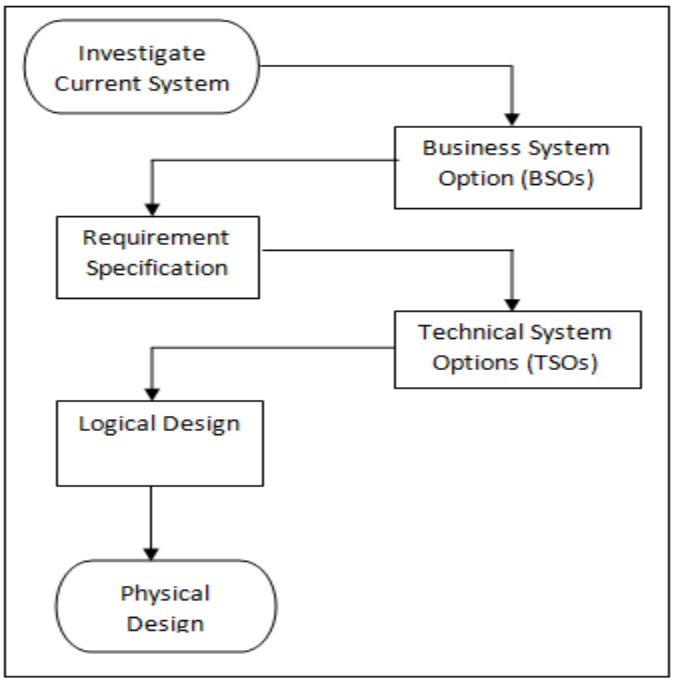


Figure 2.5 Stages in SSADM

- Investigate the current system: This is the first stage in SSADM4 and involves drawing a Data Flow Diagram (DFD) and a logical Data Model (LDM) that highlights the current system. LDMs are identical to Entity-Relationship-Diagrams (SmartDraw, 2011). Investigate the current system is about carrying out Feasibility study (Moynihan, 1993) and involves outline investigation of systems that creates a list of business and technical options that satisfies requirements.
- Business System Options (BSO): This stage describes possible new systems in terms of functionality and implementation issues and uses text, simple DFDs and LFDs (SmartDraw, 2011). The requirements of a new system are determined for example, in terms of security, a series of business options are established; for example, a decentralised system, where branches are involved or a centralised system, where one of the listed option is chosen (Moynihan, 1993)
- Requirement Specification: This stage develops the system by deliverables of detailed data processes and data required (Moynihan, 1993)
- Technical System Options (TSOs): This stage outlines the cost, benefits and constraints required in implementing the specification (SmartDraw, 2011).
- Logical design: This stage according to SmartDraw, (2011) defines how data is processed by the system and specify user dialogues. Moynihan, (1993) refers to this stage as logical system specification, in which various technical options are designed and examined, such as mini and micro-computer systems.
- Physical Design: in this stage, the logical processes are used to design software that will process data, for example EDI (Electronic Data Interchange).

2.6.1.1 Benefits of SSADM

SSADM is suitable for IS systems where there is high volume of business transactions (Laudon and Laudon, 2010; Schumacher, 2001), because the methodology specifies the flow and tasks of a development process; for example through the use of logical Modelling Diagrams (LMDs) and Data Flow Diagrams (DFDs) (SmartDraw, 2011). SSADM also produces detailed documentation of the project. This can be compared with V-model development, where at each stage in the life cycle phase, such as requirement specification; there is deliverable produced in form of documentation as an activity which includes validation and verification by Quality Assurance and testing experts.

SSADM popularity in large IS projects are due to the following (Schumacher, 2001):

- Timeliness: SSADM allows management and control of a project; this is because the activities in the projects are divided into stages. Management and control is emphasized in the project management process such as Prince 2; which allows specialist working on projects to monitor and track the project. The way SSADM is decomposed into stages can be compared with sequential SDLC such as Waterfall model, where activities are broken down into stages and each stage is completed before the next.
- Usability: (this comment on usability is from Schumacher who made no comment on security) Within SSADM emphasis is on ensuring the system satisfies user's requirements as a result, the system model is developed and a comprehensive demand analysis is carried out.
- Respond to changes in the business environment: In SSADM documentation of the project progress is a very important activity, issues like business needs and business objects are taken into consideration while the project is being developed; this allows tailoring planning of the project to business requirements. However, because of its nature, response to major changes can be very slow indeed.
- Effective use of skills: SSADM does not require specialised skills and is easily taught to staff and users, Modelling and diagramming tools are used, for adaptability and flexibility, commercial Case tools are tailored to the specific system and user requirements. This supports SSADM ability to respond to changes in business environment, usability and timeliness as previously discussed.
- Better Quality: SSADM defines certain level of quality at the start of the project, and monitors and control quality by checking the system. This can be compared with SDLC methodology e.g. V-Model, which emphasizes validation and verification at each life cycle stage.
- Improvement of productivity: SSADM improves productivity of IS and projects by encouraging on-time delivery, promoting better quality, meeting business requirements and using human resources effectively.

2.6. 1.2 Evaluation of SSADM

Many analysts see SSADM, as detailed method which includes many techniques that deals with each aspect of the system but more especially the hard aspects of security than the soft aspects of usability. The logical data modelling present the data structure of the system and

the data flow defines the data process. The requirement catalogue stores the requirement information about the development system, including the required resources that are defined in feasibility and quality assurance that ensures the product satisfies specified quality standards, at each phase in the life cycle there is quality assurance activity.

SSADM supports user involvement through the use of interviews and discussions in identifications of system requirements. Users review deliverable at each lifecycle stage in collaboration with the analyst to identify defects in the requirements. Early identification of defects is supported in that, the earlier a defect is identified, the less expensive it is to fix. This is known as cost escalation model, and called early testing in some models, where each activity in is tested at the life-cycle phase.

Prototyping is used to verify requirements of the system. The prototype supports the user in accepting and using the new system and increases the user's usability acceptance of the system. Prototyping is used in incremental models, where the users get to know how the system is and if they are ready to use the system. This is an important aspect of Human Computer Interaction (HCI). User's interaction with the analyst to choose options to use in the system within the Business System Option (BSO) increases user familiarity, acceptability and usability of the system. Other methodologies such as Prince 2 project management methodology define the role of users as mandatory role for the success of the project.

Overall, SSADM provides the following benefits:

- It is a mature process
- Very hierarchical
- Better management and control because it can be used in conjunction with other methods such as Prince 2
- Presents three different views of the system
- Separation of logical and physical aspects of the system
- User involvement – very structured user involvement early in requirements rather than in design stages.
- Develop Better quality systems
- Well-defined techniques and documentation

However, a methodology like SSADM does not come without drawbacks as can be seen:

- It is large, therefore cannot be used in all circumstances
- Long processes
- Involves a lot of investment in cost and time
- Staff training and learning curve is long due to several techniques involved
- Not flexible enough to be highly iterative
- Large amount of standard documentations involved (preparation and presentation)

The above refer to standard SSDAM; however, there are variants such as micro-SSADM which have been developed to address some of these limitations (Avison and Fitzgerald, 2003; 2006).

2.6.2 Soft Systems Methodology (SSM)

Soft Systems Methodology (SSM) deals mainly with the soft aspects of systems development and only touch on some elements of the hard systems aspects (Checkland & Scholes, 1990). As a weakness SSM does not support the other elements of the hard approach such as data, events and designing interfaces. It is more a usability oriented than security oriented system design and development process because of its high user involvement (cognitive and/or psychological aspects). The SSM includes techniques, such as rich pictures, which help the users understand the organisational situation and therefore point to areas for organisational improvement through the use of IS (Avison and Fitzgerald, 2006) and was designed and used to solve real life problems (Lester, 2008) in management organisation and policy context in situations where it is challenging to provide easy solutions to system problems (Patching, 1990). Schmidt (2006) looks at SSM as a learning system with the emphasis that; system ideas, metaphors etc. can be used and helpful in understanding problems and the situations. This is an important concept in HCI in that, a metaphor improves user's understanding of the system. Learning in SSM is achieved by comparing pure models of human activity systems (HAS) with the realised ones (Bell and Wood-Harper, 1998; Schmidt, 2006, Avison and Fitzgerald, 2006). Therefore, an understanding of the model is crucial in understanding the conception of the real problem. A mental model and perceived user's model of task is important in HCI as it promotes usability because when users are familiar with the model, they will be familiar with the application.

Schmidt (2006) argues that, the process of analysing and modelling has been altered over the years to reflect circumstances and/or situations. The first model elaborates on identifying clients, problem domain and problem owners, and the list identified at this stage is a potential list of model for relevant system. The second model examines problem situation from a social view point, focusing on values, roles and the social system, social consideration is important for designing systems for users, socio-cultural context is important in HCI which improves Usability. The analyses from the social view point ensure attention is given to problem situation as a culture. The final analyses method views problem from political position by analysing the decomposition of power, how it is obtained, preserved and passed on.

The monitoring and control parts may be analysed by defining three types of system criteria (Schmidt, 2006):

- Effectiveness: is this the correct thing to do?
- Efficacy: do the means work?
- Efficiency: relates to rescores, asking the question; is there a minimum use of resources?

The methodology is divided into seven stage process (Lester, 2008).

- Identification of problem situation designed to intervene in
- Internal representation of a picture by researching the situation
- Choosing perspective and key processes that will take place in the system
- Developing a Conceptual Model of the systems to be changed
- Comparing the model with real-world situation
- Define Changes to be implemented
- Take Action

Stowell, (1995) described SSM (which is more than 20 years old), as a set of methods or a group of concepts used in a way that can easily be adapted to the situation being analysed. In other words, the situation is divided into smaller parts to make analysis easier and more thorough. SSM uses a diagrammatic model in which the stages can be alternated and returned to depending on the situation (Patching, 1990). It is flexible and can be tailored to suit specific needs of an organisation or user. Checkland (1998) stated that, SSM is like a "formalised and organised version of the process of purposeful thinking which human beings undertake in their everyday affairs". The flexible nature of SSM will make it useful for online

systems development, because it incorporates a lot of acceptability and usability features within the system design and development process. The ability to alternate and return to stages in the process will be useful in an online web system development, as it provides a less rigid means of development.

In the SSM process, stage 1 describes the unstructured problem situation and is concerned with trying to find out about the problem situation from as many people (users) involved as possible. The analyst must try to establish the aims and objectives and examine the role of individuals. Formal and informal communication takes place. When the analyst has as much information as possible, it will then be used to produce a more formal rich picture, which is shown by diagram. The rich picture diagram depicts the processes in the system and how they relate to each other. The rich picture is devised to show the principal human, social and cultural activities relate to one another in the set environment (Bell & Wood-Harper, 1998). It will include the clients of the system, those involved in it, the tasks being performed and the environment in which it all occurs. It can also be a means of describing the problem to the person for whom the new system is being designed. It should identify the problems with the situation. This process of developing the rich picture is to provide an overall diagram of the situation and the factors that will affect the system design. In SSM, user requirements are clearly defined and the requirements follow software quality standards defined by the ISO9126 which are divided into usability, functionality, system reliability, efficiency and maintainability. It has been argued that SSM is a better means of pinpointing the problems rather than solving them.

The next stage in SSM provides the root definition of relevant systems. Problems are taken from the rich picture and the analyst suggests possible solutions to suit the problem e.g. conflicting departments may need a system to provide more communications between departmental boundaries. In other words a problem is examined and possible solutions are suggested. Checkland (1981; 1998), Bell & Wood-Harper (1998) all stated that the "root definition is a concise, tightly structured description of a human activity system which states what the system is". The root definition is produced using the CATWOE (Customer, Actors, Transformation, Weltanschauung, Owners, and Environment) checklist. The other stage is a pictorial representation of how functions in the root definition should be sequenced; also known as the Human Activity System (HAS) or Conceptual Model. In stage 5, the conceptual model and the rich picture are compared to establish what is happening in the real world situation. Stage 6 is the analysts' opportunity to again discuss the proposals for the new

system with those involved in order to ensure that it meets their requirements and will be approved when completed.

Finally the new system is approved (or may be modified if it is unsatisfactory) and it is implemented in the business. Approval of the new system is also vital in SSM in which the developer should investigate user satisfaction on existing applications as well as the application being designed for them. If the developer takes time to test the user response to other systems, then it is a very good way to assess what they themselves will like for their own system (acceptability of the system). This is done during the application domain stage where the developer will focus on user concerns, trust, feelings and readiness of using the system. SSM is a collaborative methodology in which the users and the analyst are a team who work together to achieve the best possible result, (i.e. a fully functional system that meets the needs of the business).

However, since human thinking is often as complex as the system being designed, there can be disadvantages to the division of processes. When a human role is examined individually, people can react differently than when everything is examined as a whole. Different people see problems differently and can have conflicting objectives and attitudes. Also it is important to look at the situation in its entirety as it is pointless to have effective sub-systems which cannot work together to make the whole system work its best. In a similar way, online systems can also be intimidating for the novice user. This is due to many factors e.g. security, maintainability, etc. It is important for the developer to alleviate any fears about any systems for a business through communication and explanation (Bell & Wood-Harper, 1998).

2.6.2.1 Strength and Weaknesses of SSM

The above analysis has shown SSM to be a very flexible method which can easily be adapted to suit user needs. The involvement of users early on and throughout the process makes it better of the soft aspects of usability than the hard aspects of security. As a weakness SSM does not support the other elements of the hard approach such as data, events and designing interfaces. SSM is very limiting when it comes to adopting it for very large complex projects.

2.6.3 Dynamic Systems Development Methodology (DSDM)

DSDM or Dynamic Systems Development Methodology is a framework which is based originally on RAD (Rapid Applications Development) and supported by its continuous user involvement in an iterative development, and incremental approach which is responsive to changing requirements, in order to develop a system that meets the business needs on time and on budget (DSDM Consortium, 1995). It is one of Agile methods which came into use in the early 1990s as a result of the need for a methodology which can develop new software systems as quickly as possible. This will suit online projects as it is evident that the majority of businesses require their online systems very quickly. RAD in its early days was said to lack structure and analysts did not really follow any set rules in systems development using RAD. Because of this lack of structure, in 1994 a consortium made up of big computer organisations, user organisations and vendors was set up to standardise and regulate RAD; this is when RAD became DSDM. A change in business practice meant that DSDM proved effective in developing systems quickly and cost effectively. DSDM is also a flexible methodology and can be adapted to suit the system being developed. This may mean that it can be a suitable basis for mobile web based systems development because the process can be cost effective. In DSDM, the user involvement in the development process is very much encouraged. This is to ensure that the user is satisfied with the end result.

According to DSDM Consortium (1995), DSDM is comprised of a process and a set of products adapted to suit individual business needs. It takes a pragmatic holistic approach to systems development using existing business requirements. The development process uses iterative, incremental prototyping carried out within time constraints and strives to utilise the available resources. It is said to be carried out using structured analysis and design (SSADM) or object orientated analysis and design (OOADM). Within DSDM, teamwork and testing are encouraged at various stages along the way. Prioritisation approach is used, such as MoSCoW (Must-have, Should-have, Could-have, Want-to-have, but not this time). DSDM uses time-boxing, which is about investigation, consolidation and refinement of essential requirements within a set time constraint. Any non-essential requirements are satisfied outside the time-box should there be any extra time left over. However, this has always been a major constraint in information system design and development, especially if the major requirements/objectives such as usability/accessibility and security are to be met. A main time-box is established which contains smaller time-boxes lasting a few weeks. DSDM is

suitable for small projects and larger ones if they are split into smaller projects. It tends to be useful when the product is not too complex and the requirements are clear.

2.6.3.1 Principles of DSDM

There are 9 underlying principles of DSDM consisting of four foundations and five starting-points for the structure of the method (DSDM Consortium, 1995).

These principles form the cornerstones of development using DSDM.

- Active user involvement – this is the main key in running an efficient and effective project, where both users and developers share a workplace, so that the decisions can be made accurately.
- Empowered teams with the authority to make decisions - The project team must be empowered to make decisions that are important to the progress of the project, without waiting for higher-level approval.
- A focus on frequent delivery of products - DSDM focuses on frequent delivery of products, with assumption that, to deliver something "good enough" earlier is always better than to deliver everything "perfectly" in the end. By delivering product frequently from an early stage of the project, the product can be tested and reviewed where the test record and review document can be taken into account at the next iteration or phase.
- Using fitness for business purpose as the essential criterion for acceptance of deliverables - The main criteria for acceptance of deliverable in DSDM is on delivering a system that addresses the current business needs. It is not so much directed at delivering a perfect system addressing all possible business needs, but focuses its efforts on critical functionality.
- Iterative and incremental development to ensure convergence on an accurate business solution - driven by user feedback to converge on effectiveness.
- Reversible changes during development – all changes are reversible.
- The high level scope and requirements should be base-lined before the project starts (i.e. requirements that are baselined at a high level).
- Integrated testing is carried out throughout the project life cycle.
- Communication, collaboration and cooperation between all project stakeholders is required to be efficient and effective.

Like other methodologies, DSDM process divides the system development life cycle into phases (stages) which includes:

- The feasibility study (i.e. can the project be completed within a required time; and will the existing resources be utilised with minimal costs).
- A business study is carried out to identify the requirements.
- Functional model iteration takes place during which a prototype of the new system is produced to demonstrate how it will work.
- System design and build iterations allow modification of the prototype to include non-functional requirements.
- Implementation as it implies will implement the new system in the user environment to test its functionality.
- Training the end user and providing user manuals is part of the implementation stage.
- Finally, maintenance of the new system is seen as an extension of the entire DSDM process.

2.6.3.2 Analysis of DSDM strength and Weaknesses

An important aspect of the DSDM is that of flexibility and adaptability of the requirements which caters for both the hard and soft aspects of the systems. Therefore, DSDM seem to be addressing only some aspects of security and usability in the process. It delivers better quality products because of increase testing carried throughout. However, the DSDM can be very complex, time consuming and expensive because of the high level of user involvement and /or iterative processes, product delivery at each stage of the process etc. It can be used for both small and large projects if properly managed.

2.7 Classification and characteristics of the methodologies

An important part of systems development is the decision making process when a systems developer comes to choose a methodology before a project. Many developers specialise in using one methodology but others use a combination of methodologies to develop a system. Therefore it is useful to classify methodologies by their characteristics. This makes the decisions on which to use easier. Some things are true of all methodologies, for example a methodology must be written so that it can be taught, learned and applied to a variety of development situations. They must not be too complex, but understandable and socially

acceptable. Sometimes the use of a methodology must be rewarded in some way, as they often involve people having to change their working habits and/or ways of approaching situations. They must eventually be approved by the end users.

As the project discusses DSDM, SSADM and SSM, again they will be used as examples in order to briefly demonstrate the way in which methodologies are classified. Avison and Fitzgerald (1999) maintain that it is essential to compare methodologies and classify them for academic reasons and to examine the nature of methodologies.

The tables below show brief results of analysis of some chosen methodologies.

Table 2.1 Classification of selected methodologies

	SSADM	***DSDM***	***SSM***
Approach	Reductionist and toolkit	Holistic	Embedded, open and reactive
Users	Government and commercial	Government, industry and commercial	Commercial and industry
Focus of methodology	Business requirement satisfaction	User involvement and business needs	Improving the problem situation

Table 2.2 Characteristics of the selected methodologies

	SSADM	***DSDM***	***SSM***
Characteristics	Encourage user involvement	Encourage user involvement	Encourage user involvement
	Focus on customisation and prototyping	Dynamic development	Based on Checkland's seven stage model
	Standard techniques and default structure	Users and developers share decision making powers	Problems seen as intellectual constructs and can be improved
	Flexible development	Development within	Analyst as part of the

		set time periods	problem situation
	Includes business modification by downsizing and delayering	Importance of meeting business requirements emphasised	Can be grafted onto hard systems methodology
	Case tools used to create diagrams	Testing at all stages important	Agendas provided for steering committees
	Systems development Template used that includes conceptual model and internal or external design	Collaboration and co-operation with all stakeholders	Is an excellent problem solving methodology
	Government standards used e.g. PRINCE	Changes can be reversed	Alternative views considered
	Social and technical aspects considered	Iterative and incremental in nature	Systematic approach

User involvement as defined by Avison and Fitzgerald (2003) does not provide a categorisation of the level of user involvement or usability, nor does it address security issues.

2.8 Challenges to IS development

In the recent past, the creation of some information systems, for example websites have been without the use of any methodologies; without planning or structure. They have been developed mostly through the process of 'trial and error' (Brooks, 1999). Often users come across websites that lack quality, standards and have unfriendly or unusable interfaces. They often do not meet the needs of the user or business and so, do nothing to increase productivity. Sometimes Internet systems are developed by those who follow only some of the processes in a methodology or by using a combination of methodology processes. Some

developers (Bell & Wood-Harper, 1998) argue that the use of formal ISDM's can limit the scope for creativity in web system design.

Over the years, it has been very obvious that IS development continued to play significant roles in many aspects of our daily lives. Information Technology (IT) has, and still is changing the way business is conducted in many industry sectors such as banking and securities, manufacturing and design and the services as well. And IS specialists are striving to exploit the power of IS to extend capabilities and expand business potentials; therefore, experts are challenged to understand the effect of IS on business (Daniels, 1994) and on users (Beynon-Davies, 1998). The visibility and need of IS in business development cannot be over emphasised, and if businesses are to go forward and expand internationally, then, there is need for better usable and secure international communications systems to be developed. There is also the demand for good global communications for engineering and product development. Therefore, the role of IS in influencing how organisation operates is vital. According to Daniels (1994), globalisations of Information Systems is effecting changes to business and accelerating the dynamics of global economy which are destroying traditional concepts of time, geography and strategic advantage. Therefore, innovations, resulting from globalisation, demands new management styles and new approaches to develop and integrate usable secure systems in a global context. Although technologies have so much affected processes, analysts such as Landauer (1995), Bignell and Fortune, (1984) are still of the opinion that, information systems offer operations that do not help users adequately. For example, information systems schemes developed to assist users to find books in libraries make it harder for them to find the books. The IS data manipulation, storage and retrieval through use of structured query language (SQL) has been found to be weak. Despite the fact that the technology was originally implemented to make data retrieval user friendly, but this has never been the case in many situations. Users (such as managers, salespeople etc) are very rarely able to perform tasks themselves, and often rely on system specialist for help. Situations like this, demonstrates the inadequacies in the design, development /implementation and use of technology, which supposedly was designed to make life easy for users. It is interesting to note that, at the beginning of any disruptive innovation, the new technology takes root in areas of non-consumption – where the alternative is nothing at all. Therefore the simpler, new innovation is infinitely better. And more users will adopt it as the disruptive innovation predictably improves. However, there are many instances where

technology has been designed and developed, but simply abandoned due to complicatedness, inaccessibility, and difficulty and/or not easy to use.

For centuries, systems have been designed and developed that are not effective in meeting human needs, even though most business processes are human activity systems (i.e. depends on human activities) and have failed in the process. For example, the London Ambulance Computerised system, Heathrow Air traffic control system, Heathrow Terminal 5 passenger system, and the NHS systems etc. all failed initially due to their inability to meet business and users' needs. We have many car drivers today whose in-car Satellite Navigation systems (SATNAVs) have sent them in the wrong direction or into dead-ends; which have been due to no fault of their own, but as a result of either poor design or inadequate user guide/instructions or feedback provision. Such failures create a gap between technology driven nature of computer science and user requirements. Landauer (1995) had very strongly pointed out that emerging techniques for user-centred development can turn the situation around; through task analysis, iterative design, trial use, and evaluation; computer systems can be made into powerful tools for business and the service economy. There are huge benefit-to-cost ratio that can be achieved through user-centred design (UCD) activities, and backed by descriptions of how to do the necessary things (user guide/instructions), of promising applications for better computer software designs in business. Also UCD can be used to map the relations of user-centred design to business process reengineering (BPR), quality, and management of resources.

2.9 Summary

In this section we have discussed the origins of information systems in an attempt to define what an information system really is; and looking at reasons for the development. The various types of IS and their use in society today have been explored, including the various methodologies involved in the creation of valuable IS. Selected methodologies (SSADM, SSM and DSDM have been explored to determine how their use in developing information system addresses the hard aspects relating to Security and soft aspects in relation to Usability. The Structured Systems Analysis and Design Method (SSADM) is a detailed method, known to cover almost every element of the information system, but with strong emphasis on the hard aspects and very little on soft aspects. Also, we analysed the latest version of the SSADM, which allows the use of SSM in the early phases. The analysis has shown that the

focus of SSADM is on product quality and less user involvement. This very structured and disciplined process can therefore be very good in addressing more of the security aspects and less of the usability aspects. SSM on the other hand, is an approach that tends to have high users involvement at every stage of the process, and as a result focuses more on usability aspects than on the security. In another way, SSM deals with some elements of hard aspects and all of the soft aspects. For example, SSM does not support hard system aspects such as data structures, events and the design of interfaces. Meanwhile DSDM, although good for fast delivery projects and high user involvements (usability aspects), but is not very good for projects that requires complex control mechanisms and/or complicated safety aspects. And, because of its very nature, some of the delivery products from different development teams may not necessarily fit well into the system when combine or integrated as a whole.

Chapter 3 Issues of Usability and Security in Information Systems

3.0 Introduction

Here, we investigate further the origins of IS, whilst also looking at the problems and failures in developing better IS, particularly with the issues of usability and security of the systems. It will cover some of the challenges involved, whilst also investigating some of the models currently used for designing secure and usable systems.

3.1 Background

In the beginning, and according to Landauer (1995), information revolution was meant to replace human mental work with more efficient electronic processes. This statement can be true to a certain extent and in some contexts due to the fact that, computers and robotic technologies have endeavoured to replace humans in carrying out and reducing some of the physically demanding and mundane tasks humans usually performed. Computer science developments has over the years drastically transformed the IS landscape with more sophisticated and emerging derivatives. For example; the evolution of mainframe computing towards client-server architectures; and the improvements in User Interface (UI) from character-based to Graphical User Interface (GUI), and also incorporating multimedia and World Wide Web (WWW) technologies to meet business needs has seen tremendous changes. Early indications of revolution in information systems came about when Bush (1945) published a paper about design of machine-man-systems to augment human memory. Many analysts' and researchers such as Cash and Konsynski (1985) amongst others believe this to be the first milestone towards both Computer Science and information systems academic field development. However, information systems (IS) only became an important issue in the early 1960s when, Langefors (1966) published his "Theoretical Analysis of Information Systems" in which was provided, some clear definition of the fundamental concepts of information systems – to include, Data, Information and realisation of information systems within organisational context with or without computer technology.

Researchers (Benson and Standing, 2002; Bell and Wood-Harper, 1998; Mclean and Swanson, 1980; Clarke, 1990; Ein-Dor and Segve, 1993; Jayaratna, 1994; and Keen, 1996) have all reported that the rapid developments in IS has been as a result of contributions from

many other disciplines such as; Computer science, Management science and Organisation science, used to build-up the foundations of IS. The information systems domain includes both IT related and non-IT related activities and can be contextualised only through organisation context and its associated activities (Jayaratna, 1994). However, such interdisciplinary nature of IS, according to Edwards et al. (1991), is to some extent, as a result of the failures of the computer science to understand users and behaviours (i.e. Human factors), its problems/requirements and the complex nature of operating environment.

In recent years there have been huge increase investments in computers and other information technologies to facilitate and support business and processes (Haag, Cummings and McCubbrey, 2004). However, Landauer, (1995) noted that these increase investments have not been matched by increase productivity in the very service industries which they were aimed at. In many cases, they have either stagnated growth or have made negative impact on business; for example IT has brought about the many causes and/or increase in the in-security in almost every sector of business across the globe. On the one hand, Landauer questioned the reasons for the increase investments by saying, "if computers are not making businesses, organisations, or countries more productive; why then are we spending so much time and money on them?" However, Landauer (1995) acknowledged the fact that other factors (such as mismanagement, hardware and software incompatibilities/failures, organisational barriers, learning curves, social and cultural issues), can also play a part in the productivity paradox. But the main culprit to this demise has been the individual utility and usability of the systems.

3.2 Usability and Security issues in IS

Computers have revolutionised the way we live and operate, and rightly so. For over the past quarter Century there has been enormous growth in IS development and use; and computers have become a feature of everyday life in richer, as well as in poorer economies too. Today's machines are not only powerful, fast and sophisticated, but also complicated. According to Cranor and Garfinkel, (2005), a typical handheld device or desktop now boast possibly ten times the number-crunching power of the fastest machine on earth in 1983, and widespread too, given that the world's 3 billion or so mobile phones are, in effect, pocket computers. Many users have a PC, laptop and/or mobile devices, and they want the computer systems to be very usable and secure. However, although computers have become cheaper, more capable

and more commonplace, they have made much less progress when it comes to ease of use (Cranor and Garfinkel, 2005) of the systems and/or dealing with ever changing security issues (Whitman and Mattord, 2012). Their potential remains surprisingly out of reach for people who find their control systems, or "user interfaces", too complex. And even users, who have no difficulty navigating menus, dialogue boxes and so on, might use computers more productively if their interfaces were better or simple and the security protocols were easy to follow. For example, in using the Nokia 6680 mobile phone, Greenfield (2007) reckons users needed 13 clicks just to change its ringtone and says that "it's an interface designed by engineers for engineers," Other researchers such as Steven Kyffin (a senior researcher at Philips), have conceded that "computer programmers and engineers (himself included), are often guilty of designing complicated systems packed with too many features"; and went on to say that "we're compelled by complexity," but "there's a point where humanity just can't handle it." However, interestingly and very appropriately, the field of interface design even has an unwieldy name: called "human-computer interaction", or HCI; and it's supposed to be an area where systems are to be designed for humans to interact simply with. In terms of usability, systems fail if the desired output is not achieved (Bignell and Fortune (1984); and also any failure in the security system can result to a breach and some consequences.

Greenfield (2007), believes part of the problem is that, programmers have traditionally had more power than designers. Programmers put in place the myriad features they want; interface designers then struggle to wrap them all up in a product that is simple to use. The results, all too often, are crowded clunky interfaces. But the balance of power may now be shifting to the designers. But why now, some may asked? Over the years there have been mounting pressures from many quarters (especially users) on usability and security of systems. Ken Wood, deputy director of Microsoft's research laboratory in Cambridge, England, says his company is now putting greater emphasis on interface design. Three years ago, he says, none of his lab's budget was earmarked for pure HCI research. Today, a quarter of the lab's budget goes on it. Therefore the future is bright for interactivity and thus usability.

3.3 Aspects of Usability

Usability is defined as "The level to which a product can be used by specified product users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context

of use" (Bevan, 1995). "The capability of the software to be understood learned, used and attractive to the user, when used under specified conditions" (ISO/IEC 9126-1, 2001; Veenendaal, 2002). "The ease with which a user can learn to operate, prepares inputs for, and interprets outputs of a system or component" (IEEE Std.610.12-1990). In a nut shell usability is the quality of a product (software or other) which addresses its suitability for the people who will use it. Consideration of usability affects the definition of the requirements, the design and the build of the product, and also testing of the product. Good usability is as vital as good functionality and reliability to delivering products which can be used successfully, are appealing and encourage people to use and re-use the product. In a broader sense, Usability is defined by using the product in its operational environment. The type of users, the tasks to be carried out; physical and social aspects that can be related to the usage of the software products are taken into account.

The narrow focus definition of usability describes usability as a set of attributes that can be measured; these attributes are quality of the software including understandability, learnability, operability and attractiveness:

- Understandability: attributes of the software that bears on the user effort for recognising the logical concept and its applicability
- Learnability: attributes of software that bears on the user effort for operations and operation control
- Operability: attributes of the software that bears on the users effort for operations and operation control
- Attractiveness: the capability of the software to be liked by the user (OpCit)

The broad focus definition of usability is defined using ISO 9241-1 standards. This is described in terms of effectiveness, efficiency and satisfaction with users who can achieve specified goals in particular environment.

Effectiveness, efficiency and satisfaction are explained as follows:

- Effectiveness: Refers to the capability of the software product to enable users to achieve specified goals with accuracy and completeness in specified context of use.
- Efficiency: Refers to the capability of the software product to enable users to achieve results while minimising resource usage.
- Satisfaction refers to the capability of a software product to satisfy users in specified context of use.

Jacob Nielsen (1993) looks at usability from another perspective. The five attributes of usability according to Nielsen (1993) is shown in figure 3.1.

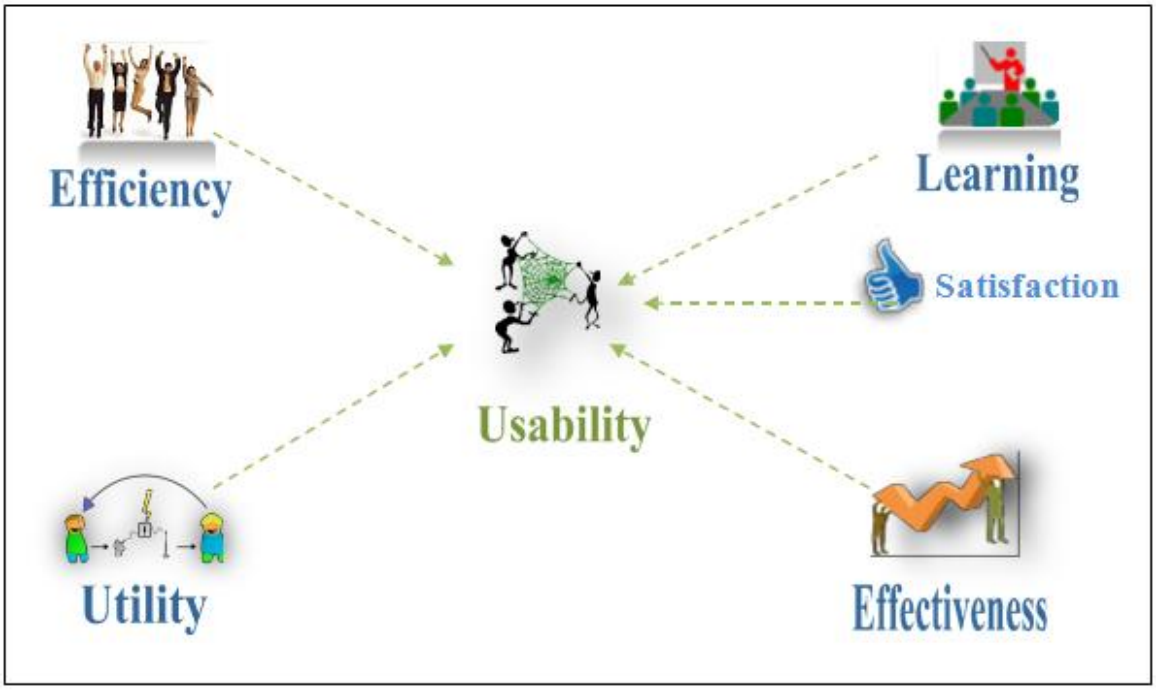


Figure 3.1 Five Attribute of Usability (Source: Norman and Panizzi, 2006).

3.3.1 Efficiency

Efficiency can be described as how quickly system user can perform a task accurately and correctly after the system user learnt the basic operation (Winschiers and Fendler, 2007). Once the system users learn how to use the system, high productivity is expected, so efficiency is related to systems performance (Minati et al., 2006).

3.3.2 Learnability

Learnability has been used as a phase of software usability in 1976 (Grossman et al., 2009). It got fame as important aspect of usability in mid-1990's (Dix et al., 2004). Learnability is among the important aspects of usability (Abran et al., 2003). Different definitions of learnability are available in literature (Petrie et al., 2006).

3.3.3 Errors

A good system should have few rates of errors that help system users to face a small number of mistakes while using the system. When the system users make mistakes or any error occurs, the recovery from mistakes or errors have to be easy, and ensure that catastrophic errors should not occur (Folmer and Bosch, 2004).

3.3.4 Memorability

The concept of memorability states that human actions on a system must be easy to remember. The concept within the usability context is that a user can leave a program and, when he or she returns to it, remember how to do things in it. How many times have we all gone through a training exercise with someone who knew the system only to come back to it and be completely confused? This is the issue that memorability tries to address. The aim of the system must be such that actions performed can be easily remembered as human memory is temporally restricted by an interim ability of roughly seven plus or minus two objects (Yan et al., 2000). For system users if they return to the system after some time and they do not use the system during that time, they don't have to learn the whole system again (Folmer and Bosch, 2004).

3.3.5 Satisfaction

In essence, the system should be satisfying in use; that is, it should fulfil the requirements of system users when they use it (Folmer and Bosch, 2004). In short, users should be happy with the responsiveness and performance of the system.

3.4 Importance of Usability

Over time, frequent changes in technology have become very stressful for people; and most often developers keeps focus on developing the newest products irrespective of products end user's interests and needs. Most often product users are not part of the development process; this creates difficulty for the developers to fulfil the user's expectations. Therefore the main intention of developers must be to develop user-cantered products in order to fulfil the expectations of product users. Modern Software expansion lifecycle is

split into different stages; in previous times usability testing was performed late in the software development lifecycle. However, in the last decade usability testing has become the vital part of development stages particularly for web-based applications (Shneiderman and Plaisant, 2005). In traditional development process, product end-users are not involved in development stages, but by involving end users in development process developers can make product better (Folmer and Bosch, 2004). Usability testing plays a vital role to ensure that the interface design meet the needs of end users. Usability testing has become a vital part in software expansion lifecycle (Dix et al., 2004). Many different events and methods for usability testing have been developed and convinced by many different researches which differ from one another on the basis of significance (Mack and Nielsen, 1993). Usability testing has various purposes or goals. Its most significant goal is to find out the major problems in the user interface of software product. It also has other targets such as to increase performance, efficiency, user satisfaction and also make sure that the system is easy to understand (Norman and Panizzi, 2006). The software engineering community ISO 9126 has related usability with the design of interface. To measure the usability of software there are different standards according to its definition. The research looks at usability by evaluating the usability of some online systems. Usability is evaluated by measuring products end users performance issues. This is due to fact that, mainly usability issues are only exposed late in the development process, during testing and deployment (Battleson et al., 2001). However, if usability is to be maximised, system designers must adhere to and incorporate the 'High Availability' approach – which involves associated service implementation, which ensures a prearranged level of operational performance will be met during a contractual measurement period (Piedad and Hawkins 2001)

Various methods in software and product development are suggested for measuring, efficiency and satisfaction (Nielsen, 2010). Customers, designers, users and stakeholders need to have a solid understanding of requirements for usability and what can be measured for the project. According to Veenendaal (2002) and Nielsen (2010), good usability is crucial in delivering products that can be used successfully, appealing and encouraging people to use and re-use the product. An everyday example of how usability affects a variety of people through the use of ATM (Automatic Teller Machine).Decisions taken into consideration during the Design phase of the software including will affect the usability of the ATM for the users. Examples of ATM's design and installation that will impact usability are the following:

- Colours that are used on the ATM screen
- Font size and font shape used on ATM screen
- The complexity of language used on the screen, this includes the use of jargon.
- Wording which includes the use of help messages.
- Speedy navigation
- Height of the screen and controls located above the ground
- Privacy and safety for the user of the ATM.

Usability has contributed enormously to success of software provision, whether for in-house or third party, bespoke or package systems. Increase in development of intranets and extranets to deliver information, usability is crucial and critical to the success of the application. An Important aspect of usability according to (Nguyen, Johnson, & Hackett, 2003) is Usability testing. Usability testing involves a variety of methods for setting up the product, assigning users to carry out the tasks and observing users interacting and collecting information which will measure ease of use or satisfaction. Usability testing at times might be beyond the scope or responsibilities assigned to the testing group; this however depends on the charter of a software testing organisation.

Usability as used in Quality Assurance and Testing: is a metric that assists the designer of a product or service e.g. Web application, software application and mobile application determine the user's satisfaction through their interaction with a product, application or services through interfaces, including User Interface (UI). An effective UI design is one that offers the maximum usability to the users. In designing for usability, some of the important questions to take into consideration are as follows:

- How easy will it be for a user who has never seen the product or application before to carry out simple tasks?
- How easy will it be for a user who has used the product or application to remember to carry out the same tasks?
- How effective and efficient will it be for a user who is familiar with the product and has used the product before to quickly carry out tasks which are frequent?
- How often does a user run into errors while using the product? How serious are those errors? How forgiving is the product -does the product allow the user to easily recover from errors? How informative is the product or application helpful in communicating error conditions to the user?

- How good is the user's experiencing in using the product

3.5 Aspects of Security

Information systems that are used to capture, create, store, process or distribute classified information must be properly managed to protect against unauthorized disclosure, loss of data integrity, and to ensure the availability of the data and system (Whitman and Mattord, 2012). In short, information security is about protecting a system. Therefore Information systems and the information they contain need to be safe and secure to be relied upon, and appropriate security measures must be put in place to achieve this, but in doing this one need to understand the essence of security. Interestingly, many organisations see technology or 'technical solutions' as the immediate solution to their information security problems. The fact is that, this warp view is highly promoted by the vendors the very same 'technical solutions'. According to Hinson (2003) technology-based information security products such as firewalls, antivirus software, PKI systems and VPNs are very valuable defence mechanisms in the security manager's arsenal but there are many severe drawbacks to rely purely on the technological approach. Here are few examples (Hinson, 2003):

- To start with, the technology does not come cheap, whether bespoke or off-the-shelf. Worse still, the standard packages are often sub-optimal and offer very little in terms of competitive advantage.
- Every technology is fallible even if it is not cheap. Despite the best efforts of the software quality engineering movement, hackers, testers and users continue to find loopholes, unchecked buffers, unexpected exceptions, backdoors and other miscalculated vulnerabilities in the systems. This problem is compounded by the complexity of modern IT systems. Although organisations that employ multilayered security have the right idea but it is very hard to believe that every layer of security is near perfect. But attackers nowadays have never been known to come through the main gate, they bypass defences by taking an alternative approach as is now common on the internet / web.
- Another problem is that very few organisations understand their information security problems in sufficient detail to ensure that they specify appropriate technical solutions. Typically, they recognise the need for standard information security packages (such as antivirus software) to address individual concerns, but seldom have a comprehensive view of their requirements. Most organisations buy 'plug and play'

technologies such as firewalls with no regard to monitoring the security alarms, updating attack signatures, or responding to new forms of network traffic. For example, all they do is, virus-scan "Emails," but ignore the JavaScript.

- The ultimate is that, someone inevitably has to implement and operate the technology and this is where the problem(s) can be found in information systems security.

Recently, there have been many breaches in system security. For example, in April 20 2011, Sony's online Play Station system was hacked and users personal details were stolen (BBC News, April 20 2011) and the system was down for over a week. Users no longer had access and were not able to use the system whilst it was down due to some security vulnerabilities in the system infrastructure. In another incident in 2010, the BBC reported "HSBC admits huge Swiss bank data theft: about 24,000 clients of HSBC's private banking operation in Switzerland had personal details stolen by a former employee, the company has admitted." (BBC News, 11 March, 2010). The BBC has over the years reported many other security breaches such as, "Call centre 'scam' details sought: India's IT industry has urged Britain's Channel 4 television to co-operate with the authorities after a sting alleging data theft from Indian call centres" (BBC, 4 October 2006). Meanwhile in another article on (BBC, November 2009), the UK Information Commissioner's Office put out a statement commenting on the, "Unacceptable level of data loss: The number of incidents of loss or theft of personal data has risen to an "unacceptable" level in the past years, the privacy watchdog has warned. The Information Commissioner's Office (ICO) pointed out that, NHS hospitals holding private medical records were among the worst offenders. In total, 434 organisations reported data security breaches in the past 12 months of 2009; up from 277 the year before." (BBC News 11 November, 2009).

The UK Government defines Information security as: "the practice of ensuring information is only read, heard, changed, broadcast and otherwise used by people who have the right to do so" **(**Source: UK Online for Business, Accessed, December 2010). Security according to Whitman & Mattord (2009) means to be free from danger: protection against adversaries, from those who can do harm intentionally or non-intentionally. For example, National security is a multi-layered system which offers protection for the sovereignty of the state, its assets it's resources and its people. They further defined Information Security (InfoSec) using the standards suggested by the Committee on National Security Systems (CNSS), and which states that; "Information security is the protection of information and elements which

are critical, this includes the systems and hardware that uses, stores, and transmit the information' (Whitman & Mattord, 2009). Information security extends to information security management which includes computer and data security, and network security in order to protect information and related systems.

To protect an organisation's operations, it has been suggested (Schneier, 2000; Whitman & Mattord, 2009) that the system should have the following multiple layers of security:

1. Physical security: is required to protect physical item or, objects from unauthorised access and misuse by users.
2. Personal Security: This is security required to protect users who are authorised to access the organisation and its operations.
3. Operations security: is a type of security which protects series of activities or details of a particular operation.
4. Communication security: is security that protects communications, media, technology and content
5. Network security: is required to protect networking components, connections and content.
6. Information security is required to protect information assets.

But most importantly, the organisation must be willing to implement tools such as awareness, training and education and technology /reviews processes.

Businesses are very much reliant on information systems for key business processes. Therefore it is vitally important that such critical systems are adequately protected and rightly so to allow business continuity. Organisations may face security issues during the information system use. The weaknesses in the information system can be identified while using it, but it can be better if these defects are fixed before any security threat emerges. The security threats while using the system can be in two ways; threat from the external environment (i.e. threat from people outside the organisation) and threats from internal environment (i.e. threat from the people who work in the organisation). Let us now look at some important types of security threats. Information systems can be at security risk from many sources:

- **Human error**: This is one of the most vulnerable areas of risk and has been blamed for most system insecurity. This may occur by entering incorrect transactions; failing to spot and correct errors; processing the wrong information; accidentally deleting

data and/or failure to follow procedures (which may arguably be as a result of unusable security systems designs)

- **Theft and Commercial espionage**: unauthorised access and/or competitors deliberately gaining access to commercially-sensitive data (e.g. customer details; pricing and profit margin data, designs)
- **Fraud**: deliberate attempts to corrupt or amend previously legitimate data and information
- **Technical errors**: such as hardware that fails or software that crashes during transaction processing
- **Accidents and disasters**: may occur through for example, loss, fire or flood
- **Malicious damage**: where an employee or other person deliberately sets out to destroy or damage data and systems (e.g. hackers, creators of viruses)

3.5.1 Security issues in the System Development Process

To begin with the analysis of security issues in system development process, we must start by looking at the phases of system development process themselves. However, not all system development processes have clearly marked phases. For example Service Oriented Architecture (SOA) does not have these phases. System development process such as the Traditional or Waterfall model usually has phases such as; requirement analysis, design, implementation, testing and maintenance (Avison and Fitzgerald, 1995; Akhgar, 2003). The requirements analysis phase is where the requirements of the clients are gathered. Usually business analyst and the project manager meet the client, analyse their business operations by engaging with the employees and in the process, gather the requirements. The requirements gathered are first written in a document call BRS (Business Requirements Specification). Another document called SRS (Software Requirements Specification) is later designed, giving the details required to build the software application. Once BRS and SRS are finalised, the process then move to the next phase of development. Different methodologies do exist, and the use of the phases with them, may differ from one methodology to another.

Most organisations tend to use a method which suits the system they are building. The DSDM (Dynamic Software Development Methodology) is one method that is increasingly gaining importance because of its Rapid application development approach. However, the security requirements during the system development process cannot be tested. Therefore, it

is very essential that, from the start of development (which is requirements analysis), the security features have to be incorporated and aligned with the development process until the completion of system development. Let us now look at some potential risks involved during the development process:

- The requirements gathered may not be 100 per cent correct. This can come from a lack of communication between the development team and the clients.
- The risk of storing or inputting the requirements into the system.
- The designed documents are stored in separate database created for the project by using other applications for quality control. There is a risk that these documents can be misused or tampered with.
- The server may crash because of improper installations or handling, which may result in the loss of valuable data.
- During the implementation phase, there is a chance that some developers may include malicious applications like trap doors, which may cause potential damages to the organisation using the developed system.
- In the testing phase, there is a chance that testers may not create or perform the correct test scenarios, which may result in building a system full with bugs.
- Chances of making improper use (because of lack of proper training) of the system by the client's employees which can result in severe damage to the system.

The issues highlighted above are some of the risk elements that can be analysed during the system development process.

There has been a tremendous rise in the misuse of computers and computer crimes. Some of it is down to poor system development which leave system vulnerable and others from careless employees. Landreth (1989), Hafner & Markoff (1991; 1995), reported that many US companies have been victims of crime. In another study of 283 large companies by the American Bar Association found that, 48 percent of them have been victims of computer crime. Though it has been many decades since, the problem is still the same and it can be much worse today. With the rapid developments in technology, the cybercriminal have now developed new modes of attack using the bugs or weaknesses in the technology. A closer look at the UK Cyber Crime Report 2009 shows that, there is increasing dependence on the internet nowadays, and because many households are connected to internet. The Garlik UK cybercrime report (Fafinski and Minassian, 2009) will be used here to analyse the effect of

different security breaches. As per the report, 70 percent of the UK households had internet access in 2009 and out of which 90 percent were using broadband connection. The increasing dependence on the internet led the fraudsters to deploy new ways of cybercrime. In 2008, there were 3.6 million criminal acts (one for every 10 seconds) identified. There is an increase in fraud in all the categories except sexual offenses. The table 3.1 illustrates some of the incidences:

Category	**2008**	**2007**	**2006**	**Change 07/08**
ID theft and ID fraud	86,900	84,700	92,000	+2.6%
Financial fraud	207,700	203,700	207,000	+1.9%
Online harassment	237,4000	2,240,000	1,944,000	+6.0%
Computer misuse (excluding viruses)	137,600	132,800	144,500	+3.6%
Sexual offences	609,700	617,500	850,000	-1.3%
Total	3,415,900	3,278,700	3,237,500	4.2%

Table 3.1 UK Cybercrime report (Source: Garlik, 2009; Online identity experts group)

Identity theft and Identity fraud cases

ID fraud	**77,642**
Application fraud	**77,023**
Impersonation	**62,658**
Total	**217,323**

Category	**2007**	**2008**	**Change**
Facility takeover fraud	6,272	19,275	+207%
Misuse of facility	23,400	39,447	+69%

Table 3.2 ID fraud cases (Source: Garlik, 2009 online identity experts group)

Table 3.2 shows there was a shocking 207 per cent increase in facility and takeover frauds from the year 2007 to 2008. Another survey of 1000 UK businesses by NCC in 2004 shows the necessity for better risk management approaches in today's business environments. The figures in the report show the ever increasing level of security threats in the Information System environment.

Reported Hardware Incidents	**Percentage**
Equipment Failure	47%
Theft	28%
Network Failure	30%
Sabotage	2%
Fire	2%
Lightening	9%
Flood	5%

Reported Software Incidents	**Percentage**
Viruses	34%
Untested Software	26%
Misuse	9%
User errors	23%

Table 3.3 IS Security threats (Source: NCC, 2004)

The reports demonstrate that there has been sharp increase in the number of security breaches in recent years. So, how do we tackle such increasing threats from unauthorised users, some of whom are people sitting next to us? How can systems vulnerabilities be reduced?

The answers to these questions will be to analyse and improve on the psychological human factors in the use of systems. For example, increase investments in social aspects of work environment such as, increasing the value of staff (pay and conditions), improving trust amongst staff, regular monitoring output, training and lifelong learning processes.

The increase in demand for stolen data has also contributed to increasing crime rate and given criminals a better way of making quick easy money through hacking and sniffing means. And if the trend is not seriously checked, many people may be tempted to choose these routes as a

means to subsidise their incomes. Some security risks can be lessened or prevented by motivating users, encouraging and building trust in them, and also educating them etc. The importance of some of these issues will be highlighted later.

Many analysts and IT professionals in the industry have different opinions as to why there are many security breaches in many organisations' information systems, and have argued that, the security problems emanate from the improper design and development of systems. But it will be improper to put the blame completely on system developers because, the technology and its use, are not the same always. This can be one reason for the increasing security attacks. However, organised and well implemented regular changes to the system can be one way to overcome such security risks; although such changes will increase the overall costs of IT investments, which many organisations are very reluctant to do always.

The two main security issues to be considered during the system development are:

- security regarding the system development resources and,
- Security weaknesses in the system being developed.

The development related resources can be safeguarded and properly managed by using available project management applications such as Quality Centre, PRINCE 2 techniques.

The main problem during the development is the security weaknesses in the system being developed. These issues cannot be tested because these types of bugs come out only during the use of system and not during or while the system is being built, therefore can only be reviewed. However there is a chance that they can be found during the system testing phase but it may not always be a positive result. So the development team have to ensure that they are building the system perfectly secure with respect to the user requirement specifications.

3.5.2 Information System Security

3.5.2.1 Introduction

This section provides analysis in the area of information system security, the related elements and details to facilitate understanding of the discipline. It covers key terms and explaining essential concepts and strategies for managing information security. According to James Anderson, executive consultant at Emagined Security Inc.; Information Security in an enterprise is a "well informed sense of assurance that the information risks and controls are in balance." This basically means aligning information security needs with business objectives

must be top priority. Information systems must be prevented from malicious attack software programs like worms, Trojans and/or viruses. And if the worse happens, the systems should be able to contain such breach and limit the damage caused to the systems and/or organisation.

3.5.2.2 The need for Security

Information security began with Computer security (Whitman and Mattord, 2012); the need to secure hardware, software and physical location from threats. In many situations we sometimes have multiple levels of security implemented to protect facilities and maintain the integrity of their data. For example access to sensitive areas or systems files can be controlled by means of keys, badges, passwords, and/or facial recognition of authorised staff by the security guards. The growing need to maintain infrastructure security has led to more complex and more technologically sophisticated computer security safeguards. According to Whitman and Mattord (2012), information security in the early years was a straight forward process, which consisted predominantly of physical security and simple document classification schemes. The primary threats to security then were physical theft of equipment, espionage against products of the systems, and sabotage. However nowadays, the shift has move from the safety of physical locations and hardware, to include securing the data, limiting random and unauthorised access to data, and also high involvement of personnel from many and different levels of the organisation in matters pertaining to information security. Nowadays, the threats have evolved and they come in various sophisticated forms that can cause maximum damage to any computer or information system. Most threats now come from remote and uncontrolled sources, miles away from the target destination; this is down to the fact that most systems are now in highly interconnected networks. The advent of Internet, www and other mobile/wireless communication systems have internationalised and compounded the information security problems through the interconnected network of networks; wired and wireless (LANs and WANs), and continuous ubiquitous communication systems that have facilitate remote access to information systems anywhere, anytime anyhow (Arreymbi, 2007; Kim and Solomon, 2012; Workman, Phelps and Gathegi, 2013). Hackers such as “Black hats and Grey hats” (Kim and Solomon, 2012), are always on the lookout for loopholes or any system vulnerabilities through which they can gain access (Arreymbi, 2007). And because businesses have to protect their data for competitive advantage, there is increasing demands for adequate security measures to stem the flow of system attacks from

intruders and unauthorised users. In the past few years and more recently, the media around the world have reported many incidences of security breaches that have occurred. For example, the recent News of The World (News Corporation) phone hacking scandals and the Diplomatic cables leaks (BBC News, 2010; 2011) from Wikileaks and other criminal organisations made up of hackers who are dotted around the world and whose sole aim is to intercept and leak out private communications between individuals, businesses and/ or Government agencies. Incidentally, there has been a growing awareness amongst users and the general public, of the need to improve information and information systems security; as well as a realisation that information security is important to national infrastructure and defence. In fact, the growing threats of cyber-attacks and criminality have made businesses and governments more aware of the need to defend critical infrastructures (Workman et al. 2013). The examples here give credence as to why security is vital for the survival of business and government information infrastructures.

The method of design and development of information systems can render them either secure or vulnerable. Vulnerabilities occur due to weaknesses or faults in the system or protection mechanism that leaves them open to attack or damage. For example, flaws in a software or hardware package, an unprotected port or unlocked door. Information systems can be made more secured through various techniques, and designers need to be trying to achieve the following (Arreymbi, 2007; Brown and Stallings, 2008; Whitman and Mattord, 2012):

- **Prevention**: Actions or control measures designed to prevent security errors, breaches and/or accidents, to ensure data integrity; including physical security controls which play a key role in prevention techniques.
- **Detection**: These measures are often combined with prevention controls to ensure effectiveness. To spot when things go wrong is very crucial for example, keeping a log of all attempts to have unauthorised access to a network; detection needs to be done as soon as possible and more particularly if the information is commercially sensitive or used for competitive advantage.
- **Deterrence**: deterrence controls are about discouraging potential security breaches.
- **Data recovery** – The possibility of something going wrong is ever increasing for example, if data is corrupted or hardware breaks down; there is need for continuity, therefore it is important to be able to recover any lost data and information as soon as possible.

According to Whitman & Mattord, (2009; 2012), the definition of security concepts evolved from a concept developed by the National Security Telecommunications and Information Systems Security Committee (NSTISSC) also known as the C.I.A triangle (see figure 3.3) In general, Security is " the quality or state of being secure – to be free from harm or danger" (Whitman and Mattord, 2012). .In other words, it is protection against intentional or unintentional adversaries. The Committee on National Security Systems (CNSS), formerly NSTISSC, defines information security as "...the protection of information and its critical elements, including the systems and hardware that use, store, and transmit that information."

In later discussions, we will see that most security systems are multi-layered, and to achieve the appropriate level of security for any organisation, also require a multifaceted system which will be incorporating physical security, personnel security, operations security, communications security, network security and information security. However, all these systems must operate on the tenets of the CIA security triangle, Therefore, in modelling information security system, one need to approach it from holistic viewpoint and to include the broad areas of information security management, computer and data security and network security as shown in figure 3.2.

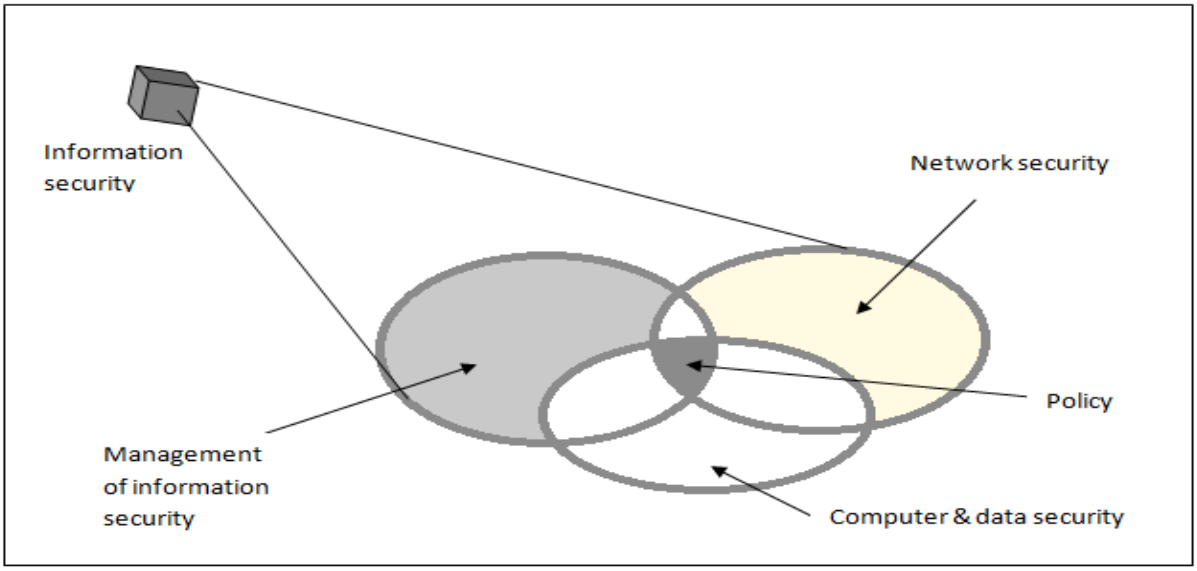


Figure 3.2 Components of Information Security (Source: culled from Whitman and Mattord, 2012)

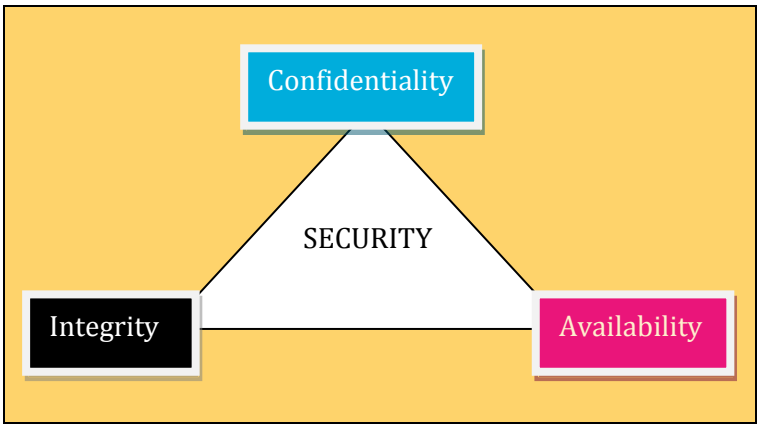


Figure 3.3: The CIA Security triangle

The CIA triangle has been the industry standard for Computer security ever since the development of mainframes; and is based on the three characteristics of Confidentiality, Integrity and Availability that provides value for information in organisations. Somehow, these three characteristics are still important today, but the CIA triangle model no longer addresses the constantly changing environment of IS industry. The new environment of evolving threats has prompted the development of a more robust intellectual model which addresses complexities of current information security environment. The expanded CIA triangle covers authenticity and accuracy (Kim & Solomon, 2012; Whitman & Mattord, 2009; 2012)

- **Confidentiality**: information has confidentiality when disclosure or exposure to unauthorised individuals or systems is prevented. Confidentiality enables users with rights and privileges to have access to information. Confidentiality is breached, when unauthorized users or systems can view information
- **Integrity**: according to (Stallings & Brown, 2008) covers data integrity and system integrity
  - Data integrity: provides assurance that information and programs are altered in a specified and authorised manner
  - System Integrity: Enables a system to perform intended function free from deliberate or unauthorized manipulation of the system.
- **Availability**: This characteristic of information according to Brown & Stallings (2008) enables timely and reliable access to information and use of information; a loss of availability is the disruption of access to or use of the information.

- **Accuracy**: According to Whitman & Mattord (2009), information is accurate when it is free from mistakes or errors and has the value that the end user expects. Information that has been intentionally or unintentionally modified is not accurate e.g. if a user of a bank account accidentally enters an incorrect amount into the account register, this changes the value of the information. Inaccuracy of the bank account can cause a legitimate cheque to bounce, and prompt mistakes such as bouncing a cheque.

From usability point of view this characteristic is much related, in that, if a computer system is inaccurately designed or implemented, it will lead to the users making mistakes / errors in the use or manipulation of the system, such as pressing the wrong button.

- **Authenticity**: authenticity of information is the quality or state of being genuine or original. Information that is fabricated or reproduced is not authentic. An example of non-authenticity is phishing, which happens when an attacker attempts to obtain personal or financial information through a fraudulent way, by presenting a copycat or fake system of an organisation or individual.

System security mostly deals with prevention and protection of the unknown. Computer Security according to Arreymbi (2007); Brown and Stallings (2008), is seen as the protection provided to an automated information system in order to preserve the Integrity, Availability and Confidentiality of IS resources. The emphasis here is on confidentiality, integrity and availability. However, it does not cover accountability and assurance, all of which contribute to maintain/enhance the quality attributes of the system.

- **Confidentiality** confirms that confidential or private information is not revealed to unauthorized users. Confidentiality also covers privacy which assures that users control what information is collected and stored, and to whom and by whom the information is disclosed.
- **Integrity** means that the information is consistent and accurate, and cannot be easily change and/or must maintain its original form.
- **Availability** is the ability for users to access the system, making it readily available to use and fit for purpose.

3.5.3 Implementing some security-protection measures

Researchers Brown & Stallings (2008), Whitman and Mattord (2012), Kim and Solomon, (2012), have reported that, information system protection requires a balanced approach to

include IS security features; and to also include, but not limited to the following: administrative, operational, physical, computer, communications, and personnel controls. Protective measures must be proportionate to the classification of the information; the threats, and the operational requirements associated with the IS environment are required.
Listed here are some of the measures to be identified and implemented to achieve adequate security (http://www.fas.org/sgp; Accessed 10/3/2011):

Protection Profiles: Protection profiles required for a particular IS are determined by the Level of Concern for confidentiality and by the operating environment of the system as reflected by the clearances, access approvals and need-to-know, embodied in the user environment.

Level of Concern: The level of concern reflects the sensitivity of the information and the consequences of the loss of confidentiality, integrity or availability.

a. **Information Sensitivity Matrices.** The matrices presented in Tables 3.4, 3.5 and 3.6 are designed to assist in determining the appropriate protection level for confidentiality, and the level of concern for integrity and availability if contractually mandated for a given IS processing a given set of information.

   The Information Sensitivity Matrices should be used as follows:

   (1) A determination of high, medium or basic shall be made for each of the three attributes: confidentiality, integrity, and availability. It is not necessary for the level of concern to be the same for all attributes of the system.

   (2) When multiple applications on a system result to different levels of concern for the categories of confidentiality, integrity and availability, then the highest level of concern for each category shall be used.

b. **Confidentiality Level of Concern.** In considering confidentiality, the principal question is the necessity for supporting the classification levels and the categories of information (e.g., Secret National Security Information) on the system in question. The Protection Level Table for Confidentiality (Table 3.7) combines the processing environment with the level of concern for confidentiality to provide a Protection

Level. The Protection Level is then applied to Table 3.8 to provide a set of graded requirements to protect the confidentiality of the information on the system.

c. **Integrity Level of Concern.** In considering integrity, the principal question is the necessity for maintaining the integrity of the information on the system in question.
d. **Availability Level of Concern.** In considering availability, the principal consideration is the need for the information on the system in question to be available in a fixed time frame to accomplish an objective.

Protection Level: The protection level of an information system is determined by the relationship between two parameters: first, the clearance levels, formal access approvals, and need-to-know of users; and second, the level of concern based on the classification of the data on a particular system. The protection level translates into a set of requirements (tables 3.8, 3.9 and 3.10) that must be implemented in the resulting system. Table 3.7 presents the criteria for determining the following three protection levels for confidentiality.

a. Systems are operating at Protection Level 1 when all users have all required approvals for access to all information on the system. This means that all users have all required clearances, formal access approvals, and the need-to-know for all information on the IS, i.e. dedicated mode.
b. Systems are operating at Protection Level 2 when all users have all required clearances, and all required formal access approvals, but at least one user lacks the need-to-know for some of the information on the system, i.e. a system high mode.
c. Systems are operating at Protection Level 3 when all users have all required clearances, but at least one user lacks formal access approval for some of the information on the system, i.e. compartmented mode.

Protection Profiles: Protection requirements are graded by levels of concern and confidentiality protection levels. Tables 3.8, 3.9 and 3.10 present the requirements as detailed. (See column representing the protection level for confidentiality).

a. **Confidentiality Components:** Confidentiality components describes the confidentiality protection requirements that must be implemented in an IS using the profile. The confidentiality protection requirements are graded according to the confidentiality protection levels.

b. **Integrity Components:** Integrity components, if applicable, describes the integrity protection requirements that must be implemented in an IS using the profile. The integrity protection requirements are graded according to the integrity level of concern.
c. **Availability Components:** Availability components, if applicable, describes the availability protection requirements that must be implemented in an IS using the profile. The availability protection requirements are graded according to the availability level of concern.

As has been mentioned earlier on, the matrices presented in Tables 3.4, 3.5 and 3.6 are designed to assist in determining the appropriate protection level for confidentiality, and the level of concern for integrity, and availability, if contractually mandated, for a given IS processing a given set of information.

Level of Concern	Qualifiers
High	TOP SECRET and SECRET Restricted Data
Medium	SECRET SECRET Restricted Data
Basic	CONFIDENTIAL

Table 3.4 Information Sensitivity Matrix for Confidentiality

Level of Concern	Qualifiers
High	Absolute accuracy required for mission accomplishment; or loss of life might result from loss of integrity; or loss of integrity will have an adverse effect on national-level interests; or loss of integrity will have an adverse effect on confidentiality.
Medium	High degree of accuracy required for mission accomplishment, but not absolute; or bodily injury might result from loss of integrity; or loss of integrity will have an adverse effect on organisational-level interests.
Basic	Reasonable degree of accuracy required for mission accomplishment.

Table 3.5 Information Sensitivity Matrix for Integrity

Level of Concern	Qualifiers
High	Information must always be available upon request, with no tolerance for delay; or loss of life might result from loss of availability; or loss of availability will have an adverse effect on national-level interests; or loss of availability will have an adverse effect on confidentiality.
Medium	Information must be readily available with minimum tolerance for delay; or bodily injury might result from loss of availability; or loss of availability will have an adverse effect on organisational-level interests.
Basic	Information must be available with flexible tolerance for delay.

Table 3.6 Information Sensitivity Matrix for Availability

(NOTE: In this context, "High - no tolerance for delay" means no delay; "Medium - minimum tolerance for delay" means a delay of seconds to hours; and "Basic - flexible tolerance for delay" means a delay of days to weeks).

Level of Concern	Lowest Clearance	Formal Access Approval	Need-To-Know	Protection Level
High, Medium, or Basic	At Least Equal to Highest Data	NOT ALL Users Have ALL	Not contributing to the decision	3
High, Medium, or Basic	At Least Equal to Highest Data	ALL Users Have ALL	NOT ALL Users Have ALL	2
High, Medium, or Basic	At Least Equal to Highest Data	ALL Users Have ALL	ALL Users Have ALL	1

Table 3.7 Protection Level Table for Confidentiality

Confidentiality	Protection Level		
Requirements	PL 1	PL 2	PL 3
Audit Capability	Audit 1	Audit 2	Audit 3 Audit 4
Data Transmission	Trans 1	Trans 1	Trans 1
Access Controls	Access 1	Access 2	Access 3
Identification & Authentication	I&A 1	I&A 2,3,4	I&A2,4,5
Resource Control		ResrcCtrl 1	ResrcCtrl 1
Session Controls	SessCtrl 1	SessCtrl 2	SessCtrl 2
Security Documentation	Doc 1	Doc 1	Doc 1
Separation of Functions			Separation
System Recovery	SR 1	SR 1	SR 1
System Assurance	SysAssur 1	SysAssur 1	SysAssur 2
Security Testing	Test 1	Test 2	Test 3

Table 3.8 Protection Profile Table for Confidentiality

Integrity	Level of Concern		
Requirements	Basic	Medium	High
Audit Capability	Audit 1	Audit 2	Audit 3
Backup and Restoration of Data	Backup 1	Backup 2	Backup 3
Changes to Data		Integrity 1	Integrity 2
System Assurance		SysAssur 1	SysAssur 2
Security Testing	Test 1	Test 2	Test 3

Table 3.9 Protection Profile Table for Integrity

Availability	Level of Concern		
Requirements	Basic	Medium	High
Alternate Power Source		Power 1	Power 2
Backup and Restoration of Data	Backup 1	Backup 2	Backup 3

Table 3.10 Protection Profile Table for Availability

(Tables culled from: http://www.fas.org/ Accessed, 25/03/11)

3.5.4 Protection Requirements for enhanced usable security

The implementation requirements for the different protection measures are highlighted as follows:

1. Alternate Power Source (APS): An alternate power source ensures that the system availability is maintained in the event of a loss of primary power. An APS can also provide a time period for orderly system shutdown or the transfer of system operations to another system or power source.

a. **Power 1 Requirements -** Procedures for the graceful shutdown of the system shall ensure no loss of data. The decision not to use an alternate source of power, such as an uninterruptible power supply (UPS) for the system, shall be documented.
b. **Power 2 Requirements-** Instead of Power 1, procedures for transfer of the system to another power source shall ensure that the transfer is completed seamlessly within the time requirements of the application(s) on the system.

2. Audit Capability: Security auditing involves recognising, recording, storing, and analysing information related to security-relevant activities. The audit records can be used to determine which activities occurred and which user or process was responsible for them.

a. **Audit 1 Requirements**

(1) Automated Audit Trail Creation: The system shall automatically create and maintain an audit trail or log (On a PL-1 system only: In the event that the Operating System cannot provide an automated audit capability, an alternative method of accountability for

user activities on the system shall be developed and documented.) Audit records shall be created to record the following:

(a) Enough information to determine the date and time of action (e.g. common network time), the system locale of the action, the system entity that initiated or completed the action, the resources involved, and the action involved.

(b) Successful and unsuccessful log-on(s) and log-off(s).

(c) Successful and unsuccessful accesses to security-relevant objects and directories, including creation, open, close, modification, and deletion.

(d) Changes in user authenticators.

(e) The blocking or blacklisting of a user ID, terminal, or access port and the reason for the action.

(f) Denial of access resulting from an excessive number of unsuccessful logon attempts.

(2) Audit Trail Protection - The contents of audit trails shall be protected against unauthorized access, modification, or deletion.

(3) Audit Trail Analysis - Audit analysis and reporting shall be scheduled, and performed. Security relevant events shall be documented and reported. The frequency of the review shall be at least weekly and shall be documented in the SSP.

(4) Audit Record Retention - Audit records shall be retained for at least one review cycle or as required by the CSA.

b. **Audit 2 Requirements:** In addition to Audit 1

(1) Individual accountability (i.e. unique identification of each user and association of that identity with all auditable actions taken by that individual) and periodic testing of the security position of the IS.

c. **Audit 3 Requirements:** In addition to Audit 2

(1) Automated Audit Analysis. Audit analysis and reporting using automated tools shall be scheduled and performed.

d. **Audit 4 Requirements:** In addition to Audit 3

(1) An audit trail, created and maintained by the IS, that is capable of recording changes to mechanism's list of user formal access permissions.

3. Backup and Restoration of Data (Backup): The regular backup of information is necessary to ensure that users have continuing access to the information. The periodic checking of backup inventory and testing of the ability to restore information validates that the overall backup process is working.

a. **Backup 1 Requirements:**

(1) Backup Procedures - Procedures for the regular backup of all essential and security-relevant information, including software tables and settings, such as router tables, software, and documentation, shall be documented.

(2) Backup Frequency- The frequency of backups shall be defined and documented in the backup procedures.

b. **Backup 2 Requirements:** In addition to Backup 1

(1) Backup Media Storage - Media containing backup files and backup documentation shall be stored at another location, such as another part of the same building, a nearby building, or off facility, so as to reduce the possibility that a common occurrence can eliminate the on-facility backup data and the off-facility backup data.

(2) Verification of Backup Procedures - Backup procedures shall be periodically verified.

c. **Backup 3 Requirements:** In addition to Backup 2

(1) Information Restoration Testing. Incremental and complete restoration of information from backup media shall be tested on an annual basis.

4. Changes to Data (Integrity): The control of changes to data includes deterring, detecting, and reporting of successful and unsuccessful attempts to change data. Control of changes to data may range from simply detecting a change attempt to the ability to ensure that only authorized changes are allowed.

a. **Integrity 1 Requirements**

(1) Change Procedures - Procedures and technical system features shall be implemented to ensure that changes to the data and IS software are executed only by authorized personnel or processes.

b. **Integrity 2 Requirements:** In addition to Integrity 1

(1) Transaction Log - A transaction log, protected from unauthorized changes, shall be available to allow the immediate correction of unauthorized data and IS software changes and the off-line verification of all changes at all times.

5. Data Transmission (Trans): Information protection is required whenever classified information is to be transmitted through areas or components where individuals not authorized to have access to the information may have unescorted physical or uncontrolled electronic access to the information or communications media (e.g., outside the system perimeter).

a. **Trans 1 Requirements**

(1) Protections - One or more of the following protections shall be used.

(a) Information distributed only within an area approved for open storage of the information.

(b) National Security Agency (NSA)-approved encryption mechanisms appropriate for the encryption of classified information.

(c) Protected Distribution System.

6. Access Controls (Access): The IS shall store and preserve the integrity of the sensitivity of all information internal to the IS.

a. **Access 1 Requirements**

(1) Denial of physical access by unauthorized individuals unless under constant supervision of technically qualified, authorized personnel.

b. **Access 2 Requirements:** In addition to Access 1

(1) Discretionary access controls shall be provided. A system has implemented discretionary access controls when the security support structure defines and controls access between named users and named objects (e.g., files and programs) in the system. The discretionary access control policy includes administrative procedures to support the policy and its mechanisms.

c. **Access 3 Requirements:** In addition to Access 2

(1) Some process or mechanism that allows users (or processes acting on their behalf) to determine the formal access approvals granted to another user.

(2) Some process or mechanism that allows users (or processes acting on their behalf) to determine the sensitivity level of data.

7. Identification and Authentication (I&A).

a. **I&A 1Requirements -** Procedures that include provisions for uniquely identifying and authenticating the users. Procedures can be external to the IS (e.g., procedural or physical controls) or internal to the IS (i.e., technical). Electronic means shall be employed where technically feasible.

b. **I&A 2 Requirements:** In addition to I&A 1

(1) An I&A management mechanism that ensures a unique identifier for each user and that associates that identifier with all auditable actions taken by the user. The following must be specified in the SSP:

(i) Initial authenticator content and administrative procedures for initial authenticator distribution.

(ii) Individual and Group Authenticators - Group authenticators may only be used in conjunction with an individual/unique authenticator, that is, individuals must be authenticated with an individual authenticator prior to use of a group authenticator.

(iii) Length, composition and generation of authenticators.

(iv) Change processes (periodic and in case of compromise.

(v) Aging of static authenticators (i.e. not one-time passwords or biometric patterns).

(vi) History of authenticator changes, with assurance of non-replication of individual authenticators.

(vii) Protection of authenticators.

c. **I&A 3 Requirements:** In addition to I&A 2

(1) Access to the IS by privileged users who either reside outside of the IS's perimeter or whose communications traverse data links that are outside the IS's perimeter shall require the use of strong authentication (i.e. I&A technique that is resistant to replay attacks).

d. **I&A 4 Requirements:** In those instances where the means of authentication is user-specified passwords, the system analyst shall employ automated tools to validate that the passwords are sufficiently strong to resist cracking and other attacks intended to discover the user's password.
e. **I&A 5 Requirements:** In those instances where the users are remotely accessing the IS, the users shall employ a strong authentication mechanism.

8. Resource Control (ResrcCtrl). The system shall ensure that resources contain no residual data before being assigned, allocated, or reallocated.

9. Session Controls (SessCtrl). Session controls are requirements, over and above identification and authentication, for controlling the establishment of a user's session.

a. **SessCtrl 1 Requirements:**

(1) User Notification - All users shall be notified prior to gaining access to a system that system usage is monitored, recorded, and subject to audit. The user shall also be advised that, by using the system, he/she has granted consent to such monitoring and recording. The user shall also be advised that unauthorised use is prohibited and subject to criminal and civil penalties. If the operating system permits, each initial screen (displayed before user logon) shall contain a warning text to the user and the user shall be required to take positive action to remove the notice from the screen (monitoring and recording, such as collection and analysis of audit trail information, shall be performed). An approved banner will be provided. If it is not possible to provide an "initial screen" warning notice, other methods of notification shall be developed and approved.

(2) Successive Logon Attempts - If the operating system provides the capability, successive logon attempts shall be controlled as follows:

(a) By denying access after multiple (maximum of five) consecutive unsuccessful attempts on the same user ID.

(b) By limiting the number of access attempts in a specified time period.

(c) By the use of a time delay control system.

(d) By other such methods, subject to approval.

(3) System Entry- The system shall grant system entry only in accordance with the conditions associated with the authenticated user's profile. If no explicit entry conditions are defined, the default shall prohibit all remote activities, such as remote logons and anonymous file access.

b. **SessCtrl 2 Requirements:** In addition to SessCtrl 1

(1) Multiple Logon Control - If the IS supports multiple logon sessions for each user ID or account, the IS shall provide a protected capability to control the number of logon sessions for each user ID, account, or specific port of entry. The IS default shall be a single logon session.

(2) User Inactivity - The IS shall detect an interval of user inactivity, such as no keyboard entries, and shall disable any future user activity until the user re-establishes the correct identity with a valid authenticator. The inactivity time period and restart requirements shall be documented in the SSP.

(3). Logon Notification - If the operating system provides the capability, the user shall be notified upon successful logon of: the date and time of the user's last logon; the location of the user (as can best be determined) at last logon; and the number of unsuccessful logon attempts using this user ID since the last successful logon. This notice shall require positive action by the user to remove the notice from the screen.

10. Security Documentation (Doc): Security documentation includes all descriptions of the security features, design descriptions of security-relevant software and hardware, certification packages, and system security plans (Kim and Solomon, 2012). The SSP is the basic system protection document and evidence that the proposed system, or update to an existing system, meets the protection profile requirements. The SSP is used throughout the certification and approval process and serves for the lifetime of the system as the formal record of the system and its environment as approved for operation. The SSP also serves as the basis for inspections of the system. Information common to several systems at a facility or information contained in other documents may be attached to or referenced in the SSP.

a. **Doc 1 Requirements**

(1) SSP - The SSP shall contain the following:

(a) System Identification

1. Security Personnel: The name, location, and phone number of the responsible system owner (e.g. CSA, ISSM, and ISSO).

2. Description: A brief narrative description of the system or network mission or purpose and architecture, including sub-networks, communications devices, and protocols.

(b) System Requirements Specification (see tables 3.5 to 3.10)

3. Sensitivity and Classification Levels - The sensitivity or classification levels, and categories of all information on the system and clearance, formal access approval and need-to-know of IS users.

4. Levels of Concern for Confidentiality, Integrity, and Availability - The confidentiality level of concern and protection level, the integrity level of concern, and the availability level of concern.

5. Protection Measures - Identify protection measures and how they are being met.

6. Variances from Protection Measure Requirements. A description of any approved variances from protection measures. A copy of the approval documentation shall be attached to the SSP.

(c) System-Specific Risks and Vulnerabilities - A description of the risk assessment of any threats or vulnerabilities unique to the system. If there are no threats or vulnerabilities unique to the facility or system, a statement to that effect shall be entered. If vulnerabilities are identified by the assessment of unique threats, the countermeasures implemented to mitigate the vulnerabilities shall be described.

(d) System Configuration - A brief description of the system architecture, including a block diagram of the components that show the interconnections between the components and any connections to other systems, and an information flow diagram.

(e) Connections to Separately Accredited Networks and Systems - If connections to other systems exist, a memorandum of understanding is necessary if the systems are approved by a person responsible for this system. A copy of any memoranda of understanding with other agencies shall be attached to the SSP.

(f) Security Support Structure - A brief description of the security support structure including all controlled interfaces, their interconnection criteria, and security requirements.

(2) Certification and Accreditation Documentation.

(a) Security Testing - Test plans, procedures, and test reports including risk assessment.

(b) Documentation - The test plan for on-going testing and the frequency of such testing shall be documented in the SSP.

(c) Certification - A certification statement that the system complies with the requirements of the protection level and levels of concern for this system. The statement shall be signed by an approved person (e.g. ISSM).

(d) Accreditation - Documentation for accreditation includes the certification package. An authorised person approves the package and provides accreditation documentation.

11. Separation of Function Requirements (Separation): At Protection Level 3 the functions of the ISSO and the system manager shall not be performed by the same person.

12. System Recovery (SR): System recovery addresses the functions that respond to failures in the SSS or interruptions in operation. Recovery actions ensure that the SSS is returned to a condition where all security-relevant functions are operational or system operation is suspended.

a) SR 1 Requirements - Procedures and IS features shall be implemented to ensure that IS recovery is done in a controlled manner. If any off-normal conditions arise during recovery, the IS shall be accessible only via terminals monitored by the ISSO or his /her designee, or via the IS console.

13. System Assurance (SysAssur): System assurance includes those components of a system (hardware, software, firmware, and communications) that are essential to maintaining the security policy(ies) of the system, (e.g. Security Support Structure).

a. **SysAssur 1 Requirements**

(1) Access to Protection Functions - Access to hardware/software/firmware that performs systems or security functions shall be limited to authorized personnel.

b. **SysAssur 2 Requirements:** In addition to SysAssur1

(1) Protection Documentation - The protections and provisions of the SysAssur shall be documented.

(2) Periodic Validation of SysAssur - Features and procedures shall exist to periodically validate the correct operation of the hardware, firmware, and software elements of the SSS and shall be documented in the SSP.

d. **SysAssur 3 Requirements:** In addition to SysAssur2

(1) SSS Isolation - The SSS shall maintain a domain for its own execution that protects it from external interference and tampering (e.g., by reading or modifying its code and data structures).

14. Security Testing (Test): Certification and ongoing security testing are the verification of correct operation of the protection measures in a system. The ISSM will perform and document the required tests.

a. **Test 1 Requirements:** Assurance shall be provided to the CSA that the system operates in accordance with the approved SSP and that the security features, including access controls and configuration management, are implemented and operational.
b. **Test 2 Requirements:** In addition to Test1

(1) Written assurance shall be provided to the CSA that the IS operates in accordance with the approved SSP, and that the security features, including access controls, configuration management and discretionary access controls, are implemented and operational.

c. **Test 3 Requirements:** In addition to Test2

(1) Certification testing shall be conducted including verification that the features and assurances required for the Protection Level are functional.

(a) A test plan and procedures shall be developed and shall include:

1. A detailed description of the manner in which the system's Security Support Structure meets the technical requirements for the Protection Levels and Levels-of-Concern for integrity and availability.

2. A detailed description of the assurances that have been implemented, and how this implementation will be verified.

3. An outline of the inspection and test procedures used to verify this compliance.

15. Disaster Recovery Planning: If disaster recovery planning is contractually mandated, the ISSM will develop a plan that identifies the facility's mission essential applications and information, procedures for the backup of all essential information and software on a regular basis, and testing procedures (http://www.fas.org/sgp/library/nispom/change_ch8.htm, Accessed 25/03/2011).

3.6 Overview of computer security models

Computer security models are formal models of computer security that can be used in verification of security designs and implementation (Stallings & Brown, 2008; 2012). Aspinall (2008) outlines the difference between a security model and a security policy. A security policy describes system requirements for implementing security. It defines goals and elements of an organisation computer system. A security model is a way of formalising an Information system or computer security policy.

Information system specialists explain that there are two distinct meaning of security model in security literature (Mclean et al., 2002). The limited use of security model specifies a particular mechanism used in enforcing confidentiality through access control which was introduced into computer security from the world of documents and safes. The general usage of security models are specification of system's requirement, in this context, though they are given the name security model, they are not model because they specify requirements without describing the mechanism required for implementing the requirements. The models are used to specify restrictions on a system interface; this ensures those implementations which are able to satisfy the restriction will enforce confidentiality.

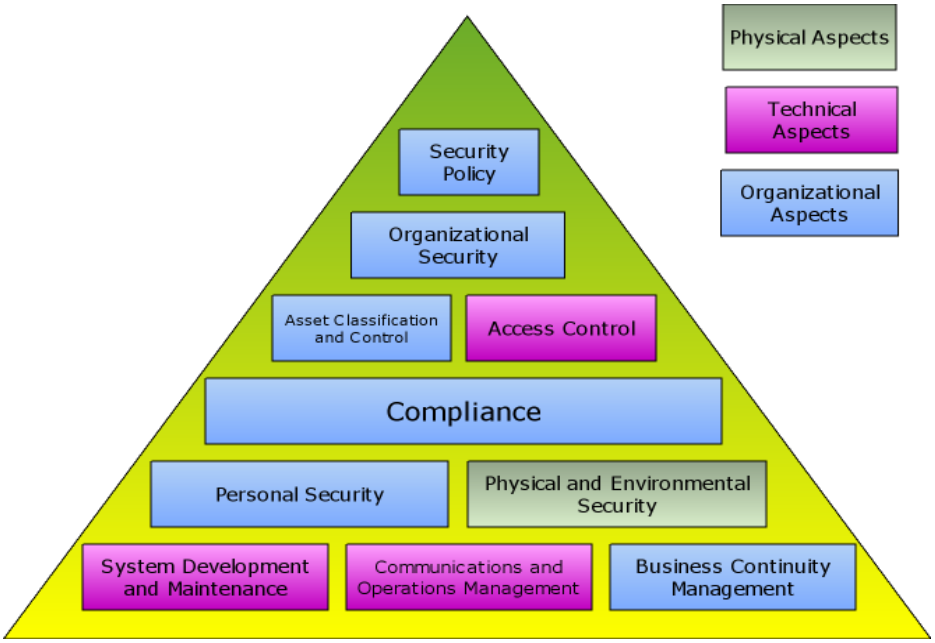


Figure 3.4 YSI security Standard: A security model (Parallels, 2009)

The figure 3.4 is YSI security standard, which follows a simple security model. Parallel (2009) suggests that instead of changing system values, network user attributes based on a magazine article or a book, an organisation should follow a simple security process as outlined in the diagram and includes the following:

- Define a security program: the organisation security policy outlines the basis of the security program. The security program implemented by YPsilon includes procedures, documents, standards, compliance enforcement measures, training and personnel, and software.
- Implement the security program, a security policies combines the policies required by senior management with regulatory policy requirements: depending on the organisation's location and industry, the addition of regulatory content to a security policy may include detailed outline policy information as required by a government, industry or legal requirement.
- Monitor Compliance with the security program
- Obtain independent confirmation that the security program is sufficient and implemented. YSI security tools include out of the box policies such as GSD331 standard, which is BS7739 equivalent of the security guidelines applied in corporate

organisations. Verifying continuous policy compliance with security standard is critical in enforcing highest level of security.

3.6.1 Security Models

There are many types of security models (Stallings and Brown, 2008; 2012), the following are representative model examples, and which are very widely used.

- Bell-la-Padula Model (BLP)
- Biba Integrity Model (BIM)
- Clark–Wilson Integrity Model (CWIM)
- Chinese Wall Model (CWM)
- Role based Access Control Model (RBAC)

3.6.1.1 Bell-La-Padula Model (BLP)

The Bell-La-Padula Model (BLP) was developed as a formal model for access control. It makes use of a security class; each subject and each object is assigned a security class. Security classes form a hierarchy and are classified as security levels (see tables 3.4, 3.7 and 3.8). The US military classification scheme uses the following security levels:

Top secret > Secret > Confidential > Restricted > Unclassified

In any security level, a set of categories or compartments can be added. When added, a subject must be assigned both the appropriate level and category in order to assess an object. BLP groups information into gross levels and categories; it is possible for users to be granted access to specific data categories; for example, corporate planning may be the highest security level which is allocated for strategic corporate planning, and data accessible by corporate officers and staff. The next category is sensitive and financial data; this category is accessible only by, for example administration personnel and corporate officers. It can be seen to follow the sensitivity matrix for confidentiality as demonstrated earlier (see table 3.4).

The following suggests a classification scheme:

Strategic > Sensitive > confidential > Public

The security classes control how subjects may access an object. The Model defines four access modes, and in specific implementation environment, different set of modes might be in use.

The modes are as follows:

- **Read**: In Read-mode, the subject is allowed only read access to the object
- **Append**: In Append-mode, the subject is allowed only write access to the object.
- **Write**: in Write-mode the subject is allowed to read and write access to the object
- **Execute**: The subject in execute mode is allowed neither read nor write access to the object, but is allowed to invoke the object for execution.

Stallings & Brown, (2008) pointed out that when multiple categories of data are defined, the requirement is known as multilevel security. In a confidential-centred multilevel security, a subject at a high level is not allowed to disclose information to a lower level subject unless the flow reflects accurately the will of an authorized user as revealed by a declassification that is authorised. The requirement therefore, is in two parts for purpose of implementation.

A multilevel secure system for confidentiality must enforce the following:

- **No read up**: In No read up multilevel security, a subject can only read an object of less or equal security level this is known as simple security property (ss-property)
- **No write down**: In the multilevel security, a subject is only allowed to write into an object of greater or equal security level. This is known as *-Property (pronounced star property)

The two properties outlined above offer confidentiality form of Mandatory Access Control (MAC). In MAC access is not allowed, when the two properties are not satisfied.

Figure 3.5 Multilevel Security (MLS). (Illustrated by Red Hat (R) Inc. (n. d.))

The classification in figure 3.5 evolves from the defence community‘s security classification; which states that, “individuals must be granted appropriate clearances before they can view information. Those with confidence clearance are only authorized to view confidential documents; they are not trusted to view secret or top secret information.” (See NIST Special Publication 800-18 Rev 1)

3.6.1.2 Limitations to the BLP Model

According to Stallings & Brown (2008; 2012) the BLP model in theory laid the foundation for secure computing within a single-administration realm environment. However, there are important limitations to its usability and difficulties to its implementation. In the model there is incompatibility of confidentiality and integrity within a single MLS system. The MLS system can either work for powers or security, but not for both. This mutual exclusion prevents interesting power and integrity centred technology from being implemented effectively in BLP MLS environment.

Another limitation to usability in this model is known as cooperating conspirator problem which is in the existence of convert channels. In the existence of shared resources the *-Property may not be enforceable and a malicious document can carry in it, a subject that when executed will broadcast classified documents using shared-resource convert channels. The BLP model breaks down when non-trusted executable data are allowed to be executed by a high clearance subject.

3.6.2 The Biba Integrity Model (BIM) (Biba, 1977)

The Biba Model according to Krause & Tipton (1997), was the first model proposed to address integrity in computer systems. It was defined on a hierarchical lattice of integrity as proposed by Biba in 1977; and demonstrated in tables 3.5 and 3.9. The integrity model is identical to Bell-la-Padula Model for confidentiality. It employs subjects and objects, and control object modification in the same manner that Bell-la-Padula controls disclosure.

The Model is divided into three parts: the first part suggest that, a subject cannot execute objects that have a lower level of integrity than the subject. The second part of the model states that a subject cannot modify objects that have a higher level of integrity. And the third part states that a subject may not request source from subjects that have a higher integrity level. In explaining Biba Integrity model (Stallings & Brown, 2008, 2012) suggested that the model is intended to deal with situation in which there is data that must be visible to users at multiple or all security levels but can only be modified in controlled ways by authorized agents. The model emphasises on access-mode, outlined as follows:

- **Modify** : Involves to write or update information in an object
- **Observe**: In observe mode, information is read in an object
- **Execute**: To execute an object
- **Invoke**: This mode deals with communication between one subject to another.

3.6.2.1 Limitations of Biba Integrity Model (BIM)

Despite the benefit of Biba integrity model, there are limitations affecting its practical use in real life. One of the criticisms of the model according to Karger, Austel & Toll (2000) is that, Biba does not model any practical system. This can affect the usability, as the system will not be fit for use. Unlike other security models such as BLP, which evolves from military security systems, Biba integrity model (BIM) is developed from mathematical analyses of the security models. BIM does not suggest how to actually decide which programs deserved a high integrity access, and which were not. Therefore, it has made practical implementation of the model very difficult and challenging. Another downside of the model with regards to usability is that, implementation of the model requires the use of trusted processes to meet different administrative and down grading requirements. This means that trusted processes have been allowed to violate the requirements of the model.

3.6.3 Clark-Wilson Integrity Model (CWIM)

The Clark-Wilson model (CWIM) mainly targets commercial applications and closely models real commercial operations according to Stallings & Brown, (2008; 2012).

The Model uses two concepts to enforce commercial security policies:

- Well-formed transactions: A user is not allowed to manipulate data arbitrarily. Data can only be manipulated in a constrained manner that preserves or ensures integrity of data.
- Separation of duty among users: Individuals permitted to create or certify a well-performed transaction may not be permitted to execute it.

CWIM enforces integrity controls on data and transactions which manipulates the data. The main components of the model are as follows:

1. Constrained data items (CDIs): They are subject to strict integrity controls
2. Unconstrained data items (UDIs): It imposes integrity controls on unchecked data items e.g. a simple text file.
3. Integrity verification procedures (IVPs): Checks that all CDIs adapt to some application-specific model of integrity and consistency.
4. Transformation Procedures (TPs): Refers to system transactions that modify the set of CDIs from a consistent state to another.

3.6.4 The Chinese wall Model (CwM)

The concept of Chinese wall model as a security model is that, individuals are allowed access to information which is not held to conflict with other information they possess. As far as the information system is concerned, the only information owned by a user must be information that is held on the computer. For example, a company has the following datasets as shown in figure 3.6.

- Bank-A
- Oil Company-A
- Oil Company-B

A new user is free to access whatever datasets he likes, he does not own any information, therefore no conflict can exist. A user accesses the oil company-A dataset first, later the user

requests access to Bank-A dataset, this is permitted because Bank-A and Oil Company-A datasets belongs to different conflict of interest classes, and as a result no conflict exists. If the user request access to Oil Company-B dataset, the request must be denied, since a conflict exist between the required data set (oil-Company-B) and one already possessed (Oil Company-A)

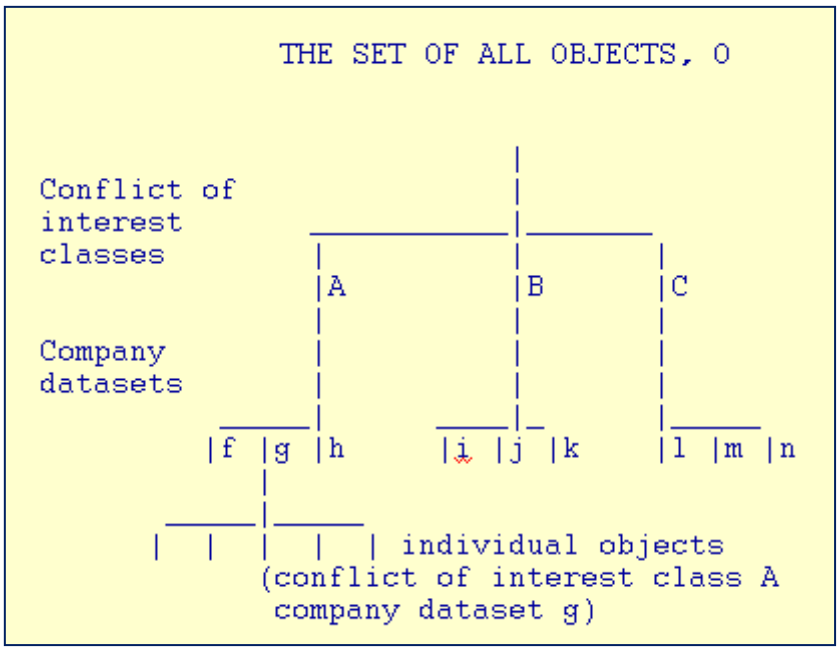


Figure 3.6 illustration of the Chinese-Wall (CwM) model (Stallings & Brown, 2008)

This CWM is a model commonly used in the financial and legal professions to prevent conflict of interest. An example of CWM in the financial world is that of a market analyst who works for a financial institution. An Analyst is not allowed to provide advice to one company when the analyst has confidential information (inside knowledge) about the plans of a competitor. Nonetheless, the analyst is free to advise multiple Corporations who are not in competition with each other and to draw on market information which is opened to the public.

3.6.4.1 Limitations of CwM

Locasto et al, (n.d.) believes that the model relies on assumption the user is only permitted to execute specific set of programs (TPs). The system should ensure that it is not possible for a user to augment the set of programs to pass the SOD rules: The concept of emergent properties makes the assumption difficult to guarantee because future configurations of the

system may contain programs that may be combined in unexpected ways. The requirement is unrealistic for current and future software systems. The model relies on the notion of authenticated principles with roles that are non-overlapping for authorisation. This requirement has the most challenging effect on security, since the notion relies on certification rules. Translating the model to real software is a challenge, because the complexity of modern computer systems, threatens to violate many of the fundamental security expectations.

3.6.5 Role Based Access Control (RBAC) Model

This model is one which highly impacts on usability of IS and tend to hinder users a lot in their daily operational tasks. In organisations, roles are assigned for various job functions. There are different levels of job roles in an organisation. Assigning their roles, powers and controls is also an essential task in the organisation.

The rules in RBAC model are:

- Role assignment: A subject can execute a transaction only if the subject has selected or been assigned a role.
  AR(*s*: *subject*) = {*the active role for subject s*}.
- Role authorisation: A subject's active role must be authorized for the subject. With rule 1 above, this rule ensures that users can take on only roles for which they are authorized.
  RA(*s*: *subject*) = {*authorized roles for subject s*}.
- Transaction authorisation: A subject can execute a transaction only if the transaction is authorized for the subject's active role. With rules 1 and 2, this rule ensures that users can execute only transactions for which they are authorized.
  TA(*r*: *role*) = {*transactions authorized for role r*}.

Ferraiolo and Kuhn (1992), defines the conventions used in this model as:

- *S = Subject = A person or automated agent.*
- *R = Role = Job function or title which defines an authority level*
- *P = Permissions = An approval of a mode of access to a resource*
- *SE = Session = A mapping involving S, R and/or P*
- *SA = Subject Assignment*
- *PA = Permission Assignment*

- *RH = Partially ordered role Hierarchy. RH can also be written as: ≥ (The notation: x ≥ y means that x inherits the permissions of y).*

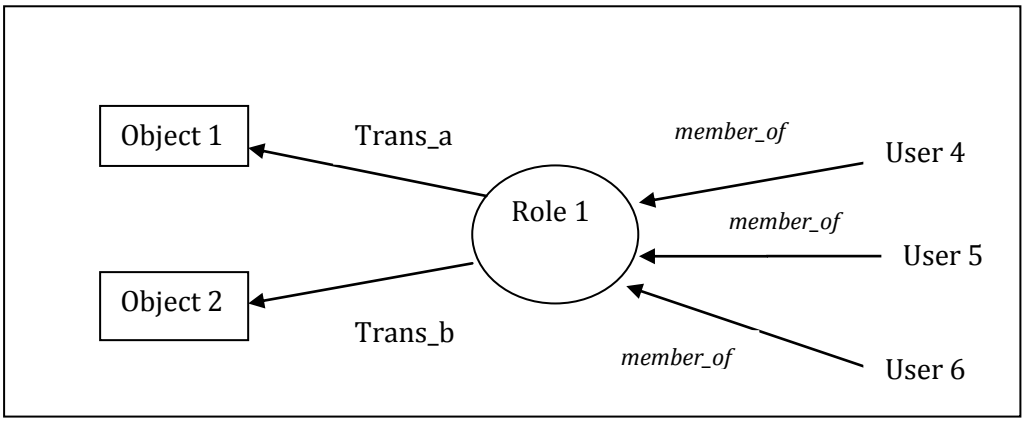


Figure 3.7 Roles and Relationships

(Source: culled from http://csrc.nist.gov/rbac/ferraiolo-kuhn-92)

This model is easy to implement, but does not cover all the aspects of securities in operating strategic business information systems. It mainly focuses on assigning access controls to different types of roles in an organisation, which makes it an incomplete model. However, as far as the access controls to different roles are concerned, this model is the best suitable approach, but which has to be combined with other models to use within the organisations.

An example of how this security model affects usability can be seen in an EPOS system such as TESCO checkout tills. At the point of checkout, the till attendant (depending on his/her role assignment, role authorisation and /or transaction authorisation), cannot perform certain tasks such as product cancellations and/or Cash refunds. This may require an override key with permission from a senior staff with a different role and authority. This process delays the performance of the task by the till attendant, which of course, causes more delays and customers queues build up pressure on the staff concern. The result is that, they may sometime cut security corners because of such security bottlenecks which reduce usability in certain areas of business.

3.7 Examples of Security vs. Usability failures:

Nguyen et al. (2003) stated that, security is protective measures taken to ensure safety of data and resources of users and owners of IS. The level of security and privacy to be provided

depends on the requirements. Clearly the users will want to ensure that the system will not violate the business or the user privacy. Security vulnerability occurs in application when the user authorisation level is poorly or not enforced; and the user is granted inappropriate access to data (Arreymbi, 2007). Most Computer vulnerabilities stems from internal abuse, active attacks from external sources and /or loss of information. An example of computer security failure (in an application) is when the application displays data from a particular account which belongs to a different client; especially when the client did not request the data, this indicates the application has a security failure. The failure in the systems security configuration and settings gave access to data which the user did not have permission to access. Therefore authentication by the system to verify user's identity within any information system is very important. In areas such as Banking and IT security sectors, system specialist have developed and applied the concept of 'Chinese wall' to enforce security uniformity. In this situation, the bank's front office staff are not given access to customers' data like bank account, where the integrity of such data can easily be compromised.

Security as we know it today is one of the most challenging experiences and consequences of a rapidly changing environment. This comes as a result of many factors: the rapid development of technologies; the increasing techno-savvy nature of users; and the very dynamic environmental and/or operational demand pressures.

Increasingly users are becoming very knowledgeable in the application, use and manipulation of technologies; a privilege that was hitherto enjoyed and left only for the few highly skilled technical specialists. For example the Web2.0 technologies can now be used and manipulated by almost everybody in society. However, this comes with its own security implications, for example, in Facebook (a social network system), the privacy/security settings have not been very user friendly. Most of the security features are not easily visible; buried in obscure menus which are difficult to find or setup, and therefore they do not truly give users the liberty to opt-in or opt-out. The users have little or no control of the information published or shared online, which leaves the users vulnerable to cyber threats such as malicious tagging of photographs without the user consent. But in some recent developments (BBC, July 2011), the owners of the Facebook system have improved on this aspect of usability and security; they have provided some added features which gives users the easy privacy setup facility and control of their information. In other situations, sometimes, when users try to access the site, the system allows the user to revisit a page that is frequently visited without asking for any

security password or any security protocols. This involuntary use of cookies within the system is to some extent good because it allows the site to be relatively usable, but also bad because it leaves the system unsecure and vulnerable to malicious attacks. Therefore, in trying to make the system very usable or user friendly, we are at the same time, leaving the exposed to security threats and vice versa.

Another example point where usability is hindered because of security in a Facebook system is encountered during sign-in access process. In order to gain access, users are often asked to enter ID and password; when this is done, you sometimes have situations where the system/website is not able to process the authentication effectively. And the system or webpage then freezes up and/or closes down, with an error message saying; 'browser (internet explorer) is closing down'. And as such, the whole system becomes more unusable and shuts down completely. However, this sometimes depends on the browser in use, and when this happens, it takes much longer time for the website/system to recover from the system error. This causes delay which can be very frustrating to users; a performance bottleneck and security challenge which many users face on social networks. Most often, when users want to use any networked/online email systems for example Microsoft Hotmail; users log-on (authentication) to the system, it sometimes takes longer for the User ID and password to be processed; and users have to sit and wait (reduced usability), just to get access, therefore cannot easily use the system. The process can be repeated many times just to work around the problem. It has been observed that security and usability are closely linked to performance bottleneck, and any ability to manipulate or balance them can be very tricky, a task that can leave the system dangerously vulnerable or unusable.

Looking very closely at some of the banking technologies widely used today, one can clearly see that the security and usability risk implications are very genuine. For example, the use of PIN to access bank accounts at ATMs. Most PINs are very easy to access and use, however, with the many accounts cards and services that are available from different providers, and which all require PINs or passwords; users tend to easily confuse or forget which PIN belongs to which account, card or service or provider. And when users try to access/use the card account, or service with three consecutive PIN/password failures, the system automatically blocks/refuse access until a new PIN/password is reset. This process is good for security but very daunting to the user because of the level of demands, anxiety and stress it

put on users during that period. Again, when users forget their pin and make request for a new pin number from the bank or provider, it takes a long time (3-5 working days) to get it back and during which, it amounts to a period of inactivity for the user as per that account. Also prominent in this issue is the fact that, when the bank manages to send a new pin number, it comes mostly on a security enhanced paper which can be difficult to access. And to access or view the new pin number from the paper, it requires users to carefully scratch through it with a coin. However, most often, even after careful scratching, users cannot clearly see the pin, and the paper and/or pin is destroyed out of scratching. Again, a new request for another pin is made but the same problem still occurs. Sometimes, to resolve the problem, users have to go into the bank branch to get the pin accessed. But even that, the bank staff most often, also struggles to read the pin digits. This security process has greatly reduced the usability of the entire system for the user. There are many ways to improve the system, to make the pin more readable/visible and user access friendlier and yet secure, as will be highlighted later in the discussion.

Another good example where security conflicts with usability is evident in most University or Organisations' security systems where smart ID cards have been introduced to give easy access to the facilities. However, most often, the cards either do not work or cannot be read properly when swiped or touched. When this happens, the system becomes unusable and the users (students/staff) have to sometimes swipe many times to get access, or make request to a security desk officer to give them access to the facilities using another means such as, automated button or keys. In this case, although the system was introduced to protect the facilities from unauthorised user access, and to keep the organisation safe, secure and yet provide quick/easy access to bona-fide users; the technology has sometimes made things more difficult and very daunting to users wanting access to these facilities. From these examples, it is obvious that balancing usability and security is a very difficult task and tricky act to achieve in designing and developing information systems, with no quick fixes.

3.8 Aspects of Human factors in Security of Systems

Any security mechanism is only effective when used properly and a systems' security effectiveness is only as good as the last user; depending on knowledge and ability. In fact, this shows that, a system, no matter how well designed and implemented, will have to rely on people. Information security involves both technical side and human side that must be well

managed. Gonzalez and Sawicka (2002), citing Reason's, J (1997) book on "Managing the Risks of Organisational Accidents" stated that, human factors play a crucial part (80-90%) in the majority of organisational accidents (security problems) such as the case in the recent Swiss Bank (UBS) London branch rogue trading crisis (BBC September 2011) . This leaves us in a complicated situation where nobody understands and/or knows why and how the accidents happen. It is a troubling feature of modern "security know-how" because we can implement appropriate technical solutions, but have continually failed to handle the human factors – "people security problems". Schneier, (2000; 2003) paints a vivid picture of the security situation by saying "...I tell prospective clients that the mathematics are impeccable, the computers are vincible, the networks are lousy, and the people are abysmal. I've learned a lot about the problems of securing computers and networks, but none that really helps solve the people problem." Gonzalez and Sawicka (2002) are of the opinion that "human performance must be seen as embedded in a work environment shaped in subtle ways by technology and human behaviour. Any improvements in security and safety require improved understanding of feedback" Therefore, a better understanding of the dynamics of the problem, that is, propagation of effects linked by causative mechanisms, is essential.

Many researchers (Whitten & Tygar, 1998, Sasse, 2010) have also highlighted the fact that human factors are perhaps the biggest and most common current barrier to effective computer security. In fact, most security mechanisms are perhaps too difficult and confusing for the average user to manage correctly. Therefore, developing security systems that are usable enough to be effective is a very big and difficult challenge; and user interface design strategies that are appropriate for other types of systems will not be sufficient to solve it. Whitten and Tygar, (1998) stated that, strong cryptography, provably correct protocols, and bug-free code will not provide security if the people who use the software forget to click on the encrypt button when they need privacy, give up on a communication protocol because they are too confused about which cryptographic keys they need to use, or accidentally configure their access control mechanisms to make their private data very readable by all and sundry, especially in a networked community. Majority of users are now connected to the internet and most tend to carryout financial transactions in this environment which pose a big challenge for security developers. Such has been the headache to many systems designers, who are now looking for batter ways to simply good systems configuration that is easy to manage by even novice users. According to Bishop, (1996), more than 90 per cent of all computer security failures are probably due to user configuration errors.

There are many ways to make system security more usable and effective; through training of users, automating mechanisms and /or improving the UI such that systems security can be made sufficiently clear and intuitive to be used and managed effectively by users. In their research Gonzalez and Sawicka (2002) looked at behavioural patterns and the perception of risks by users, in order to ascertain where the problem lies. Others (Whitten & Tygar, 1998; Hinson, 2003) have used different processes and models to try to find solutions and believe that, active IT risk analysis and risk awareness and effective security policies and controls may be the ultimate way to resolve the mounting problems of information security now faced.

3.9 Summary

This chapter has considered the key components of usability and security in relation to the problem of design and development of IS. It looked at some of the challenges involved, and evaluated some of the models used for designing information systems for security and usability. Some aspects of system failures and human involvement were also investigated.

For the successful implementation and use of business information systems in solving security and usability problems, there are not many available models that fits the bill. Some existing models concentrate mainly on one particular area of the business information system. Also, there are some guidelines given by standardized organizations such as ISO, COBIT, IT Governance, etc., but they mostly tend to cover technical aspects which are very important. However, when we analyse most business information systems, it is not just technical factors that needs to be taken into account, but also social factors which surround the system.

These are the key characteristics that summarises and form the basis for consideration of a novel approach to IS design and development in the next chapter.

Chapter 4 A perspective on Software Development Life Cycle (SDLC) and Usability Life Cycle Models

4.0 Introduction

This section provides a review of Software development Life cycles to highlight the areas of similarities and/or differences in the various methodologies. It will discuss the suitability of the models in the development of user interface (UI) and looking at the merits and demerits with respect to usability of the information systems. The discussion will attempt to find the best systems models for the User interface design process by using the User centred design (UCD) approach. As can be seen, every software project depends on two things: User interface design and functional parts. Although both of the parts belong to one project, they differ in processes and have equal importance in the successful completion of the product. It is evident that both the SDLC and the Usability Life Cycle seem to have common components, but they differ in many ways, by the internal processes of development in each of the phases.

4.1 Software Development life Cycle (SDLC)

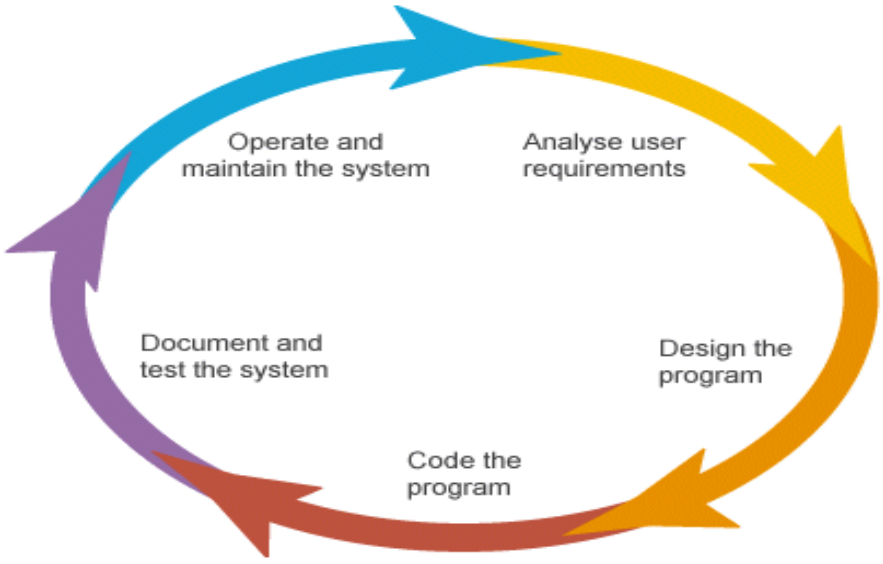


Figure 4.1 Traditional SDLC (Source: www.samsvb.co.uk/.../development_life_cycle.gif)

The traditional software development life cycle components as observed in figure 4.1 are:

- Requirements and Analysis
- Design
- Development/Coding

- Test and Documentation
- Implement and maintain.

Pressman (2001, & 2010) suggested the following stages in system development lifecycle:

4.1.1 Requirements and Analysis

The requirements and analysis process can be carried out, not only to collect the information from the client but to also understand what the client wants exactly. This phase has to be performed to have the knowledge of understanding the domain of the software system built and know the required function, behaviour, performance and interface.

4.1.2. Design

Design is to translate the gathered requirements into form of software which can be further evaluated for quality to begin the coding. Design is a multi-stage process, which concentrates on the interface requirements, Methods of development, procedure of development and architecture.

4.1.3. Development/Coding

This is phase of converting the requirements into usable form with the machine understandable language through the designed procedures and processes. All the designed stage, anything done on paper will be kept and translated into the machine readable at this stage.

4.1.4. Test and Documentation

Testing will be conducted to check the developed code to yield the accurate output required to the customer. This is done through the requirements document buy the testing team in a software firm. This phase begins when the development of the software ends or a portion of software is tested based on the model of development.

4.1.5. Implement and maintain

This is the last phase in the software development life cycle where the final product is delivered to the client and maintained for some time at the client's environment for real life practical use. Software design methodology (SDM) provides a methodical approach to

software design that uses notation and guidelines for software design, Sommerville (2011). Examples of structured design methods are Structured Systems Analysis, Object-Oriented Design, and Jackson Systems Development. The Structured design methods uses notation to describe design, design guidelines and design report format. It supports models of a system using some or all of the following approaches:

- A Data Model where data transformation is used in modelling the system
- An entity-relation model describes entities in the design and relations between them
- In structural model interactions between systems components are documented.
- Object–Oriented methods models static and dynamic relations between objects, which includes inheritance models of the system.

Software design according to Pressman (2010) refers to set of principles, concepts and practices which lead to developing high quality application which may be a product or system. Design principles establish a framework that governs the design of the work. There is creativity in design and contribution from many specialists, for example stakeholder requirements, business needs and technical considerations which leads to the development of a system or product. IS specialist emphasise the importance of design (Hughes et al, 2004) if it is decided to build a new system, rather than the purchase of off the shelf application. A design phase will be important in the Software development life cycle (SDLC). The design phase translates business specification for aspects of the automated system into a design specification of the computer processes and data stores.

Aspects of the system to be designed in a new application are the following:

- Inputs
- Outputs
- Processing
- Information and data structures.

The design phase in the SDLC provides a foundation that leads to successful implementation of the system (Sommerville, 2001). The design process involves adding details as the design evolves, with modifications of earlier designs. Decomposition of design allows omissions and errors in earlier phases to be modified.

The following Agile development illustrates phases in the SDLC including design as suggested by (OTS Solutions, 2008-2010).

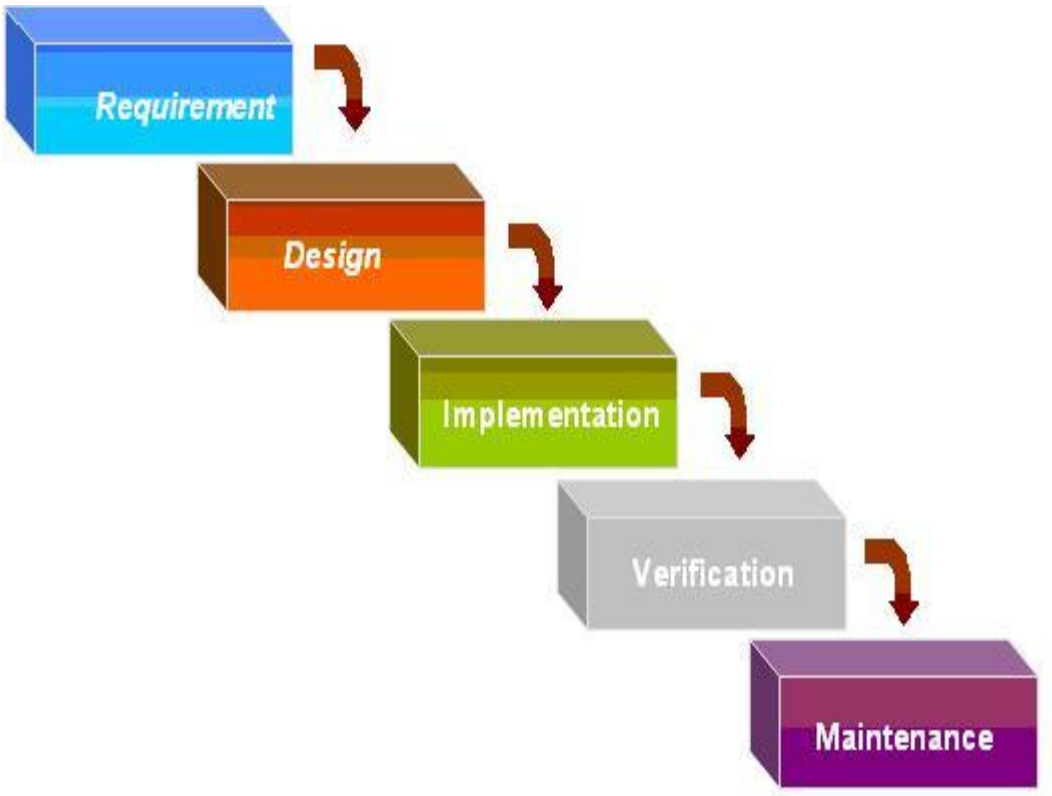


Figure 4.2 SDLC stages.

According to Kaner et al, (1999) software and application designers split the design phase into stages. The stages are:

- external design and,
- internal design (which imposes constraints and requirements on the other).

The design phase is very important in usability, particularly the external design, this is because the external design includes description of the user interface, and it describes screen and outputs including commands to be used and syntax. The user manual and external specification are documents produced during the external design. The Internal design composes of structural design that describes how tasks are to be subdivided among different pieces of code, and data design that describe details of data the code will work with, and the

working of the code is the logic design. Pressman (2010) point out that, design is the place in software engineering where quality is promoted. Design provides software representations that can be assessed for quality. Exclusion of the design phase in the SDLC results in risks of building a system that will fail when small changes are made, and may be challenging to test, and the quality cannot be evaluated until late in the SDLC.

The figure 4.3 shows software construction and role of detailed design. Trung (2007) explains that, aspects of detailed design may be performed before construction. Many aspects of design are performed within the software construction activity. Software construction is linked with software design.

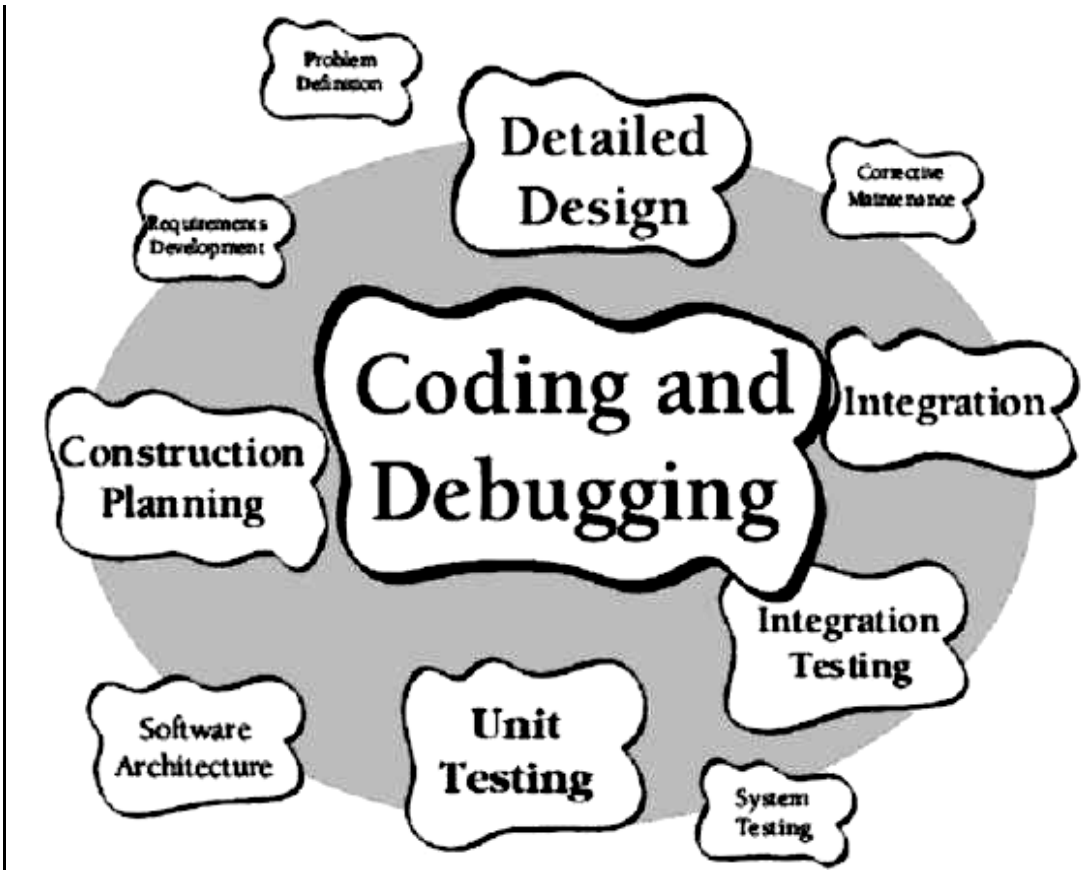


Figure 4.3 Software Construction and role of detailed design (Source: Trung, 2007)

4.2 Usability Engineering Life Cycle

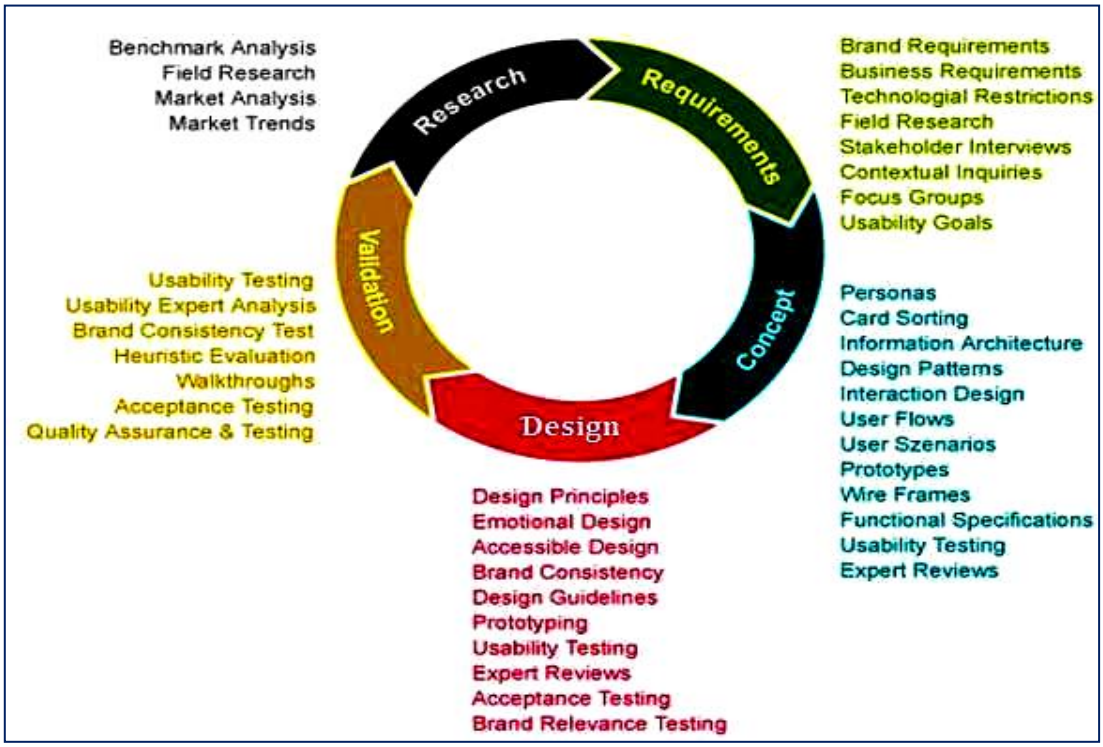


Figure 4.4 Usability Engineering lifecycle (Source: http://usablebrands.de/usability.html)

As with the software development life cycle (SDLC), the Usability engineering life cycle goes in the same way, with similar processes and components.

Looking at the above figure 18, it is clear that the components of the Usability Life cycle are:

- Requirements
- Concept
- Design
- Validation and
- Research

4. 2.1. Requirements

The requirements are gathered in the user prospective that who will use the product? And who is the ultimate /end user. What is the technology use on the device and how will be the flow of the navigation and how should be the performance of the device?

4.2.2 Concept

This phase mainly concentrates on the work flow of the requirements gathered in the first phase and it also emphasises on the design patterns to be used.

4.2.3 Design

It is the most important phase in the user interface because this phase will have all the contributions of the above two phases: It contains the concept that what are the interface components to be used and where they are to be used.

4.2.4 Validation/testing

In User interface design testing is to as the user or the subject that the required outcome has been accomplished or not. If the required scenario does not meet the condition it is to be altered to meet the condition until desired outcome is met.

4.2.5 Research

Conducting research on developed product to refine and update for future products.

4.3 Similarities between Usability Engineering and Software Engineering

1. Both of the life cycles have requirements keeping the user as a main subject to understand and know the user's or subjects interest to satisfy his requirements.
2. Convert these requirements by following the models and processes to design.
3. Either of the life cycles tries to convert the designed models into practical use by adapting their individual methods of testing to rectify the errors and deliver the product.

4.4 Differences between Usability Engineering and Software Engineering

1. Both have the different levels of iterations and evaluation :
   Usability engineers will iterate early and frequently with paper prototypes, screen sketches design scenarios etc. But software engineers will test the process and requirements to eliminate the errors.
2. Both have the different terminology :

This is the biggest problem where at one stage the software engineer should have to implement the usability into development and they need to understand the terminology of usability for better use.

3. The way of representing the requirements are different in both of the life cycles and so there is no possibility of incorporation of processes rather than re-defining the process model.

4.5 User Centred Design (UCD)

Often developers think to develop the software with good business goals and heavy graphics and high level components and forget how the user can handle and use the developed product and adjust to it. User centred design allows the developers or designers to remember that the system is to develop for the end user by understanding their attitudes and behaviour but not asking the user to adjust with the developed system. Adapting UCD will boost the sales of the product by having customer satisfaction, Efficiency and user friendliness. Increasing the end user satisfaction will be attained by using all the good principles and guidelines of user interface design as demonstrated (http://www.usabilityfirst.com/about-usability/introduction-to-user-centered-design, Accessed on 04/05/10). User centred design is a process which provide its support for the whole development process for creating easily usable application by keeping the users as a key element through its activities (see www.UsabilityNet.org)

There are four activities of user centred design:

- Understand and specify the context of use
- Specify the user and organisational requirements
- Produce design solutions and
- Evaluate designs against solutions

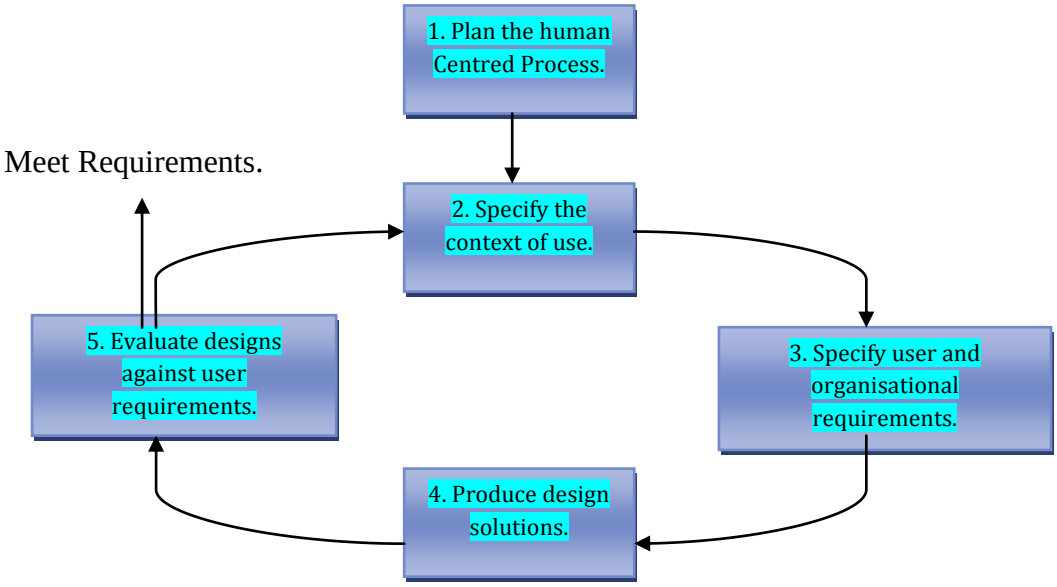


Figure 4.5 Flow of the User centred design activities

(*Nigel Bevan:* Usability.Net Methods for User Centred Design)

Therefore, developing an application for a mobile device using the concepts of usability engineering and user centred design needs a software development model, which can help the developer to develop an easy to use and pleasing, effective and usable product.

The most common software models available are:

- Waterfall model
- Prototype Model
- Spiral Model and
- "V" Model
- Iterative Models.

These models use the basic components of the software life cycle and as we have already discussed them already we will focus on the flow or the process of each model with respect to the user centred design.

4.5.1 Waterfall model

This model is again being highlighted here for the purpose of clarity of discussion

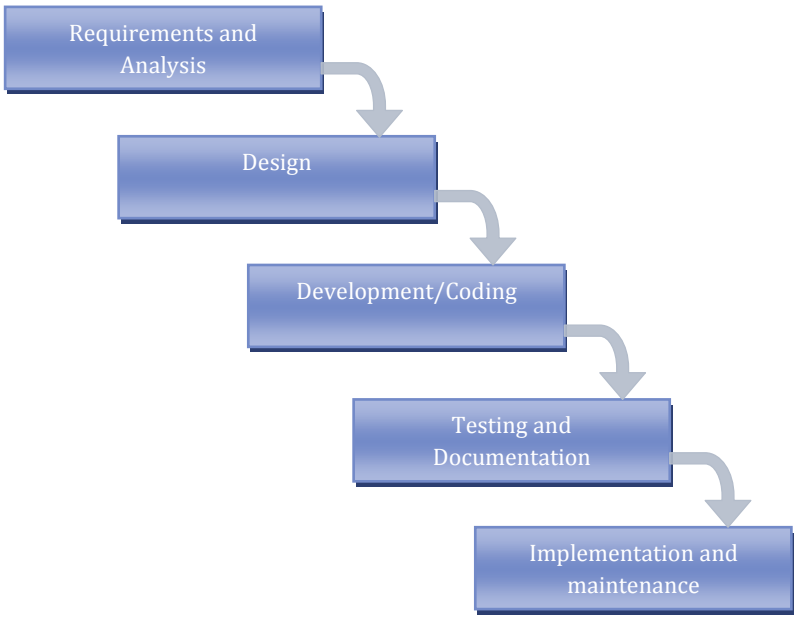


Figure 4.6 Waterfall model

This model is considered as the traditional model of software development life cycle and as it appears, it is a sequential model. The requirements are gathered, analysed and this process is done until the final stage of requirements is met; then proceed to the next step. The design begins after requirement analysis is done; coding commences after completion of design. After the programming has been completed, the code is integrated and testing is done. After enough testing is done, the system is integrated and is followed with regular operation and maintenance of the system. Sommerville (2011) highlight the fact that, the result of each phase is one or more documents that are signed off. Looking at the suitability of the model, Pressman (2010) believes the waterfall model can easily be adapted in IS projects when the requirements are well understood; when activities are sequential from communication through deployment; for example, when there is well-defined adaptations or enhancements to an existing system. It may also be used in new development applications or systems but only when requirements are defined well and stable. The Waterfall model is not very good on usability due to its inflexibility; it may well provide for good security but not necessarily follows a secure design, which utilises a bottom-up approach to IS security system design.

According to Pressman (2010) some of the challenges faced by this model stems from the fact that:

- Information Systems projects in the industry rarely follow the sequential phases of the project, though the sequential model can foster limited iteration, this will cause so much confusion between different phases.
- It is challenging for customer to state all requirements in the initial requirement analysis phase. Uncertainty of the unknown takes place at the beginning of the project, and as work progresses uncertainty decreases, this is normally referred to as cone of uncertainty in Project Management.

4.5.2. Prototype Model

Somerville (2000, 2011) highlighted the fact that the Prototype model is an advancement of the waterfall model, covering the drawbacks and change in the development process. This model is considered to be the reducer of the risk at the requirements stage. The process of the prototyping is shown in the figure below.

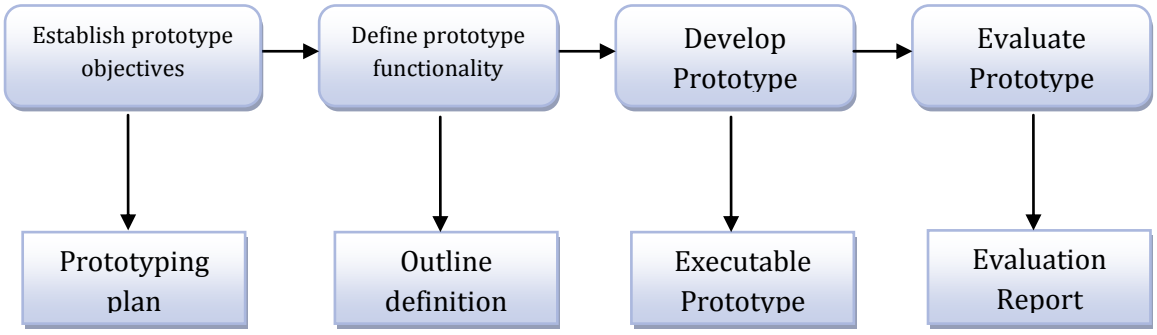


Figure 4.7 Prototyping model (Source: Ian Somerville, 2001)

Prototyping has many versions of the model and each has its pros and cons. Here are few examples:

1. Throw away

2. Rapid prototyping

3. Incremental prototyping

4.5.2.1 Throw-away Prototyping

This type of prototyping is done to put the system in a process at the initial stages of the project to evaluate the working of the system. If the outcome suits to the user requirements then the prototype is thrown away and another method is used to complete the system. Because this type is thrown away this is not considered as a final system.

4.5.2.2 Rapid prototyping

Rapid prototyping is an iterative process which involves the users throughout its development process where the traditional or normal prototyping will follow the old waterfall model in its own way. In rapid prototyping the initial system is developed using a throw away system to evaluate the requirements and flow of system capabilities. A basic iterative prototyping model can be seen in the figure below.

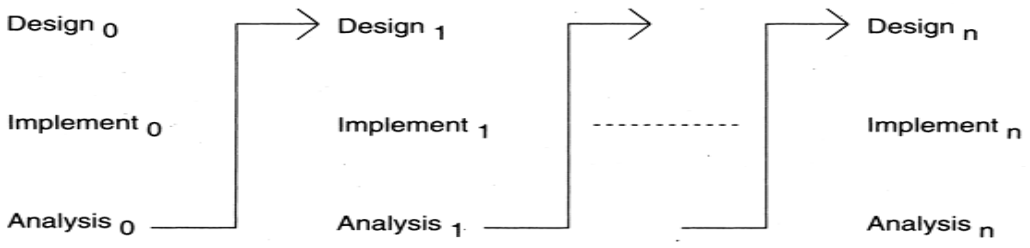


Figure 4.8 Iterative prototyping

It can be seen here that as first phase of design is done, the design is implemented and analysed and afterwards, the analysis report is used to refine the pitfalls in the design and this is done up to "n" stages, until the desired output has been achieved.

4.5.2.3 Incremental prototyping

This type of incremental approach can be compared to 'building blocks'; which is incrementing each time a new component is added or integrated, and based on an overall design solution. When all of the components are in place, the solution is complete. An advantage of this method is that, it provides an opportunity for t the client and/or end-users to test the developed components and their functionality; and also to provide feedback while

other components are still in development, therefore, can influence the outcome of further development.

4.5.3 The V-model

The V-model emphasises on Verification and Validation as illustrated in the diagram. Verification is the process of demonstrating that a program meets its specification, validation demonstrates that a program, application or system meets the needs of its stakeholders and users (Somerville, 2011). It can be argued that usability is emphasised in Validation. For example, are we building or developing the right system as far as users are concerned? The V-model shows that Verification and Validation complement each other. The verification task is completed against the business/user requirements. There is a quality check to establish whether the right requirements specified for that phase has been captured. This is also refined in subsequent phases to verify that the user requirements are enhanced to specify the detailed non-functional and functional requirements.

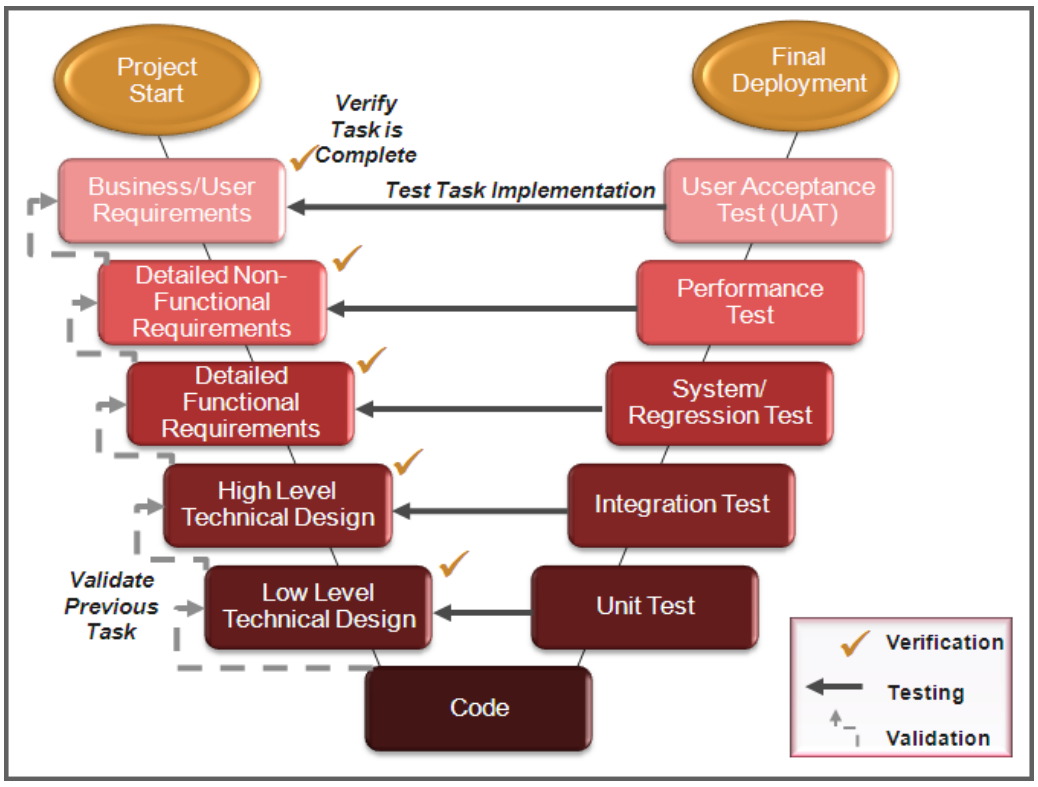


Figure 4.9 Illustration of the phases of the V-Model

(Source: Infinite Computing Systems, 2011)

The V-Model is usually regarded as extension of the waterfall model, Hughes et al (2004). It is a useful model with two quality control processes - one between stages and another across the V. In the model the requirement specification is a major document and should contain quality attributes and function which the customer requires in the system. And at each phase in the development, there is a test activity, for example using the acceptance test plan. User acceptance testing is conducted to ensure that the system meets the requirements identified. Testing is carried out throughout the lifecycle, known as the v-model testing (Veenendaal, 2002). Early testing has the following advantages:

- Early work products such as requirement definitions and specifications are used to build subsequent products, as errors are identified in the early deliverables; early test activities improve the development activity.
- Early testing reduces cost elements; as previously explained, the earlier errors are fixed in the life cycle the cheaper it is.
- The model saves time; the V- model allows for the deliverable right before moving to the next stage. For example, the requirement has to be correct before we build the system, so less rework is needed.
- In contrast, the waterfall model activities are done sequentially and testing is done at the end; it is challenging at this stage to work up the phases and modify a user requirement that was not adequately captured as the model is not designed to be iterative and flexible. The waterfall model will fit on a small application where requirements are relatively stable, in such instance Usability and Security will be relatively balanced, but not suited for large complex system. Therefore, if the model is used in a large application, the balance or trade-off between Usability and security will be a complex challenge.

4.5.3.1 Disadvantages of the V-Model

Despite the advantage of test early in the life cycle, the model has its shortcomings. The SoftDevteam, (2010) suggests that the model is only applicable in projects where software requirements are clearly defined; this is also applicable to waterfall model. The model can only be applicable in situations where the tool and development technologies are well known.

Another drawback is that V-model like waterfall model is not flexible; once requirements are defined, it is difficult to change. It is rigid and follows a strict sequence just like the waterfall model; therefore in the real world when developing software systems and applications, adjusting project scope is difficult and expensive. In the V-model, the software is developed during implementation and, prototypes to software are not produced.

The lack of prototype in the V-Model impacts on usability. Development model that builds on prototype promote usability and if the system is usable IS specialist can provide a work around on security challenges. Also, by increasing user familiarity with the application, they are mentally ready to accept and use the application. Iterative approaches uses prototype before the system is implemented.

4.5.4 Iterative Models

Although this research is to carry out comparison of different software models for a usability and security design of IS, The iterative Spiral and "V" models are not suitable for developing small scale applications. The models are basically designed for very large projects with many development and other resources. As has been established some of the models will not be dealt with in much detail. Table 5.5 provides a summary of the models, their benefits and disadvantages in relation to systems design and development.

4.6 Summary of advantages and disadvantages of system development processes

Model	Advantages	Disadvantages	Applicability
Waterfall	Simple Easy to execute Intuitive and logical Easy contractually	Too risky Requirements are frozen in early stages. May chose outdated hardware/technology Disallows changes when passed the prior stages. No feedback from users	Well understood problems short duration projects automation of existing manual systems

		Encourages requirements bloating.	
Prototyping	Helps requirements elicitation, Reduction of risk Final system are better and more stable. Strong Dialogue between users and developers Encourages innovation and flexible designs Missing functionality can easily be identified	Possibly higher cost and schedule. Very good at requirements stage. Do not accepts later change	Systems with novice users; or areas with requirements. Heavy reporting based systems can benefit from UI prototypes
Iterative	Regular deliveries, leading to business benefit, Can accommodate changes naturally Allows user feedback Avoids requirements. Naturally prioritizes requirements.	Spiral : Not fully compatible with Re-Usability. Totally different behaviour to different Projects.	Risk of long projects cannot be taken requirements not known and evolve with time For businesses where time is important.

Table 4.1 Summary of systems development processes.

4.7 Designing a Secure and Usable information system: An analysis

Fidas et al (2010) highlight the fact that system designers involved in designing usable security are at a crossroads. They try to achieve a compromise between a highly usable system for the users, as well as protecting the assets of the users; but sometimes exposing the users to security threats. Many analysts are suggesting that, users of the system perceive

security as someone else's problem, and therefore make poor decisions to related security issues, and sometimes considering them unrelated to the tasks at hand. And as a consequence of this complexity, designers have been developing systems with two separate objectives: security for the provider and usability for the user. Mead et al., (2008), have stated that most often, security requirements have been identified during the system life cycle. But generally, the requirements tend to be mechanisms such as password protection, virus detection tools and firewalls. And in most instances the security requirements are not integrated, are developed independently of the rest of the requirements engineering activities. As such, security requirements that are specific to the system and that provide for protection of essential services and assets are often highly compromised or neglected. The requirements elicitation and analysis that is needed to get a better set of security and/or usability requirements seldom takes place. Interestingly, Fidas et al (2010) believe that, a possible method for designing usable system that is also secure, is to adopt a user centric approach (in the design of information systems with usable security). User centric approaches are widely used in situations where user requirements are difficult to gather and understand. The approach puts the end users to be at the centre of the software development cycle, and enables the designers and the users to have a common mental model of the system. A mental Model is an internal model representation of an external reality, for example, information system and its functionalities. The early days of IS development was considered a privilege that must be controlled. However, over the years IS devices and networks have evolved into everyday tools for ordinary users to perform their tasks with little or no training. Therefore in trying to meet these challenges designers must look for ways to balance security and usability.

There are areas of overlap between aspects of HCI and security. The areas are: user authentication, secure interface design and usability of security products (Op cit).

User Authentication: In most information systems design and development, passwords, passphrases and personal identification (PIN) are widely used for user authentication and access control. Users prefer easy to remember passwords that they can reuse in other application, and are upset when system policies do not accept their password or require them to change it often. Users often require assistance to recall or reset their password. Password problem is used as an example of conflict between security and usability. It is important to note that, security dictates hard to decipher/guess passwords and usability dictates easy to remember passwords.

4.7.1 Interface Design and security indicators

The way internet and electronic mail (email) protocols were designed originally; they were not designed with security as a goal. The initial goal was to promote fast communication of messages between networks (Fidas et al., 2010). However, the increase availability, high usage and the commercialisation of the internet resulted in the need for encrypting or signing electronic messages or both. A common metaphor from the physical word in email is the notion of letters, documents and books. In the digital world these objects are active, executable code integrated with rich user interfaces and advanced software capabilities. The way the technology is presented may be misconstrued and result to privacy invasion.

Schulz et al (2001) have suggested that human factors and usability have in the past not played a prominent role in IS security issues. Human factors and usability issues have traditionally had only limited effect in security research and secure systems development. It can be argued that most security designers ignored usability factors because they lacked the expertise and/or awareness of the importance of human factors. User perception of security and how they respond to security protocols is an area for further research. However, there is growing recognition within IS development industry, that security problems can be largely resolved by addressing issues of human factors, trust and usability (Arreymbi, 2007; 2011). There are numerous well documented evidence and publicised security breaches which are linked to human errors, that might have been prevented through design and development of more usable systems. There is an inherent trade-off between computer security and usability. On the one hand, it is true to say that a computer system without password is very usable but not secure; but on the other hand, a computer that makes you authenticate every five minutes with a password or access code might be secure but not very usable. There are certain applications that require maximum security protocols, for example Back-end banking Account management systems; where high security is needed to keep personal account information very safe and therefore such systems will have very low usability. However, with the front-end user access system, usability should be high with some adequate but comparatively low security features. Users need computers, and if they can't use one that is secure they will use one that's not secure. However unsecured systems are not usable as they get clogged up with viruses, spam or hacked worms and soon become useless. Therefore, the question is; at what level does the user want to take or leave risk selectively? This will depend on the environment /situation the user is faced with or operating in; that is, certain situations requires specific actions such as high security and others high usability.

Most often nowadays, there is increasing demands within the industry for system designers and developers to come up with good design and development of secure systems that are usable according to user needs. These demands have also been supported by testing and quality experts, noting that, information systems designed and developed should be fit for purpose for the user, and should satisfy user's needs; both in terms of security and usability. There are IS methodologies for example V-model, which emphasises the importance of designing and developing usable needs systems. The v-models stresses that the requirements should be adequately captured and at each life stage, carryout an evaluation of work product in validation and verification. The Agile and prototyping development methodologies also do emphasise user involvement. The Agile methodology incorporates or makes use of user stories to understand user requirements. JAD method emphasises user involvement in implementing a usable system through user workshops.

4.7.2 The dilemma

The growth of computer network connectivity through the internet has given users more flexibility and the ability to access information anywhere anytime and anyhow. However, lacks in system security can have serious consequences such as unauthorised modification of systems and data, denial of service attacks and viral spreading of malware, Trojans etc. and data corruption. Numerous security methods have been developed which rely on implementation by individuals users, the method may not accomplish the intended objectives if not used properly. The role of human factors in information security is considered very important and cannot be over emphasised. According to Schulz et al (2001), the lack of consideration of human factors issues in information security is not because of the lack or scarcity of security threats to computers and networks, but rather as a result of misplaced priorities. There are many internet/network security threats for example, eavesdropping on user's sessions. Information controls are techniques and procedures used to reduce the likelihood of security related threats that may result in unauthorised access, disclosure of information, theft, loss and compromised integrity of systems or data. Some examples of security control mechanisms in use today are; passwords used to log on to systems, file permission, and cryptography/encryption. Most default permissions on newly installed systems are not security effective, for example, the NT operating system, critical system 32 directory within the C-partition allows by default, full control to all users. Therefore, such

default permissions mechanisms must be modified if resources are to be safely secured. Users are always seen to be the weakest link in information security system, and administrators are encouraged to inspect the system logs to determine whether or not unauthorised activity has occurred. User resistance to information security measures has an effect on trade-off between usability and security. A system designed with poor usability provokes a high degree of user resistance and/or circumvention. Examples of such resistance include reluctance to perform tasks, failure to pay sustained attention to tasks, Schulz et al (2001) and/or security protocols.

4.7.3 Evaluating the integration of security and usability in the requirements and design process

Flechais et al (2007) suggests that many IS systems fail because the designers protect the right things in the wrong way or protect the wrong things. Many security systems approach have targeted security from technology and user interface viewpoint. Security should take prominence and placed ahead in the life cycle, precisely at the requirement and design phase. For example the V–model and incremental and iterative approaches involve users at the early stages, so that the requirements are captured easily. By involving the stakeholders, their needs are taken into consideration. Stakeholder involvement, analysis and communication are some of the strongest emphasis points in Project Management such as Prince Methodology. Stakeholders are those who will be affected by the project and their involvement is crucial to the success or failure of the project.

Other approaches have been suggested as appropriate in filling the gap that exists in designing secure and usable systems. In this section, we will highlight the existence of a few and analyse them, for example AEGIS, and SQUARE methodologies.

4.7.3.1 The AEGIS methodology for usable security

An in-depth analysis of the Appropriate and Effective Guidance for Information Security (AEGIS) methodology (Flechais et al., 2007), has been suggested as an appropriate methodology for use in design of secure and usable systems. This implies that, the methodology has been evaluated and found to cover both aspects of usability and security adequately. The methodology uses a UML Meta model definition and reasoning of the system assets. The semantics of the model and process allows developers and users to

propose constraints and needs of the security and usability aspects of the system in a simple way that can easily be captured and implemented. And according to Flechais et al., (2007), the method has been found to provide important tools for design and development of secure and usable systems. However, the process assumes expert knowledge of users and their capability of modelling the system assets; some of whom can be novice users and/or ignorant in the modelling process. Also, the users are supposed to design counter measures on an assumption that they have the most domain knowledge. In fact, the method is not very clear on the combination ratio and/or proportionality of the security and usability features; and the paradigm it is based on.

4.7.3.1.1 The AEGIS steps

The steps consist of the following:

- identifying and securing the correct participants and specifying their needs of usability,
- getting the user to model the system assets in context,
- assign a value on the assets,
- conduct a risk analysis and design countermeasures which addresses risk in a cost effective way.

In this approach, the usability needs are addressed in relation to involvement of users in the security design with emphasis on the user context during security requirements, modelling and countermeasure design. But it does not say how the usability needs are defined and captured.

AEGIS is good as a participative design methodology, where different stakeholders in the system are actively involved in the process of eliciting security requirements and deciding on security measures. However, the assumption that, the system stakeholders have the most domain knowledge is questionable from a design and development perspective. Also the methodology emphasises on integrating security in the requirement process, by identifying and relying on a single individual who acts as security lead in the project. And, one of the responsibilities of the security leader is to document decision making (OP cit).The process again depends on expert knowledge of the individual whose security objectives may be different from the rest of the team and not aligned to corporate objectives

4.7.3.2 The SQUARE approach

The Security Quality Requirements Engineering (SQUARE) is a method developed by the Software Engineering Institute at Carnegie Mellon University (Mead et al., 2008), for eliciting and prioritizing security requirements in software development projects. The SQUARE approach has been designed for use with IT systems, to help businesses build security into the early stages of the production life cycle. The SQUARE process involves the interaction of a team of requirements engineers and the stakeholders of an IT project. The requirements engineering team can be thought of as external consultants, though often the team is composed of one or more internal developers of the project. When SQUARE is applied, the stakeholders can expect it to result in the identification, documentation, and inspection of relevant security requirements for the system or software that is being developed. However, the SQUARE approach looks more suited to a system under development or one undergoing major modification than one that has already been fielded, although it has been used both ways. The Software life-cycle models describe phases of the software cycle and the order of execution of those phases.

The majority of the models being adopted by software companies tend to have similar patterns. Typically each phase produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced during the implementation phase and is driven by the design. Code is finally tested against requirements to ensure quality. Here we focus on incorporating SQUARE with standard life-cycle models, as SQUARE can be more effective when it fits into an organisation's existing development process. Some of the most commonly adopted life-cycle models and process / methodologies have been considered, namely, the waterfall model, Rational Unified Process (iterative and incremental model), spiral model, and Dynamic Systems Development Method (agile methodology - iterative and incremental model) and explained in detail.

4.7.3.2.1 SQUARE steps

It consists of nine steps that generate a final deliverable of categorized and prioritized security requirements:

The SQUARE methodology begins with the requirements engineering team and project stakeholders agreeing on technical definitions that serve as a baseline for all future communication. Next, business and security goals are outlined. Third, artefacts and

documentation are created, which are necessary for a full understanding of the relevant system. A structured risk assessment determines the likelihood and impact of possible threats to the system. Following this work, the requirements engineering team determines the best method for eliciting initial security requirements from stakeholders, which is dependent on several factors, including the stakeholders involved, the expertise of the requirements engineering team, and the size and complexity of the project. Once a method has been established, the participants rely on artefacts and risk assessment results to elicit an initial set of security requirements. Two subsequent stages are spent categorizing and prioritizing these requirements for management's use in making trade-off decisions. Finally, an inspection stage is included to ensure the consistency and accuracy of the security requirements that have been generated. However, analysis on the SQUARE approach has shown that the whole process focuses mainly on capturing the security requirements, but with little or no integration of usability features in the design /development of system. The approach tends to have many lengthy steps that require a lot of user or stakeholder commitment and takes a lot of time and can be very costly. The Table 4.2 below summarizes the steps in the SQUARE process.

Table 4.2: The SQUARE Steps (source: SEI, Carnegie Mellon University, Mead et al., 2008)

Step 1: Agree on definitions
Input: Candidate definitions from IEEE and other standards
Technique: Structured interviews, focus group
Participant: Stakeholders, requirements team
Output: Agreed-to definitions

Step 2: Identify security goals
Input: Definitions, candidate goals, business drivers, policies and procedures, examples
Technique: Facilitated work session, surveys, interviews
Participant: Stakeholders, requirements engineer
Output: Goals

Step 3: Develop artefacts to support security requirements definition
Input: Potential artefacts (e.g., scenarios, misuse cases, templates, forms)
Technique: Work session
Participant: Requirements engineer
Output: Needed artefacts: scenarios, misuse cases, models, templates, forms

Step 4: Perform risk assessment
Input: Misuse cases, scenarios, security
Technique: Risk assessment method, analysis of anticipated risk against organisational risk tolerance, including threat analysis
Participant: Requirements engineer, risk expert, stakeholders
Output: Risk assessment results

Step 5: Select elicitation techniques
Input: Goals, definitions, candidate techniques, expertise of stakeholders, organisational style, culture, level of security needed, cost/benefit analysis, etc.
Technique: Work session
Participant: Requirements engineer
Output: Selected elicitation techniques

Step 6: Elicit security requirements
Input: Artefacts, risk assessment results, selected techniques
Technique: Joint Application Development (JAD), interviews, surveys, model-based analysis, checklists, lists of reusable requirements types, document reviews
Participant: Stakeholders facilitated by requirements engineer
Output: Initial cut at security requirements

Step 7: Categorize requirements as to level (system, software, etc.) and whether they are requirements or other kinds of constraints
Input: Initial requirements, architecture
Technique: Work session using a standard set of categories
Participant: Requirements engineer, other specialists as needed
Output: Categorized requirements

Step 8: Prioritize requirements
Input: Categorized requirements and risk assessment results
Technique: Prioritization methods such as Triage, Win-Win
Participant: Stakeholders facilitated by requirements engineer
Output: Prioritized requirements

Step 9: Requirements inspection
Input: Prioritized requirements, candidate formal inspection technique
Technique: Inspection method such as Fagan, peer reviews
Participant: Inspection team
Output: Initial selected requirements, documentation of decision making process and rationale

4.7.3.3 The Object Oriented Approach

Object- Oriented approach (OO) according to Office of the Government CIO (2008); is a relative new system development approach that encourages and facilitates re-use of software components. With this approach, a computer system can be developed on a component basis that enables the effective re-use of existing components by other application (Bruegge and Dutoit, 2010). In such situations, higher productivity, lower maintenance, lower cost and better quality can be achieved. The objective of OO is an application assembly, which is the construction of new business solutions from components which are in existence. OO applies a

single object model that starts from the Analysis and Design stage and continues to the programming level. An object contains both the data and functions that operate upon the data. An object can only be accessed through the function it publicly makes; details of its implementation are hidden from all other objects. Encapsulation in OO provides improvements in traceability, quality, maintainability and extensibility which are principle features of well-designed Object Oriented Systems. According to Bruegge and Dutoit, (2010), objects exist in many aspects of our lives, in nature and in man-made entities, business and products we use daily. Object-Oriented development suggests that, Object-Oriented (OO) techniques are applied during the analysis and implementation of the system. The approach employs the analyst to look at all the objects in a system, their similarities and differences, and how the system can control the objects. Sommerville, 2001; Bruegge and Dutoit, 2010) have stated that OO systems are easier to manipulate than systems developed using functional approaches. Objects include data and operations used to alter the data. Modifying the implementation of an object or adding components or services to the object does not affect other objects in the system. Within the OO approach, there is a distinct mapping between real-world entities e.g. components of hardware in the system. This improves understandability and maintainability of the design. The diagram shows that with the sans GUI object, system developers use abstraction of real world systems put into entities and their relations.

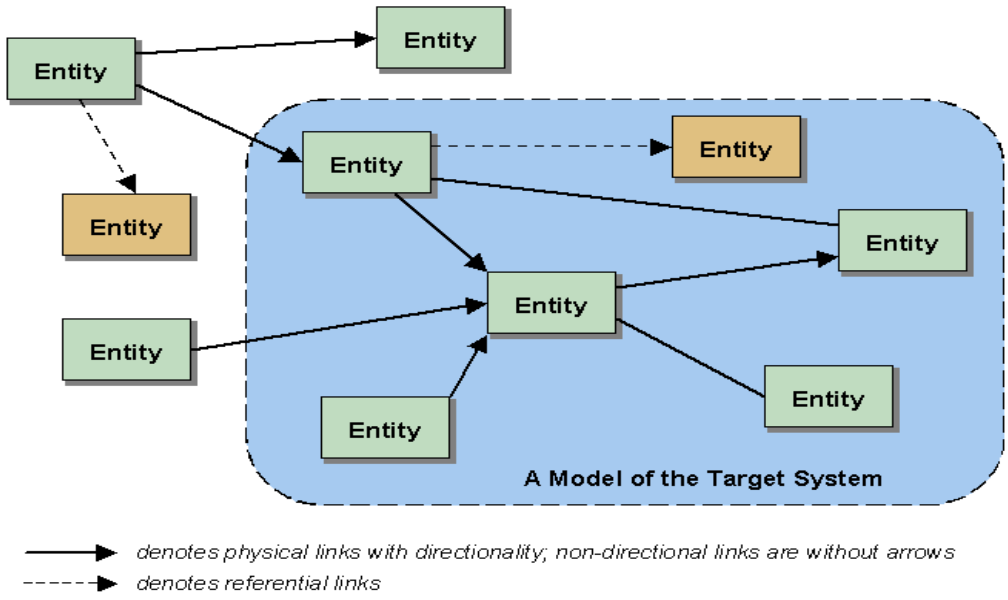


Figure 4.10 A target system model (Source: SansGUI, 2003).

The diagram shows a typical model with component, entities, reference entities, and physical links which users can build, edit and run in the GUI environment. The user's models are configured in the Sans GUI Run-time environment with the model building blocks provided by the developer.

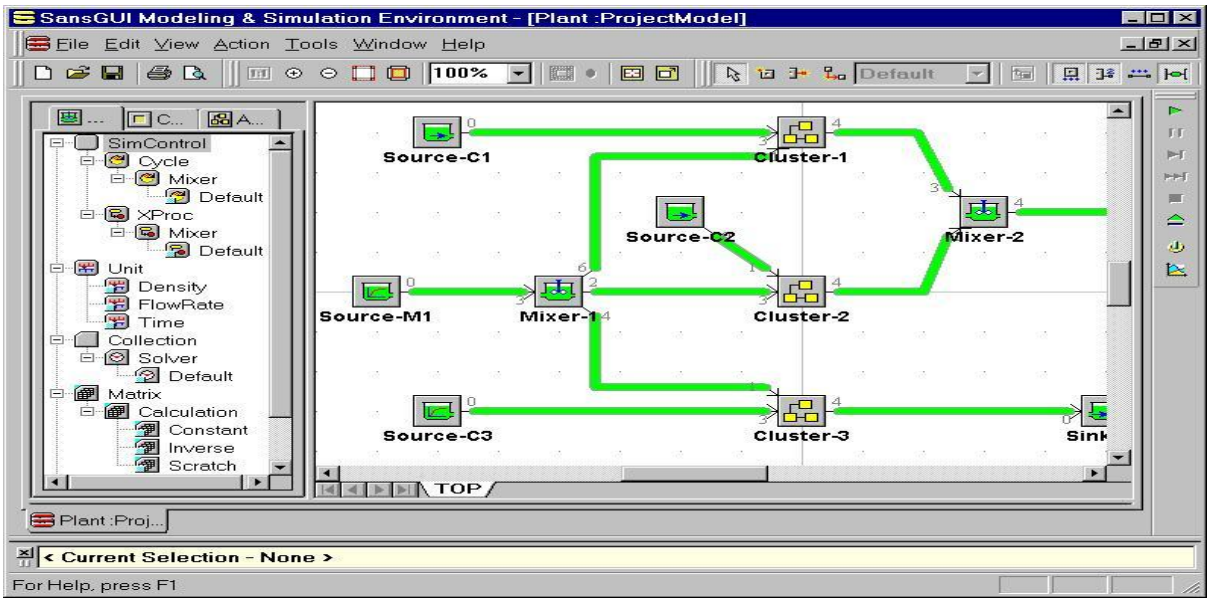


Figure 4.11 SansGUI modelling environment (SansGUI, 2003)

4.7.3.3.1 Object-Oriented Analysis and Design.

According to Softerra (2011) Object-Oriented Analysis and design (OOAD) is industry proven approach for implementing Object-Oriented Systems of high quality.

The OOAD approach comprises of three aspects:

- Object-Oriented Analysis (OOA) which manages the design requirements and all the architecture of the application or system.
- Object-Oriented Design(OOD) interprets architecture used in a system into programming constructs e.g. interfaces, classes and method
- Object-Oriented Programming (OOP) is during which the programming constructs are implemented.

The object-oriented process is identical to traditional approach of system design (Freetutes.Com, 2011). It is a sequential process of designing system with a different approach. The basic steps are: System Analysis, System Design, Object Design and implementation.

- In Systems Analysis, the developer interacts with system user to gather user requirements, which is used in analysing the system to understand functioning of the system, and based on the analysis phase, the analyst prepares a model of the required system which outlines what the system is required to do; at this phase implementation details are not taken into consideration.
- In Systems Design, the overall architecture of the desired system is considered. At this phase the overall architecture of the desired system is organized as a set of sub-systems that interacts with each other. During this phase the analyst considers specification observed in System Analysis as well as the requirement from the end users. The approach can be combined with other models such as V-model and waterfall model in the requirement gathering stage, where user needs are used. This is important if the system developed is to be validated by the user. Here fitness of use is important, and raises the question, 'does the system implemented satisfy user's requirements?
- In OO design, details of the analysis and design are implemented. Implementation of the objects is decided as data structures are defined and interrelationships between objects are defined. The designer decides whether the classes are created from scratch or whether classes can be inherited from them. The data type called Pen can be defined too at this stage, and designers can then create and use several objects of this data type, this is known as creating class.
- During the implementation phase the class objects and the interrelations of the classes are translated and coded using the chosen programming language. OO approach revolves around objects identified in the system. Every object when observed, exhibits some characteristics and behaviour, the objects recognise and respond to certain events. For example, a window on the screen is an object; the size of the window changes when resize button of the window is clicked, and clicking of the button is an event in which the window responds by changing from one size (old) to another (new size).

In developing information systems using an OO approach, the analyst uses certain models to analyse and represent the objects. The models used are:

- Object model: This describes objects in a system and interrelationships between objects. The model observes the entire object, as static attention is not paid to the dynamic nature of the object.
- Dynamic model: in this model dynamic aspects of the systems are represented. It depicts the changes occurring in the state of different objects with the events that might occur in the system
- Functional model: This model depicts data transformation of the system. It describes data transformation, data flow and the changes that occur to data in the system.

Although object model describes the basic element of the system it is the most important, all the models are interrelated and important as they describe the complete functional system.

4.7.3.3.2 Mechanism of OOAD

Wang (2001) explains that the concepts of objects, encapsulation and inheritance are foundation of object oriented systems development this is crucial to understand and express essential and important aspects of an application in the real world. An object encapsulates data and behaviour, this enables the analyst to use the object oriented approach for data modelling and process modelling. Specific objects in a system can inherit characteristic from global instance of a object, for example many types of object may have a name and a creation date, object can inherit characteristics from more than one parent object, by implementing polymorphism, functionality that is conceptually similar among different object is extracted to a global level.

4.7.3.3.3 Unified Modelling Language (UML)

Unified Modelling Language as the name implies is a modelling language, which helps to specify and document models of software systems, including their structure and design (Object Management Group (OMG), 2011). UML can be used for business modelling and modelling of other non-software system. Using anyone of UML based tools, system analysts,

developers, designers and business analysts' language can analyze future application requirements and design a solution that meets the requirements. Booch, Rumbaugh & Jacobson (1999) defines UML as a graphical language that is used for specifying, constructing and documenting software artefacts of a software intensive system.UML is a standard on writing a systems blueprints covering conceptual things such as business processes and system function, as well as concrete things such as classes written in specific programming language, database schemas and reusable software components. Using UML application of any type and combination of hardware, operating system, programming language and network can be modelled. The flexibility of OO approach allows modelling of distribution using any middleware available from the market. Fundamental OO concepts such as, class, operation; encourages UML to fit in OO language and environment such as C++, Java and more recently C#. It can be used to model other non-OO application such as Fotran, VB or COBOL (Object Management Group, 2011). Some tools as suggested by Object Management Group (2011) analyse existing source code and reverse engineer into a set of diagram. Other types of tools designed to work with a restricted application domain such as telecommunication or finance, generate program code from UML, producing most of a bug free deployable application that runs quickly

4.7.3.3.4 Use Case Modelling

Use Case Modelling is used to model the requirements of a product which includes the development of a software application or a system (The Open University, 2011). Use Case models acts as discussion tool between the requirement analyst and stakeholders and provides a common language for specifying functions of proposed systems. Sparx Systems (2011); Arreymbi and Draganova (2008) stated that a Use Case model describes the proposed functionality of a new system and symbolises a discrete unit of interaction between a user and the system, the user maybe human or machine. The interaction is a single unit of meaningful work such as create account or view account details.

The example in table 4.3 demonstrates a Use case model for mobile application system in a classroom environment; curled from Arreymbi and Draganova (2008).

Module administration and support		
Actor	**Use Case**	**Brief Description**
Student	Contact a tutor	A student sends SMS through the system to the lecturer. The system automatically sends email to the lecturer.
Lecturer/ Administrator	Send reminder/alert/ Feedback/grade	A lecturer/administrator sends reminders and alerts such as assignment deadlines, change of rooms, exam dates, coursework grades, and feedback using the system to the student's mobiles.
Lecturer	Send module information	A lecturer sends text, audio or video podcasts, links, glossary using the system to the student's mobiles.

Table 4.3 Module Admin & Support Use case model – Actors, Use case and Description

Each use case outlines the functionality to be built in the proposed system, which can include another use case functionality or extension of another use case with its own behaviour as illustrated in the diagram.

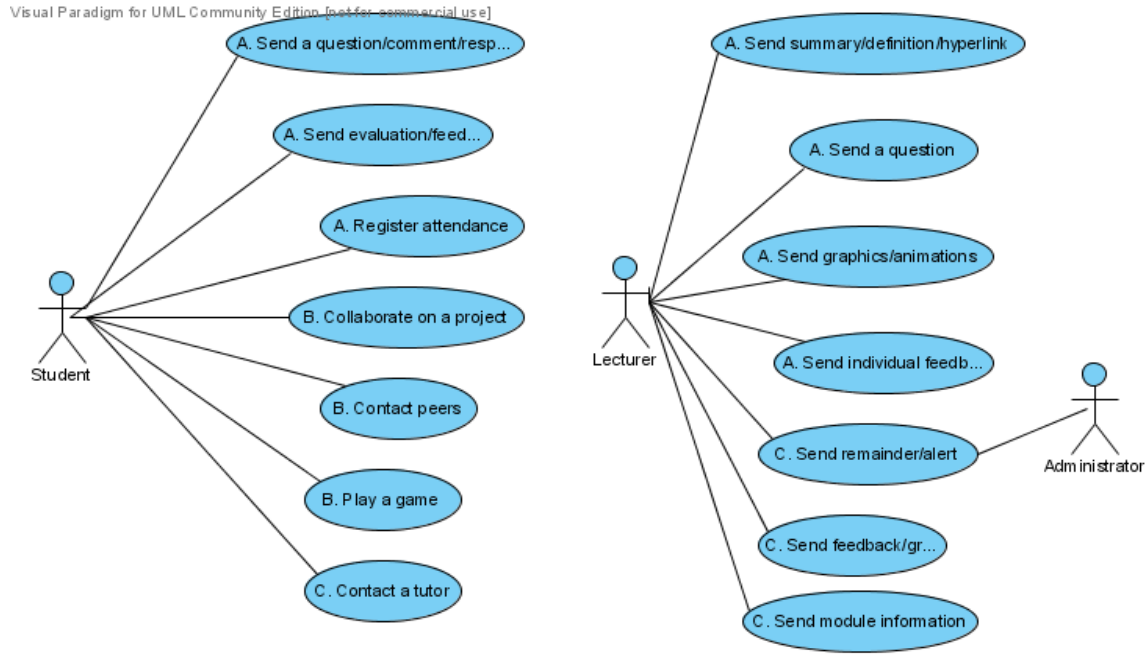


Figure.4.12 Example of Use Case model (Arreymbi & Draganova, 2008)

4.7.3.3.5 Benefits of OO approach

As earlier mentioned, Information systems that adopt OO approach can benefit potentially from the following (OGCIO, 2008):

- Improved productivity: The reuse of existing components can improve productivity in an IS. This also promotes rapid delivery of applications.

This may result from concepts such as inheritance, which allows a class to inherit characteristics of another class as part of its definition. Therefore, the developer does not have to code from scratch, allowing the developer to program faster, and application can be delivered to users on time. This may also enhance usability of the application, and if there are minor security flaws, they be easily be resolved (i.e. specialists can provide a workaround).

- Delivery of High Quality systems: Improved productivity means that the quality of the system can be improved as the system is built in a component manner, which reuses tested existing components.
- Lower maintenance cost: Traceability in OO approach ensures that impact of change can be easily traced thus reduces maintenance cost.
- Facilities Re-use: In OO approach a component system can be developed on a component basis, which enables effective reuse of existing components.
- Managing Complexity: The use of OO approach eases the complexity in managing components: each component is treated as a (black box) encapsulated from others.

The OO approach is applicable to medium and large scale projects. The component based development approach of OO approach is suited to large scaled projects in which applications complex solutions can be broken down into components. Also, the modelling techniques of OO approaches are also suited to model medium to large scale applications that uses complex business logic. Therefore using OO approach for medium to large sized projects with complex logic, involving usability and security, each can be isolated and managed as components within the system development. Therefore, there is a potential balance on usability and security in OO approach As discussed in previous sections usability and security balancing is challenging to implement using for example, waterfall, SSADM and/or V-models which are very structured, rigid and cannot be used on complex large projects. The concepts of encapsulation, inheritance and abstraction are all important in developing large applications with complex business logic.

4.8 Usability and Security trade-offs in systems design process

Security and Usability in systems design and development can be likened to playground see-saw; where a higher pressure on one reduces the functionality of the other so much in measure. Intrinsically, the ultimate is to have the best of both worlds – a fine balance. But how can this be achieved? Many interactive systems (such as ATMs or MTMs, mobile devices, online banking etc.) nowadays have security as an important quality factor as well as a high requirement for usability. However, according to Braz et al., (2007), there is also a common (but false) belief that security is only related to the software systems functionality and that it can be designed independently from usability which only relates to the user interface (UI) component. In fact, the term UI and the way usability is defined are perhaps major underlying obstacles that explain such false beliefs. Indeed, it gives the impression that the UI is a thin layer sitting on top of the "real" system and that usability can be conceived independently from the other quality factors.

Several standards (ISO 9241-11:1998; ISO/IEC 9126-1:2001; IEEE 1061:1998) have all defined usability differently and each has somehow placed much emphasis on different sets of usability factors, such as learnability effectiveness, efficiency, memorability and/or user satisfaction. Therefore a more comprehensive model of usability should include both process-related and product-related usability characteristics such as listed above and including security. Usability is generally, a relative measure of whether a software product enables a particular set of users to achieve specified goals in a specified context of use (Roger et al., 2011; Abran, 2003; Nielson, 1992, 2009).

According to Josang and Patton (2001), "usable-security" or "security usability" deals with how security information should be handled in the user interface. Both usability and security can vary depending on the context of use that includes user profiles (i.e., who are the users), task characteristics, hardware (including network equipment), software, and physical or organisational environments (Seffah et al., 2006). Usability is imperative from the user's perspective (for example, complete a task correctly without errors that can result in a security problem); from the developer's perspective (for example, success or breakdown of a system), and from management's perspective (for example, software with weak security support can be a major constraint to the usability of the system and vice versa). So far, relatively limited amount of work has been done on the area of usability of security systems or secure usability; and in particular, on the parallel but intimate relationship that exists between usability and

security. There is evidence that a system can be abused if it is "overly" usable and not strongly secure. Therefore, what is needed is an approach that can deliver better applications, and interfaces that offer instructions as well as protection (Churchill et al. 2008). Recent research on usability of security such as Flechais et al, 2007; Braz et al., 2007; and Fidas et al, 2010) have attempted to put humans in the loop, and view usability as a key component for accepting security technologies and using them correctly. However, many of the studies are unrealistic as most of the work focuses on the interface between the user and the computer system; aiming at improving the basic usability mechanisms as experienced at the user interface. Little work considers the broader task context – which is the social and organisational setting in which the users tasks takes place.

According to Churchill et al., (2008), employing a social and human-activity centred approach is systemic. Therefore, what is needed is accessing usability along with usefulness: is it serving a purpose? Negotiating, instituting and maintaining real world security problems, procedures and practices is a social activity and the resulting social protocols forms a key component in enforcing security policy (Churchill et al., 2008).

The design of usable yet secure systems has proven to be very complex and has raised many crucial questions such as, how to resolve the conflict between security and usability objectives. How do we ensure adequate usability without compromising on security and vice-versa? The task of building an acceptable trade-off system is not an easy one, but is worth a try and which can totally be based on acceptable compromise. The figure below demonstrates a common solution based on compromise of the aspects of usability and security.

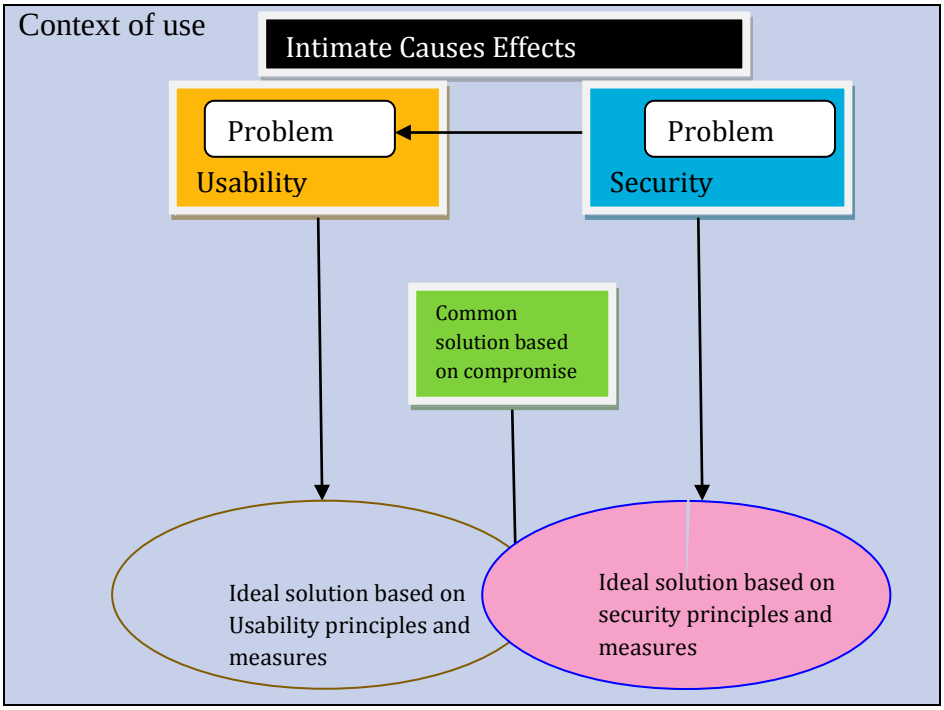


Figure 4.13: Usability and security trade-offs model (culled from Braz et al. 2007)

4.9 Usability and Security Scenarios

Braz et al., (2007) came up with some description that may help define scenarios for usability and security that will help designers to achieve a trade-offs.

4.9.1 Task Scenario:

A task scenario refers to a description of the task at hand including its context of use. The Context of Use (CoU) analysis refers to a broad technique to determine the characteristics of the User, Tasks, and their Environments (ISO 9241-11:1998). The application of the CoU analysis mostly is used as a support to data gather requirements to build the basic components at the early development stages of the application, and also to establish if the end results which consist of effectiveness, efficiency and satisfaction.

4.9.2 Usability Scenario:

A usability scenario details a user problem when doing a task in a certain context. Therefore a usability scenario is a problem related to a task scenario, but it should be well known meaning defined in a usability model, standard or evaluation method.

4.9.3 Security Scenario:

A security scenario refers to a description of a task scenario which includes the use of a particular security mechanism. A Security Scenario can be tangible or intangible. A Tangible Security Scenario (TSS) includes physical infrastructure such as controlling user's access to buildings and facilities using Biometrics, or sending a silent alarm in response to a threat at a MTM, etc. An Intangible Security Scenario (ISS) includes data or other digital information, for example, a user who enter sensitive information at registration in order to purchase a concert ticket at a MTM. A Security Scenario might be (or not) a combination of TSS and ISS (Braz et al 2007). The Security Scenario has been classified accordingly as indicated, by the overall impact of the security risks of the security mechanisms related to the system's owner; such as *High Security Impact, Moderate Security Impact, and Low Security Impact*.

- A *High Security Impact* refers to the confidentiality, integrity, or availability of the security mechanisms, and it may cause severe or catastrophic loss to the owner's system (e.g., authentication credentials like private cryptographic keys, and hardware tokens)
- A *Moderate Security Impact* refers to the confidentiality, integrity, or availability of the security mechanisms, and it may cause a moderate loss to the owner's system (e.g., data on internal file shares for internal business use only)
- A *Low Security Impact refers to the i*mpact on the confidentiality, integrity, or availability of the security mechanisms and, may not cause any significant financial loss, legal or regulatory problems, operational disruptions, etc. to the owner's system (e.g. public cryptographic keys).

4.10 HCI Design criteria (a security perspective) (Muñoz-Arteaga et al., 2009)

- **Visibility of system status**: The UI must inform the user about the internal state of the system (e.g., using messages to indicate that a security feature is active, etc.). The warning or error messages must be detailed but specific including a suggested corrective action for some security problem, and links to obtain additional information or external assistance.

- **Aesthetic and minimalist design**: Only relevant security information should be displayed. The user must not be saturated with information and options, and the UI must avoid the use of technical terms as much as possible. The security UI must be simple and easy to use, maintaining a minimalist design.
- **Satisfaction**: The security activities must be easy to realize and understand. Without the use of technical terms in the information showed to the user, in some cases, it is convenient to use humour situations or figures to present important security concepts to the user in an entertaining manner.
- **Convey features**: The UI needs to convey the available security features to the user clearly and appropriately; a good way to do it is by using figures or pictures.
- **Learnability**: The UI needs to be as non-threatening and easy to learn as possible; it may be accomplished using real-world metaphors, or pictures of keys and padlocks. The meaning of these metaphors may be incorporated to the security interface indicating users how to easily use the specific security features.
- **Trust**: It is essential for the user to trust the system. This is particularly important in a security environment. The successful application of the previous criteria should typically result in a trusted environment. The concept of trust can be adapted for the HCI criteria of trust (Johnston et al., 2003) to ‘‘the belief, or willingness to believe, of a user in the security of a computer system”. The degree of trust that users have in a system will determine how they use it. For example, a user that does not trust a web site will not supply their credit card details. In the similar manner, D’Hertefelt (2000) identified six primary factors (i.e. fulfilment, technology, seals of approval, presentation, navigation and brand) that convey trust (Atoyan et al., 2006) in an e-commerce environment. When these concepts are applied in a security environment using the HCI criteria, it is possible to achieve the user trust in the specific system’s security.

4.10.1 Usability factors (Braz et al., 2007)

Below are some of the usability factors that can be considered during the design of usable-security systems:

- **Learnability:** The features required for achieving particular goals can be mastered

- **Efficiency:** The capability of the software product to enable users to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use
- **Accessibility:** The capability of a software product to be used by persons with some type of disability (e.g. visual, hearing, physical impairments etc)
- **Usefulness:** Whether a software product enables users to solve real problems in an acceptable way.
- **Satisfaction:** The subjective response of users, while using a software product (i.e. is the user satisfied?)
- **Productivity:** The level of effectiveness achieved in relation to the resources (i.e. time to complete tasks, user efforts, materials or financial cost of usage) consumed by the users and the system
- **Safety:** Whether a software product limits the risk of harm to people or other resources, such as hardware or stored information
- **Trustfulness:** The faithfulness a software product offers to its users
- **Universality:** Whether a software product accommodates a diversity of users with different cultural backgrounds (e.g. local culture is considered).

In their quest for answers, researchers Braz et al., (2007) adopted a new usability and security inspection method called Security Usability Symmetry (SUS), based on the *Heuristic Evaluation* method by Nielsen (1992). The method aims to help usability specialists and security designers to design/inspect/evaluate an interactive system to identify any usability and security user problems and check for conformance with its corresponding usability criteria and security aspects of the system. These usability criteria and security aspects can be used to *guide a design decision* or to *assess a design* that has already been created. Nielsen (1992) reported that usability specialists were much better than those without usability expertise at finding usability problems by heuristic evaluation. Moreover, usability specialists with specific expertise (for example. security) did much better than regular usability specialists without such expertise, especially with regard to certain usability problems that were unique to that kind of interface. In the endeavour SUS is developed as a security usability inspection method for evaluators who have knowledge of usability and also computer security. In SUS, a solely usability specialist can also work in pair with a solely security specialist. The SUS can help also to develop a system profile that will impact on

whether or how usability and security aspects will be implemented in the system. A system profile might present the profile that is used by systems designers to determine their specific characteristics and needs. Prior to evolving into the iterative design phase whereby a product is designed, modified, and tested repeatedly, it is critical that usability specialists and security designers understands its own specific requirements and goals for the system. The SUS guide focuses on the following key areas: Usability and Security requirements, Interoperability, System Application, Technology, and Resources.

According to Nielsen (1992), systems usability problems can greatly be reduced through the severity rates where we are able to identify those problems that should be tackled and fixed. The ratings also assist in the allowance of resources for treating the UI problems. Nielsen, (1992) suggested that, severity consists a combination of three elements: frequency ranges (i.e. from common problems to unusual ones), impact (i.e. establishes the ease or difficulty with which a user gets over a problem), and persistence (i.e. ranges from just one problem that might be surmount to the problem that constantly replicate itself becoming annoying to the user).

4.11 Summary

In this chapter we have considered various development methodologies and have distinguished between development lifecycles and ISDMs. What we have described in this chapter are the components that make up an ISDM in the sense of SDLCs which are sometimes called methodologies so we should not get them confused. The Security and usability aspects have been highlighted as a means of attempting to understand the intricacies involved and to find better ways of a trade-off (acceptable compromise) approach to designing a usable yet secure system. Also, based on the scenarios, we attempt to address the intimate relationship between usability and security. Therefore, to be able to design and develop reliable, effective and usable security systems, we require specific guidelines that take into account the specific aspects of usability mechanisms and their potential consequences on security as needed.

Chapter 5 Identification of the canonical set of issues relating to security and usability in IS design

5.0 Introduction

Security usability according to (Josang, 2001) deals with how security information should be handled in the user interface of systems. Both security and usability can defer depending on the framework of use that is included in the user profile. Usability is very important from the users view point (e.g. success or breakdown of a system), and completing a task correctly without errors that can result in a security problem; from the developer and management perception. For example, weak security on software can have major influence on the usability of a system and vice versa. Many analysts (Sasse, 2011; Braz, et al., 2007) have all highlighted the many problems that exist in the use, manipulation and management of information systems, and have suggested various approaches in trying to resolve the issues and achieve a balance with the security and usability aspects of information systems design and development. However, the way systems have been designed and developed have not truly solve mankind problems as hitherto thought (Landauer 1995); some have in fact made the situation worst due to the diversity of requirements from the various stakeholders. It is evident nowadays that, users want systems that are very friendlier, flexible, easy to access, robust and also protect their privacy. The management on the other hand want systems that can provide confidentiality, integrity and readily available to meet the business needs, while protecting vital business information and the infrastructures for competitive advantage.

This section will attempt to make some suggestions in finding better ways of addressing the issues identified, by proposing a framework that can be used to design a trade-off usable-secure information system. The framework consists of a set of factors, which if carefully considered and implemented during the various stages of systems design and development, can be very appropriate to produce a usable-secure or secure-usable information system. The process we envisage involves looking at the effect of what we will call; the Social, Economic and Technical (SET) factors within the operating environment of the system to be developed and the overall strategic implication. These are the factors that will be considered in the design and implementation of usable-secure information systems. Later on, it will form the basis, as we explore in much detail, how the Social, Economic and technical (SET) factors can be used to design a simple ‘balanced’ system for usable-security or secure-usability. For example, section 5.4 demonstrates a simple model for an air travel information system with

regards to security and usability. But before we delve into that, it will be appropriate to look at the characteristics of usability and security in order to identify where their inherent problems lie.

5.1 A Standards view on usability and security

Usability has always been a key issue in systems design and development. The ease of use of any system is what makes it effective. Standards Organisations as well as researchers have identified some viewpoint in relation to usability of systems (Sasse, 2009, 2011; DeWitt A, 2007; Cranor and Garfinkel, 2005; Braz, et al., 2007), and have put together some characteristics of usability to include security. The table 5.1 list some of the standards where security has been included within their usability framework. These standards measure that, good usability is a very important condition for human security in a critical system such as major infrastructures and as such, the perception of security has been adopted within the standards by many organisations (Abran, 2003). Therefore, within the proposed model, we will be adopting this perception of security. The table models the relationship between security and usability. The key characteristics of the usability problem (represented via a usability scenario) related to security are briefly described thereafter.

Task	**Usability**	**Security**
ITSEC: Information Technology security Evaluation Criteria	IEC 300 V1.2:1991	It present software as a security critical
International Standards Organisation	ISO 9241-210 (2007) formerly 13407:1999	It describes human centred design as a multi discipline activity incorporating human factor and ergonomic and the technical knowledge with the objective of raising efficiency and effectiveness , improving human working condition and opposing possible unfavourable effects of the use on human health, security and performance .
	ISO/IEC 9126-1:2001	It defines security, which is a sub characteristic, as a set of software attribute which relates to its ability to prevent unauthorised access, whether accidental or deliberate to

		program of data.
Federal Aviation Administration (FAA), 1998	FAA, 1998	Security is a characteristic of the CHI which is particularly important in an industrial context.

Table 5.1: Security as usability characteristics (culled from Braz, et al., 2007)

5.2 Defining usability for security

Usability for security is a unique problem, and we have approached it from different perspective, by looking at the properties of the problem of security usability,

5.2.1 Some properties of the problem of security usability or usable Security

- The abstraction property: This is a system (e.g. security policies) of abstract rules for deciding whether to give access to resources. User interface design (UID) for security will need to take this into consideration.
- The lack of feedback property: Dangerous errors must be prevented, therefore it is imperative that good feedback is provided to the user, however, providing good feedback for security management is a difficult problem. A systems security configuration is usually complex, and attempts to summarize it are not adequate. In fact, a workable and/or correct security configuration is the one which does what the user "really wants", and since only the user knows what that is, it is hard for security software to perform much useful error checking (Whitten and Tygar, 1998).
- The weakest link property: it is a known fact that security of networked systems is only as strong as its weakest section. If an attacker can exploit a single error, s/he can cause great damage. This means that users need to be trained or guided in all aspects of security of their system (Cranor & Garfinkel, 2005).
- The unmotivated user property: Security is usually a secondary goal to users (Cranor & Garfinkel, 2005). Users do not want to sit on computer to manage security, but to carry out a task such as browsing or sending an email; and they

want the system to explicitly protect them. Most users prefer to focus on their operations rather than the certificate of server.

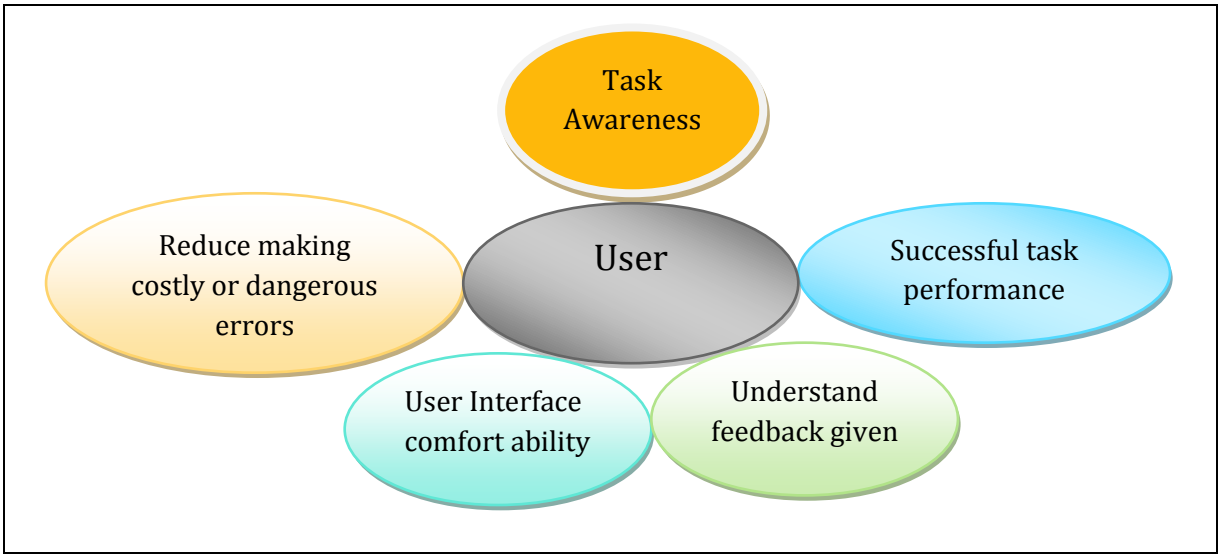
- The limited human skill property: Humans are not general purpose computers and are limited by their intrinsic skills and abilities, rather than approaching a problem from a traditional authentication based security framework (e.g. what can be secured?) A usable design must take into account what humans do well and what they do not do well.

5.3 A functional definition of the usability of security.

Putting these properties into clarification, we proposed that security software is usable if the people who are expected to use it meet the following conditions; that they are:

- Adequately comfortable with the interface
- Able to figure out how to successfully perform those tasks and don't make dangerous errors
- Reliably made aware of the security task they need to perform
- Educated about the security tool
- Able to understand the feedback given

Figure 5.1: Diagrammatic representation of functional definition of the usability of security.



5.4 A usable security Model – Business perspective

A usable-secure model can be used by any business or organisation to express the business rules to be used in their computer systems. These rules can come in form of usability policy, security policy or Usable security policy; which outlines a set of rules to be followed in order to ensure the effective use and safety of the system. The model can be different in respect to what purpose or what is it being used for. Similarly a usable security model can be designed for an information system to provide safety, easy access and use, to the data, software and hardware of the system. The example below shows a simple usable security process for travelling through security check points for a journey by flight or train.

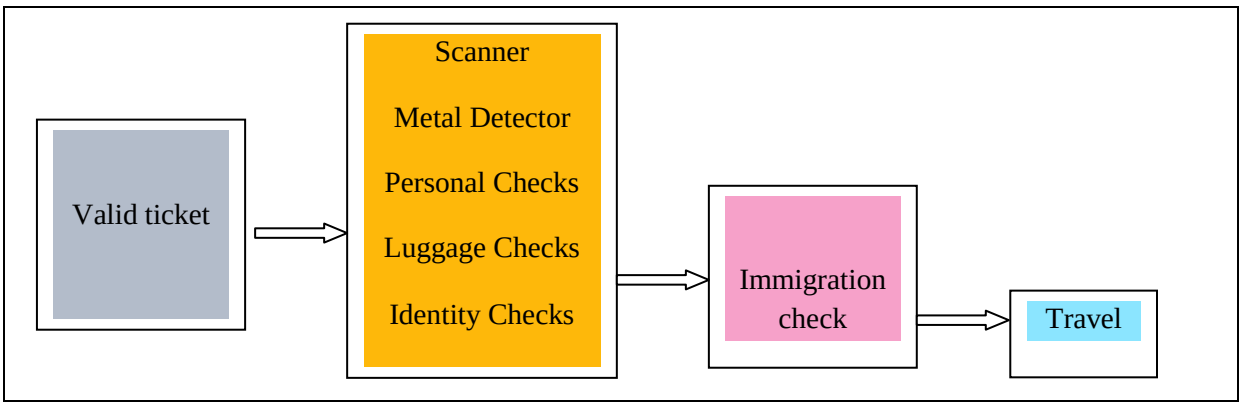


Figure 5.2 Example of simple usable security process relative to a journey.

5.5 The Proposed Model/ Framework

5.5.1 The Influencing factors

There are many factors influencing usability and security in the design and development of information systems. For example, the Must have, Should have, Could have and Won't have (MoSCoW) features and the proportion of each of the aspects is required to guarantee good effective usability and strong security at the same time (a convenient balance).These aspects could be detailed and captured during all the stages of development but more during the requirement analysis. There are many factors to consider when attempting to develop a usable –secure information system. They can be classified under the following three broad categories:

5.5.1.1 Social factors

The social factors are everything to deal with social and systems dynamics; and includes the behaviour of employees, trust in the organisation, skills and ability, experience, cultural dynamics change resistance and management, training needs and facilities, knowledge and social risks associated with internal and external environments. The system can be modelled around these factors during the requirement elicitation and/or problem definition.

5.5.1.2 Economic factors:

These are factors Business System Options (BSOs) that are associated with the use of security applications, technologies and used, to improve the trust in the organisations economically, such as decisions to buy or not to buy and/or implement. It may involve cost of ownership, risk elements, and cost of adoption, implementation and /or loss due to any malfunction in the system.

5.5.1.3 Technical factors:

They are Technical System Options (TSOs) which includes the use of computer systems, Hardware, firmware, software applications and other related technologies to enhance security. Some of this is to deal with the systems dynamics. Examples include but not limited to training needs, system features, ergonomics, and quality of software/hardware, management of information security systems, technical knowledge, available tools, in-house technical development/implementation, testing and maintenance skill sets etc.

The Social, Economic and Technical (SET) factors form the basis for the consideration of reflective approach to design usable-secure system. The SET framework will be explained in more detail in the next section. As shown in the figure 5.3, the fieldwork or environment is the area of operation, where the actual process of ensuring the system usability and security is to be performed. This includes the use of software tools, personal observations, questionnaires, and checklists etc., to investigate the effectiveness, efficiency, safety and robustness of the information system. Here, we will look at each of the SET factors in much detail.

The SET Framework

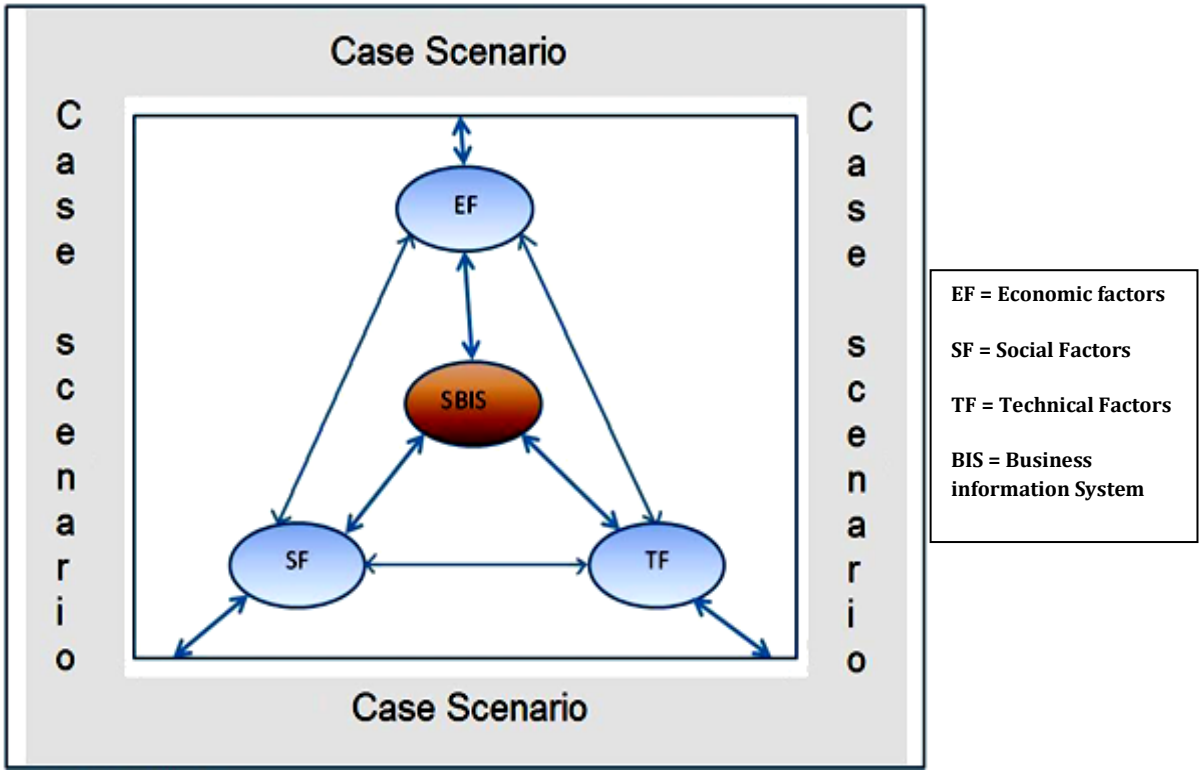


Figure 5.3 The Business Environment model

5.5.2 Rationale for the Framework

The increase use of IS in business has been due to advancements in new technologies. Businesses have found technology a weapon to use to improve their environment for competitive advantage. Many organisations are now investing millions of pounds in developing better or "smart" Information Systems. However, with any increase in the adoption of new technologies, there are always the potential danger elements of ease of use and also that of security associated with the technologies; and this can make the organisations become vulnerable to malicious attacks, as they deploy the systems applications.

However, the fast changing technologies landscape also requires that organisations adopt highly secure and user friendlier model(s) or approach(es) to ensure the security and usability of the IS infrastructures.

The proposed SET framework can be very flexible and encompassing. There is dynamism in the process as users and other stakeholders are highly engaged and interactive throughout. The framework takes into consideration the ability, skill sets and experience of the users /

stakeholders in the functionality and use aspects, as well as the safety use and protection aspects of the system. Users are very involved early on in the process, from the requirement capturing, analysis to design, development, testing and operations and maintenance of the information system.

Therefore, in designing IS for good effective usability and strong security, the proposed SET framework can be easily implemented because, it takes into consideration the social elements of how users generally interact with the systems (by means of interview, observation, questionnaire, focus group etc.); the economic elements of associated costs (for example, facilities, training, awareness and policy developments, technology purchases etc.) and the technical elements relating to the technology provisions, operations and maintenance of the system. The knowledge gained from requirement analysis is fed into the system and continually monitored and reviewed. Therefore the SET factors require the provision of high level security and usability in IS systems and this makes them arguably more immune to changing technology use and increasing security threats.

5.5.3 The Social, Economic and Technical (SET) Framework explained

5.5.3.1 Technical Factor solution

Technical factors include the technical system options (TSOs) and as described here apply to the hardware, software and other related technologies used in the business processes. Here, the logical components form the core components of the security and usability solutions. The development is not yet build but rather an application of the logical components in a different way to what exists. The aim of this technical solution is to demonstrate that using a holistic easy- to-use security mechanism, it is possible to effective and robustly protect the IS infrastructure. These aspects can be dealt with using the same technology, and with some management control, IT risk analysis and policies to effectively and efficiently deliver usable-security solutions. The security and usability applications policy and models have been elaborated in more details in other sections of the thesis. Here, we analyse some cases of technical components found in information systems.

Databases: Database/ Data warehouse is the hardware in which all the data is stored. These are the main back-end source for any information system. All the front-end operations and user's actions on the front end screens can result to the adding/deleting/updating the data in the databases.

Software: The software is a set of programs which is used for a specific purpose. They are a set of programs when executed performs the action they intended to do. For example, an antivirus software program scans the whole system for harmful programs, when the scan initiated. Similarly, when a reservation is placed on a flight, the details of the passenger are stored in the database and a booking confirmation is sent to the passenger's mail id. Examples of software Office programs include Word, Excel etc.

Hardware: It includes all the physical assets related to technology. The examples include printers, monitors, keyboards, databases servers etc. They are platforms on which the software programs are operated by the staff. For example, to write a letter in MS word, we need a computer, monitor and a keyboard. So, hardware is just a device used to run the software. LAN/WAN can also be grouped under hardware.

These are some main technical factors that can be seen in any business information system. Ensuring the security for these technical factors is crucial step being faced by many organizations. Let us now look at some of the main usable and/or not-so-usable security applications available for these aspects:

Antivirus/ Anti-spam: These include different software applications which protect the system from virus, worms, malwares and potentially harmful programs. These programs can be used in small organizations, but when large information systems are considered, special programs are needed for protection from the harmful programs.

Firewall: A firewall is a part of the network, which prevents unauthorized access in to the system and allows authorized communication channels. It is one of many preventive measures protecting the information systems in the network. Generally a powerful firewall, software or hardware does not allow any hacker or unauthorized user to access the information system.

Encryption / Decryption Techniques: This is the technique where, the data from the sender is converted or encrypted in to a different format before transferring it to the receiver over the network. When the data is received at the receiver's end, the data is then decrypted to the original format which can be read by the receiver. This is one of the major security applications being used in many business information systems.

Authentication: This is the process of checking the authenticity of the user, if he/she is genuine user or not by some kind of process. Generally in most applications, login id and a unique password are used for authentication, but with the increasing security threats, and the many passwords which are easily forgotten, other biometric forms of data such as retina scan, finger scan, voice recognition and pulse scan are the new authentication technologies being implemented and used.

Access Control: Even though, the user is allowed to access the system, the user may not have all the permissions or rights in accessing the system. For example, in a desktop computer environment, a user logged in as a guest may not have the permission to access the D drive. So, we can restrict the permissions of the users depending on their roles in the organization. Access can be controlled by using some programs too. For example, the log files records each and every action on the system. So, a program can be written in such a way that, if any action is being performed by the user who is not authorized to do so, then the system automatically log offs or shuts down.

Backups: The data in the database is backed up regularly to prevent the loss or damage of data. There are different back up techniques currently being used in many organizations.

Standards: There are some securities standards developed by authorised organisations and research agencies as to how organisations implement and use their information systems. By adhering to these standards the organizations can ensure the quality, safety and security of their information systems.

Laws & Regulations: There are some laws and regulation on the safety use of information systems and related technologies. There are some rules on the use of data by the organizations. They provide security laws which explain the actions that would be taken in

case of any the security breaches by different personnel. Figure 5.4 demonstrates an example of a security framework

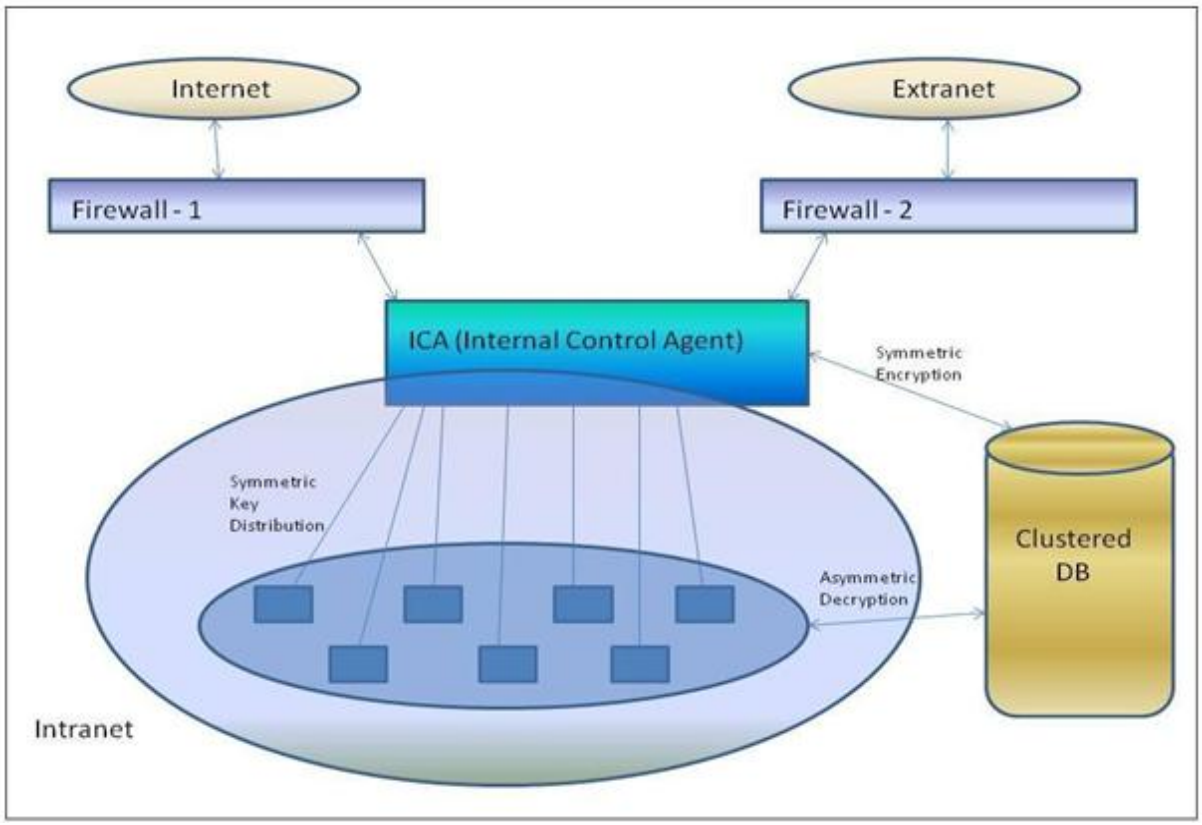


Figure 5.4 Security Framework for technical factor solution

In this situation, the main channels in which the system is involved or linked with, are internet, intranet and extranet. Internet is the network from which all stakeholders can communicate with the information system of the organisation. In extranet, the trusted partners of the business organisation can communicate with the information system and in the intranet, only the employees of the organisation can communicate. In the security framework, no user from the external organisation has a direct access to the database. There are two firewalls used to separate communications across internet and for communications across extranet, to ensure the security of the information system. The firewall-1 is very powerful and prevents any sort of unauthorized access. Firewall-2 is also a powerful program which monitors partner's actions and stops the access in case of any improper use. There is an ICA (Internal Control Agent) in-between the internet, extranet and the information system, which controls the communications across the network. The cryptographic security system is used

in the intranet, where the Symmetric encryption and Asymmetric decryption techniques are used for accessing the data from the database. Initially, the request from the local users is received by ICA and it is encrypted and sent to the database; then by using the Symmetric key, the local users can access the data from the database. All the operations in the system are controlled by ICA.

5.5.3.2. Social Factor solution

The social factors include the aspects such as user profile, trust, employee behaviour, ability, skills/experience and organisational culture and social norms, working conditions, policies etc. These are key areas in the conceptual framework. For example, the user profile determines the navigation environment, user type and activity. The user profile is dependent on the environment of operation. The system specifications are made up of the components which delivers the user service. Al Nabhan et al., (2009) reported how a user profile could determine user preferences and privacy settings. These social factors can sometimes be a usability and security threat to the organisations. If the employees are not happy and/or trustworthy, then no matter how efficient the technological applications are, the employees will be less motivated to use or manipulate the system proficiently enough; and sometimes may be motivated to compromise or steal the data or information from the system for personal gains. The employees must respect the organisation and the management has to take the necessary steps to educate the employees of their obligations legal or otherwise and the consequences of any breach. Frequent monitoring of situations, counselling and motivation are essential to keep employees' mind focused and engaged on the main tasks. It also reduces stress and time wasted in use of the system, and prevents employees from carrying out criminal activities or intentions. Trust is very important in organisations; employees must trust the effectiveness of the systems and management they work with, and the management have to trust their employees to use the system safely and more securely. If there is any mistrust, it can lead to system abuse and if not curbed can lead to soaring cost to the business.

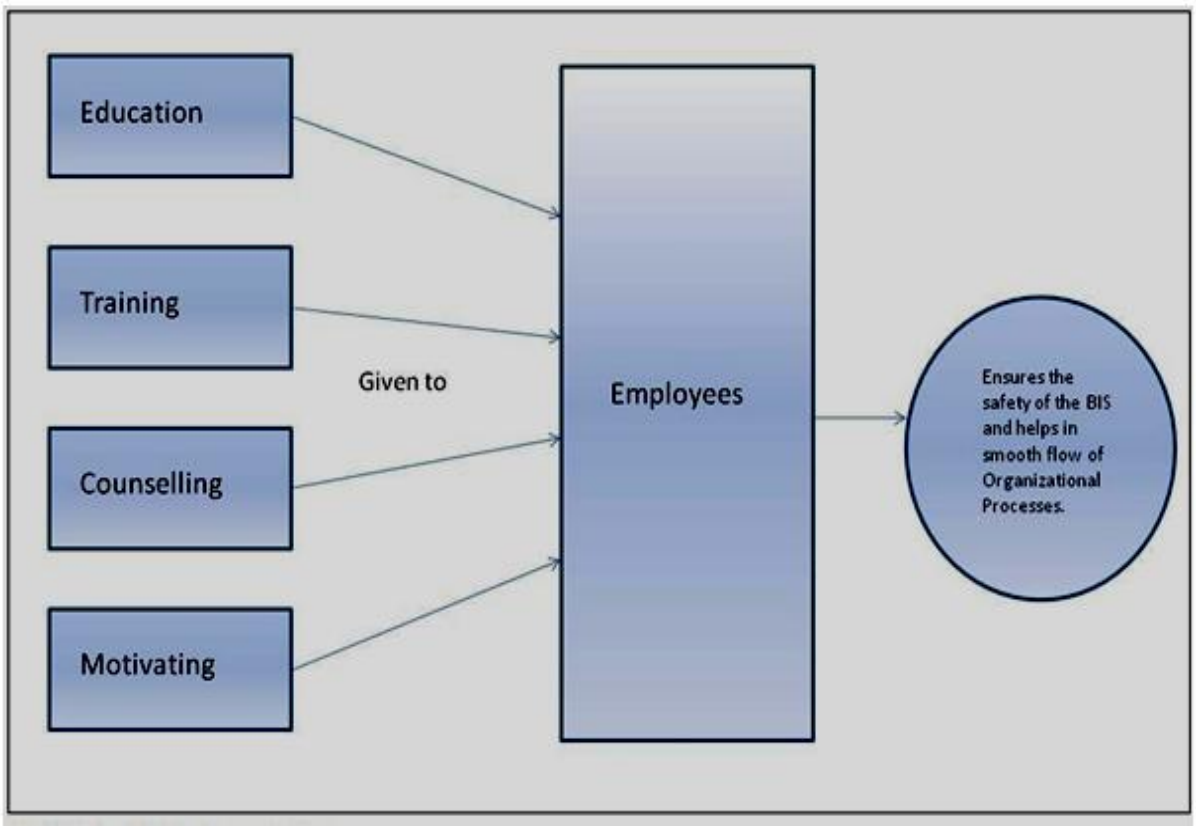


Figure 5.5: Social factors in business organisations

Employees are the main assets of any organisation. So, controlling their behaviour using motivational techniques can be very profiting to the business; and possessing good employees are the main factors which can determine the success of an organisation. Therefore, while giving priority to the use of technology, it is very essential to also consider the employees or users of the systems, who form part of the organisation's culture. Employees may be good and proficient but reliant on technology may lead to mis-control or misguidance by the technology or people from external or internal environment. Recently, there have been many cases in the News about inappropriate use of technologies by employees, which are as a result of the inadequacies of the systems and organisational management policies. Therefore, the management has to have in place adequate mechanisms to monitor the activities of their employees but should take every action not to infringe on or do anything that will reduce trust in the employees and vice versa.

5.5.3.3. Economic Factors

These are factors that include the business system options (BSOs) and are associated with issues such as costs of development and adoption, use and maintenance of applications, management, quality assurance, technologies and facilities used, business needs and alignments procedures to leverage and improve trust in the organisations economically, such as decisions to buy or not to buy and/or implement. It may involve cost of ownership, risk elements, and cost of and /or loss due to any malfunction in the system disaster recovery planning etc. In this thesis, we will deliberately not delve into the issues such as business strategy, ROI and/or Return on Capital Employed (ROCE) as they will be out of the remit of the research. However, it is worth noting that in the process of selecting or making decisions on which methodology to use, we have used Costs-Benefits analysis model to justify the decision. For example, not every organisation can use same type of security systems or make use of technology the same way. One key element in this is the cost of adoption and/or change management and employees resistance to the technology. An example case study is that, many Universities in England have a swipe card security system. For any user to gain access, they need to provide and swipe a valid ID card on the system. The system more often, is not easy to use and tend to be failing constantly, with enormous usability and security implications. But will the University management deploy a more robust system that is more usable and better secure, such as fingerprint scan authorization system for access? The answer is No, because, of the cost implications of deploying new technologies. Most times when they deploy new technology one of the main things considered is its cost effectiveness. Therefore, for any system to work effectively and efficiently, it is important to ensure that the applications are very usable, provide better security and cost effective in the long term. This is one of the most essential factors to be considered.

5.6 Case scenario

Field work includes different activities to investigate the efficiency and effectiveness of security applications deployed in the Business information system with respect to Social, technical and economical factors. The different issues such as employee's behaviours, ability, skills experience, risks awareness and, threats to the system from both internal and external environments will be considered during the fieldwork. The results of the field work gives the effectiveness of the security systems being used and the suggestions are given for improving

the quality and efficiency. Just like an audit exercise, the field work generally includes the following activities:

Personal Observations: The personal observation of the different system related issues can give good results, in terms of the ability and effectiveness of use of the applications. It is a simple exercise that is easy to perform and easy to analyse.

Interviews: The interviews are taken from the different level employees of the organization related to the information systems security and usability issues.

Questionnaires: These can be used to assess the business environment, ease of use, product quality and the potential threats to the business information systems. For example, consider the following questions:

Do you have employees working remotely?

What percentage of your services are web based?

How many branches do you have and how are they linked to the main office?

What communication channels do you use while communicating with members of organization, partners and other stakeholders?

These questions are used to assess the risks to the information system and then the quality, effectiveness, efficiency of the Information System is analysed with respect to these threats and risks and use;

Checklists: Checklist can also be used to test the security and usability related issues of the information system. The following is an example check list:

Control	Comment
Virus checking software available and being used effectively	Yes
Virus checking software regularly updated for quality and robustness	No
Virus checker is used by all the employees	No
Virus checker can easily be configured	No

Table 5.2 Security/usability control checklist

CAT (Computer assisted tools): The software tools for example, SystemSkan can be used to test the different security aspects of the information systems across the networks and internally.

The results of the field work are then analysed to design the security and usability report. The report contains the information system security issues, effectiveness of use, and suggestions to improve the security and/or usability, if there is a need to do so. After implementation of the SET framework, the results of the implementation are analysed first to obtain the security and quality status of the system. And then based on these findings, some recommendations for improving the safety and usability of the system are set out by the security and usability professionals, to be fed back into the system and updated.

5.7 Summary

In this chapter the canonical set of issues affecting usability and security in information systems has been identified, explored and categorised under the SET factors, (Social, Economic and Technical):

- **Social factors:** They include the behaviour of employees, trust in the organisation, and social risks associated with internal and external environments. They involve a reflection on user's view and experience (Bryman, 1989; Saunders et al., 2000) and the social dynamics within the organisational environment
- **Economic factors**: These are factors that are associated with the use of security applications, technologies and how they are used to improve the trust in the organisations economically, for example a decisions to buy or not to buy and/or implement
- **Technical factors**: includes the use of computer systems, coding, software applications, systems integration and other related technologies to enhance security and usability.

The above factors have influenced the development of a framework that attempts to provide a "balanced" approach in addressing equally the aspects of security and usability in IS design and development. The SET framework can be said to be grounded in theory (Saunders et al. 2000), in that the approach captures the social reality of stakeholders and the theoretical requirements that emerges and the technical aspects are covered more in practice; therefore is grounded in reality. The relationship between social dynamics and the technical factors

brings a better understanding of the system by users (Saunders et al., 2000), with regards to usability and security issues. The emphasis of this grounded theory approach is for users to derive meaning from the system being developed. The process allows designers to produce more manageable and focused systems. The SET framework as suggested, if properly implemented, will attempt to tackle the issues in relation to developing usable security or secure usability.

Chapter 6 Summary, Recommendation, Conclusion and Future work

6.0 Introduction

This section will provide a summary of the issues discussed in this report and use the findings to make some recommendations and draw some conclusions to the research. Also, herein will highlight the limitations of the thesis and indicate areas for future work with regards to finding better ways of designing and implementing information systems that are not only secure and robust but also easy to use or manipulate.

6.1 Summary

In this research, the main objective was to look at better ways of improving the security and usability of information systems. The research set out to investigate the reasons behind IS failures, the problems therein, and in the process realised that, most information system failures are as a result of poor design and development process; which together with the way the systems are implemented and operated; form a cocktail of disastrous endeavours to the usability and security of the systems. Most importantly is the fact that, the human factors have contributed enormously to many systems failures and disasters in organisations but the reasons of how and why this happens still leaves wide open points for debates, from both a corporate strategic perspective to individual users perspective. It is from this premise, that we investigated and analysed the various systems development methodologies to see how they address the issues of security and usability within them, following the various SDLC stages/approaches. Also discussed in this thesis are some of the challenges faced by developers of Information Systems in trying to meet users' needs; and provide for effective interactivity strong security; and much with the understanding that, Information System was developed to help solve certain mundane tasks in businesses and society. However, most Information systems have failed in their quests to facilitate human and/or business endeavours due to inadequate design and development processes. From the analysis of ISDMs, it is very clear that existing methodologies do not adequately provide a balance on the issues of security and usability. The resultant effect is always a trade-off between the two aspects, where either security or usability is compromised for the other in the development of IS. However, the processes of task analysis, iterative design, trial use, and evaluation can help make information systems useful tools for business. However, a probable solution may come from a combination of methodologies such as SSM and SSADM, may attempt to overcome

this hurdle, and provide a good balance between these two conflicting but very necessary aspects in the design and development of usable-secured information systems. The focus therefore should be on User Centred Design activities, and with very clear guidelines on how to use the systems for effective and efficient outputs.

Looking at the various options, we have proposed a model or framework that; if implemented correctly, it is assumed, will better tackle the issues of security and usability within IS design and development. The approach effectively addresses the usability and security requirements by recommending that these aspects be looked at early enough, at each stage of the system development lifecycle in order to come up with a more effective and efficient usable yet secure IS.

The approach albeit, has neither been used to fully develop and/or evaluate a functional system, nor to determine how effective it is and therefore weak in that respect; therefore this research report has been constrained to analysis only. This analysis will form the basis for future research and therefore expand the capability. However, we have presupposed an approach that will theoretically, but adequately address the aspects of security and usability within its framework. We also anticipate that further works will be carried out in this area in future to find better ways of using the proposed framework to design, implement field tests and evaluate the approach, to see whether a true balance can be achieved with the vital but conflicting issues of usability and security.

User feedback acts as an important aspect for the software development companies, because the users' feedback contains the defects and the positive aspects of the software system that determines the quality. This helps the software developers to improve on the usability and/or security of the software with better functionality. Many large organisations such as Microsoft and Apple have been using this technique to develop and improve on their products. At first, the products are launched and as many services become successful in the market, patches are then made to cover some defects; later the updated version of the product reveals the defects and the lack of features in the previous versions, and the cycle continues. These situations are happening daily, because after products have been released, it is left sometimes for the many users to review the products, these reviews provide details of the applications, uses and features / issues about the product to the developers. Later these issues will be patched or rectified by the software developers and then released again as new/ latest version. From the perspective of problem solving in system use; according to Jayaratna (1994), when users find difficulty in working with systems due to features such as poor usability or increase security,

they tend to switch off or look for ways to work around the system. Therefore, from a problem solving perspective, systems need to be designed and developed to efficiently meet users' needs and giving them confidence in the knowledge that they can use the system effectively and in safety and security. This is the premise on which the SET framework is based.

6.2 Recommendations

From all the critical analysis carried out, it can be seen that, none of the existing approaches or IS design methodologies covers all the hard and soft aspects of security and usability. Most of them for example SSADM, and SSM, tend to focus more on one aspect only and therefore not suitable enough in addressing the needs of IS usability and security. Although some models have been proposed such as, AEGIS (Flechais et al., 2007) and SQUARE (Mead, et al., 2008) in an attempt to redress the issues; on many accounts, the models have tended to be inefficient and not flexible enough to adequately address the balance of a strong security and good usability. For example, the AEGIS approach assumes expert knowledge of the users / stakeholders, and relies heavily on them to model the system. Also the methodology puts strong emphasis on integrating security in the requirement process, but relies heavily on a single individual to acts as security lead in the project and this same individual is also responsible for documenting decision making (OpCit).Therefore, it could be argued that the process does not take enough holistic approach to the problems and very frequently depends on expert knowledge of individuals whose security/usability objectives may be different from the rest of the team and/or may not be properly aligned to corporate objectives.

The SQUARE approach on the other hand, is more suitable only in certain situations such as, an organisation's existing development process. Also the steps involved are very long, time consuming, cumbersome and can be very expensive process (Mead et al., 2008). Most importantly both of these methodologies do not say how the usability aspects of the system will be engineered.

This research has demonstrated that information systems have critical security and usability problems even today, albeit mostly due to human failures – the weakest link (Sasse et al., 2001; Whitten and Tygar, 1998; Gonzalez and Sawicka, 2002; Hinson, 2003; Flechais et al., 2003; Holzinger, 2005). These issues if not correctly checked, may leave users exposing vital

personal or account information. The Security and usability of IS can be analysed by comparing it with other systems. A comparative analysis of existing systems has been carried out; this helps the users to find out the functionality and other important aspects of the system in question (Saunders et al., 2000). At the same time it helps developers to find out the defects in the system in relation to the aspects and to make remedial actions. Some of the security models examined such as Biba Integrity Model and Take-grant Model may not be very effective in providing adequate system security because they are very dated (Stallings and Brown, 2012) and they focus strongly on security. The AEGIS (Flechais et al., 2007) and SQUARE (Mead et al., 2008) models have been used in an attempt to address some of the issues identified. But the fact remains that, one of the biggest problems with modern day IS security and usability is, to do with the dynamism of present operational conditions or organisational environments of operation, and which also involves rapidly changing technologies (Atoyan et al., 2006; Braz et al., 2007). We have identified, examined and proposed a novel approach which involves looking at; the Social, Economic, and Technical factors (SET) approach within an operational environment. The SET framework, if correctly implemented, it is envisaged, will effectively take care of and cover the aspects of security and usability in systems; because the SET framework can be adaptable to the changing technologies, environments and other security or usability problems as they occur.

Case Scenario: - Security can be viewed from different scenarios. When we talk about security for Fire safety, the solutions are sprinklers and fire alarms and when we talk about security for a PC system, the solutions are Passwords, Antivirus, and Firewalls etc. Therefore, security can be viewed from different cases and in a business information system environment; security is a wide area of operation. Different security threats and risks have to be considered to ensure the system safety. The threats can be viewed in more complex form as social, economic and technical threats. The security models (Biba integrity model, Take-grant model, RBAC model) which have been highlighted earlier, covers only a single or few aspects of system security (Stallings and Brown, 2008). Other approaches such as security auditing, are one of the best solutions to ensure the systems security on a continuous and long term basis (Kim and Solomon, 2012). Many organisations use different security models in different areas across their information system infrastructure. The T-grant model provides security of one vertex with respect to another vertex; it is limited to just one aspect of the security in the information systems. In the Biba integrity model, the major security issues include, providing access restrictions (read/write); which is also a single aspect in providing

the information security. RBAC model is used for providing security on the role or position of the employee in the business organisations (Krause & Tipton, 1997; Stallings & Brown, 2008; 2012). All the three models cannot be used as a single security system for the business information system. Most, if not all these drawbacks have hopefully been addressed in the proposed model or framework called SET, and can be used to cover the whole information system infrastructure.

In chapter 5, what we have identified are the canonical set of issues for security and usability in information systems design and development, categorised under the SET factors. The SET factors attempt to deal with the major areas of usability and security risks in the information systems. Some benefits of the SET framework are that, it is can be very stable, sustainable and powerful when compared to other security models, as it covers most importantly, the social threats or human aspects within the security, which is one of the major security risks areas for many business information systems.

Therefore, for effective usability and strong security, we recommended good combination or integration of some of these methods, and for the issues looked at or dealt with holistically. For example, SSM can be combined with SSADM, and in the process SSM can be used to deal with soft system issues and SSADM can be used to cover the hard system issues. Together, they may form a structured but flexible approach to information system development. The SSM can be used as a client front-end method to develop an information system or a workflow system. Workflow systems are used to document and control the business processes through combining the human and information resources of the organisation. Therefore, the development of a workflow system needs a method to deal with both the human (soft) and information (hard) issues (Fisher, 1999; Flechais et al., 2003; Gonzalez and Sawicka, 2002). In fact, we recommend, the best solution will be to develop a new framework or methodology for designing secure and usable systems, for example utilising the SET factors, which as described earlier, effectively addresses the canonical set of issues for security and usability in IS.

6.3 Conclusion

This thesis has highlighted and critically analysed the conflicting issues between security and usability in the design and development of information systems. It has analysed IS from an

academic perspective in an attempt to establish why it is difficult to achieve a balance, or trade-off or better compromise in dealing with the aspects in ISDM. Different kinds of usability and security techniques have been evaluated in this thesis. To achieve our aim for good strong security and effective usability, we carried out qualitative research and analysed the data to identify areas for improvements in ISDMs. The techniques can help developers succeed as from the design phase to the production level. Handling errors, performance and improving the efficiency and acceptability of the application are the main important features in usability and security. Generally, usable security can be achieved through practical hands-on user exercise, which involves selecting a group of participants and giving them tasks to do. These tasks performance approach is also known as the evaluation criteria methods of usable security. The approach tends to highlight the problems of use of the product at an early stage.

In the thesis, we have achieved a critical analysis of IS and evaluating ISDMs. We have seen that although some good frameworks/approaches exist, and some have been used in an attempt to tackle the issues of usability and security, none has achieved a proper balance between the two conflicting aspects of usability and security to adequately address the issues in ISDMs. One aspect always seems to weigh more than the other. However, in this thesis we have proposed a SET framework that takes into consideration the social dynamics, economic and technical aspects of the environment, to assist in the process of producing IS that is secure and usable and acceptable. The thesis has outlined some usability and security techniques which can be applied and tested on any systems. With the help of these approaches, the systems' features of concerned with regards to security and usability are outlined. One of the most important aspect of the techniques is, engaging the stakeholders frequently (as often as possible) to get on-the-spot or instant feedback during the development process. This is the most important factor in determining the quality of the product and its effectiveness and/or acceptance. It is an efficient technique that can be helpful in many ways. It is always important to implement the usability techniques to find out the security issues as well. A system which lacks the usability and security requirements is meant to be interaction-less and unsecured and therefore cannot exist for long, especially in an online environment. In an attempt to effectively solve the issues, the thesis has proposed a conceptual SET framework which, it is anticipated will be capable of addressing the problems of usability and security in IS design and development.

Overall, usability and security are said to be all about the user experience, performance, and the satisfaction levels of a service or application. Therefore, the processes for design and development of an information system must meet the satisfaction levels for both of these aspects (hard and soft) before being introduced to the market.

6.4 Limitations

This research focused on the analysing usability and security issues in design and development of information system. We analysed various ISDM but considered only a few of them (SSADM, SSM, and DSDM) due to resource limitations. The body of knowledge gained from the critical analysis would have been more in-depth if many of the other existing methodologies had been explored. Also, we encountered difficulties in accessing and evaluating some important state of the art data from companies, archives, agencies and/or government organisations, and which were not readily available to the public. Therefore, the accuracy and currency of certain aspects of the research have been compromised for lack of comparative up-to-date data, and which can impact on the overall result presented. New research is directed towards the use of information from several sources by aggregating them into a statistical model in an effort to increase the accuracy of the system. Also, it has been long from the point of research set up to completion, therefore a small number of dated literature referenced materials have been included, made from earlier years of research. And given the fact that the proposed SET framework is only conceptual and not fully developed or evaluated, it has been difficult to conclude with certainty how this will fit in different environments and stand the test of time with users. Therefore it will be necessary in future to have stakeholders to effective appreciate the solutions provided by the new system approach in real time. But importantly, the research questions raised in this thesis have been answered completely.

6.5 Future Work

This thesis has considered the current issues in information systems design to incorporate security and usability features, and has concluded that there is no existing ISDM that satisfactorily addresses both of these areas to provide a good balance. We have suggested that a combination of methodological approaches may offer some benefits, but the best approach will be the development of a novel framework / methodology based on the SET factors.

Accordingly, a conceptual usable-security framework based on the Social, Economic and Technical (SET) factors has been proposed but not evaluated. Therefore, for future work, the next step will be: to build on the research thesis and develop the SET framework / methodology, to define an experimental evaluative model and to test it in action.

6.6 Chapter Summary

There are too many answers to the usability and security questions raised by the design of information systems. But it is increasingly becoming very clear that no single approach or mechanism is likely to assure both effective usability and robust security. This is because not all the uses of the IS can be pre-empted or predetermined, and not all its risks can be prevented. Further research is needed in many areas, these include but not limited to the theories of human psychology and risks associated with usable security, and the technical capabilities of the technologies. However, it is expected that by analysing and pre-empting the potential problems much in advance as possible, then users will be able to get the best out of the IS technology. Here we have identified and proposed the SET factors as a framework to effectively deliver secure usability or usable security in the right or balanced proportion to overcome some of the problems faced in design and development of information systems.

REFERENCES

- Abran, A., Khelifi, A., Suryn, W. & Seffah, A. (2003). Usability Meanings and Interpretations in ISO Standards. Journal of Software Quality Control. Kluwer Academic Press. 11, Iss 4, pp325-338.
- Adebesin, T. F., Villiers, M. R. D. & Ssemugabi, S. (2009). Usability testing of e-learning: An approach incorporating co-discovery and think-aloud. *Proceedings of the 2009 Annual Conference of the Southern African Computer Lecturers' Association.* Eastern Cape, South Africa: AC
- Adams, A. and Sasse, M A. (1999) 'Users are not the Enemy'. In Communications of the ACM:42:12, Dec. 1999 pp 40-46
- Ahituv, N. And Neumann, S., (1990); Principles of Information Systems for Management. 3rd edn. WCB
- Akhgar, B., (2003) Development of a methodology for design and Implementation of Strategic Information Systems. PhD Thesis, Sheffield Hallam University, School of Computing & Management Sciences, 2003.
- Akhgar, B., (1997a) Component Based Development and Methodological classification, System Group/ Select Software Tools working paper, WP97/234.
- Akhgar, B., (1997b) Cost benefit analysis for methodology selection: An executive guide to customisation of methodologies. System Group/ Select Software Tools working paper, WP97/3014.
- Akhgar, B., (1998) SSADM Version 4. A method for Analysis and Design. VID Publication Company, 1998.
- Akhgar, B and Siddiqi, J., (2004) A conceptual template for construction of methodologies, CTCM. ACM conference in Computing, 2004.
- AL-Humaidan, F., Rossiter, B, (n.d). A Taxonomy and Evaluation for System Analysis Methodologies in a workflow Context: Structured System Analysis Design Method (SSADM), Unified Modelling Language (UML), Unified Process, Soft Systems Methodology (SSM) and Organisation Process Modelling (OPM). Computing Science, Newcastle University. Newcastle Upon Tyne [Online]. Available at; http://computing.unn.ac.uk/staff/cgnr1/tr751.htm/ [Accessed, 20th April 2011]
- Al-Nabhan, M. M. (2009) Adaptive, Reliable, and Accurate positioning model for Location Based Services. BURA, Nov, 2009. Brunel University School of Engineering and Design, PhD Thesis, pp1-234.

- Arreymbi, J and Draganova, C. (2008) User Requirements analysis for use of mobile phones in learning and teaching. In Claes, T. and Preston, D. S. (eds), Frontiers in Education. Rodopi, Amsterdam, 2008
- Arreymbi, J (2007), Investigating Issues in Mobile Network Security. Journal of Modern Applied Science. Vol. 2 (3) May, 2008. ISSN: 19131844
- Arreymbi, J and Dastbaz, M. (2002) *Issues in Delivering Multimedia Content to Mobile Devices*. Proceedings of the 6th International Conference on Information Visualisation, IEEE Computer Society, London, UK. July 2002.
- Aspinall, D., (2008) Security Models: Computing Security Lecture 7 [lecture notes]: School of Informatics, University of Edinburgh, January: [Online] Available at: http://docs.google.com/viewer?a=v&q=cache:UzNloMCcyPwJ:www.inf.ed.ac.uk/teaching/courses/cs/0910/lecs/softwaresec.pdf+Aspinall,+D.,+(2008)+Security+Models:+COMPUTING+SECURITY+LECTURE+7%5Blecture+notes%5D:+School [Accessed: 23/04/2011]
- Atoyan H, Duquet J, and Robert J. (2006) Trust in new decision aid systems. In Proceedings of the 18th international conference of the Association Francophone D'interaction Homme-Machine IHM'06, Montreal, April 18–21. New York: ACM Press; 2006. p. 115–22.
- Avison D. E. and Fitzgerald, G. (2006) Information Systems Development: Methodologies, Techniques and Tools. 4th edition, McGraw-Hill, Maidenhead. (2006).
- Avison D. E. and Fitzgerald, G (2003). Where now for Development Methodologies? Communications of the ACM, (January, 2003).
- Avison, D. E. & Fitzgerald, G. (2003). *Information Systems Development: Methodologies, Techniques and Tools.* (3rd ed), McGraw-Hill, London.
- Avison D. E and Fitzgerald G. (1995), *Information Systems Development Methodologies, Techniques and Tools Second Edition,* Alfred Waller Ltd Publishers, Henley on Thames
- Avison, D. E. and Wood-Harper, A. T., (1990) Multiview: An Exploration in Information Systems Development. London: McGraw-Hill
- Avison D. E. and Fitzgerald G. (1988), *Information Systems Development Methodologies, Techniques and Tools,* Alfred Waller Ltd Publishers. Henley-on-Thames.

- Azari, R. (2003), Current security management & ethical issues of information technology, published by Hershey: IRM Press.Baauw, E. & Markopoulous, P. (2004). A comparison of think-aloud and post-task interview for usability
- Battleson, B., Booth, A. & Weintrop, J. (May 2001). Usability Testing of an Academic Library Web Site: A Case Study. 27, 188-198
- BBC News Online (2010), "HSBC admits huge Swiss bank data theft". March 11th 2010. Available at: http://news.bbc.co.uk/1/hi/business/8562381.stm [Accessed: 13/03/2010]
- BBC News (2011), "Sony's online Play Station system was hacked and users personal details were stolen" April 20th 2011
- BBC News (2011), "Swiss Bank UBS losses $2.3 billion in Rogue trading". London, September 15, 2011
- BBC News (2009), "ICO indicated that 434 Organisations reported data security breaches". September 11th 2009.
- BBC News (2006), "Call centre Scam: India's IT industry urge Channel 4 to cooperate on data theft investigation". October 4th 2006.
- Bell, S. and Wood-Harper, T., (1998) Rapid Information Systems Development: Systems Analysis and Systems Design in an Imperfect World, 2nd ed. McGraw Hill, 1998
- Bennett, S., Mc-Robb. S and Farmer, R (2006) Object-oriented System Analysis and Design. McGraw-Hill Education
- Benson, S. and Standing, C. (2002), Information Systems: A Business Approach. Wiley & Sons, 2002 ISBN 0-470-80003-8.
- Bevan, N. (2005), Cost benefits framework and case studies. In R.G. Bias, & D.J. Mayhew (eds), Cost-Justifying Usability: An Update for the Internet Age. Morgan Kaufmann.
- Bevan, N. (1995), UX, Usability and ISO Standards. Professional Usability Services, London
- Beynon-Davies, P., (1998), 'Information Systems Development', Macmillan Press Ltd., Third edition.
- Biba, K. J. (1977), Integrity Considerations for Secure Computer Systems. MTR 3153. Mitre Corporation.
- Bignell, V. And Fortune, J. (1984), Understanding Systems Failures. Open

University, Manchester University Press. 1984

- Bishop. M., (1996) *UNIX Security: Threats and Solutions*. Presentation given at SHARE 86.0, Anaheim, CA. March 1996. Available at: http://seclab.cs.ucdavis.edu/~bishop/scriv/1996-share86.pdf. (Accessed 21 June 2010)
- Blandford, A., Keith, S., Connell, I. & Edwards, H. (2004). Analytical usability evaluation for digital libraries: a case study. *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries.* Tuscon, AZ, USA: ACM.
- Bocij, C. P., Greasley, A. and Hickie, S., (1999) *Business Information Systems - Technology, Development and Management for the E-Business.* Pearson Education Ltd, Harlow, Essex
- Bocij, P., Greasley, A. and Hickie, S., (2008) *Business Information Systems. Technology, Development & Management. -4th edn.* Pearson Education Ltd, Harlow, Essex 2008
- Boddy, D., Boonstra, A., and Kennedy, G., (2009) Managing Information Systems: Strategy and Organisation, (3rd edn), FT Prentice Hall, 2009.
- Booch, G., Rumbaugh, J and Jacobson, (1998) (1st Edn) The Unified Modelling Language User Guide: Addison Wesley: Reading Massachusetts
- Booch, G. (1989) *Object-Orientated Analysis and Design with Applications*
- Boren, M. T. & Ramey, J. (2000). Thinking aloud: Reconciling theory and practice. *IEEE transactions on professional communication,* 43, 261-278.
- Brace, I. (2008). *Questionnaire Design: How To Plan, Structure And Write Survey Material For Effective Market Research with CDROM. Kogan Page pub. 2008*
- Brandtz, P. B., & HEIM, J. (2007). User loyalty and online communities: why members of online communities are not faithful. *Proceedings of the 2nd international conference on INtelligent TEchnologies for interactive entertainment.* Cancun, Mexico: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- Braz C, Seffah A, and M'Raihi D. (2007) Designing a trade-off between usability and security: A metrics based-model. In: Proceedings of 11th IFIP TC 13 conference on human–computer interaction (INTERACT'07), Rio de Janeiro, September 10–14. Lecture Notes in Computer Science, vol. 4663. Berlin: Springer; 2007. p. 114–26.
- Break well, G. M., Hammond, S. & A. SMITH, J. (2006). *Research methods in psychology*, London, SAGE

- Brooks L and Ratsey N (1999) *Designing for the future: A framework for Web-Based Development,* BIT 9th Conference 1999, Manchester Metropolitan University
- Bryman, A. (1989) Research Methods and Organisation Studies. London, Unwin Hyman, 1989
- Buckland, M (1998), Overview of the History of Science Information Systems. Conference on History and Heritage of Science Information Systems. ASIS, Pittsburgh, Oct. 23-25, 1998.
- Bush, V. (1945) US A Government Library publication. MEMEX
- Buzan, T and Buzan, B (2006), The Mind Map Book (Mind Set) –education Publishers LLP, Harlow Essex, England, p279; ISBN: 978-1406-61020-8
- Cash, J. and Konsynski, B., (1985); IS redraws competitive boundaries. Harvard. Business Review. (Mar/April 1985) 134-142
- Ceri, S., Fraternali, P. & Bongio, A. (2000). Web Modelling Language (WebML): a modelling language for designing Web sites. *Proceedings of the 9th International World Wide Web Conference on Computer Networks. The Int. J. of Computer and Telecommunications Networking.* Amsterdam, The Netherlands: North-Holland Publishing Co.
- Chaffey D. and Wood, S., (2005) *Business Information Management*, 1st Edition. Essex, England.
- Checkland, P., and Scholes, J., (1990) 'S oft Systems Methodology in Action', John Wiley & Sons Ltd., New York, 1990.
- Checkland, P. and Holwell, S. (1998) Information Systems and Information Systems, making sense of the field. John Wiley & Sons, NY, 1998
- Checkland. P. and Scholes. J. (1997), *Soft Systems Methodology in Action.* John Wiley & Sons Ltd., New York, 1997.
- Checkland, P. (1981; 1998) Systems Thinking, Systems Practice, John Wiley & Sons, New York, NY. 1981; 1998. ISBN 0471986062
- Checkland, P. (1991), From Frameworks through Experience to Learning: The Essential Nature of Action Research, Information Systems Research: Contemporary Approaches and Emergent Traditions. H. Nissen, H.K. Klein and R. Hirschheim (Eds), North-Holland, New York, NY, pp. 397-403
- Chen J (2001), *Building Web Applications, Information Systems Management*, Winter, Vol.18, No.1

- Christiansen, CA. And Pintal, G. (2009). Effective Information Security: A win-win proposition for Enterprise and IT. [Online] Available at: .(Accessed: 25/4/2011)
- Churchill, E., Nelson, L., Smetters, D., (2008) Useful computer Security: IEE Computer Society.
- Clarke, R., (1990); Information Systems: The Scope of domain. Australian National University, Working paper version 5
- Cloyd, M. H. (2001). Designing User-Centered Web Applications in Web Time. *IEEE Softw.*, 18, 62-69.
- Collette, Ronald D. (2009), CISO soft skills: securing organisations impaired by employee politics, apathy, and intolerant perspectives, Published by Boca Raton: CRC Press.
- Collis, J. and Hussey, R (2003), Business Research: A practical guide for undergraduate and post graduate students (2nd Ed) Basingstoke: Palgrave Macmillan. ISBN: 10: 0333983254
- Commission of the European Communities: Information Technology Security Evaluation Criteria (ITSEC), Standard EIC 300 Version 1.2 (1991).
- Conallen, J. (2000). *Building Web applications with UML*, Addison-Wesley Longman Publishing Co., Inc.
- Craig A. Schiller ... [et al] (2007), Botnets : the killer web app, Published by Rockland Mass: Syngress
- Cranor, L.F. & Garfinkel, S. (Eds) (2005), Security & Usability: Designing Secure Systems that People can use. O'Reilly Media Inc. ISBN 10-0-596-00827-9
- Crinnion. John (1995), *Evolutionary Systems Development*, Pitman Publishing, London
- Crunchbase (2009), *System* [Online] Available at: http://www.crunchbase.com/company/system
- CSTB. (1990), *Scaling up: a research agenda for software engineering*, Communications of the ACM, Vol.33, No.3, 281-293.
- Daniels, C, V (1994) Information Technology: The Management Challenge: England: Addison-Wesley Publishing Co. Inc
- Davenport T. (1999), *Information Ecology: Mastering the Information and Knowledge Environment, UK.*

- DeLone W and McLean E (1992), *'Information Systems Success: The Quest for the Dependent Variable'*, Information Systems Research 3(I), 60-95.
- DeWitt, A.J.A.G (2007), Usability Issues with security of Electronic Mail. PhD Thesis. Brunel University, UK, 2007.
- DeWitt, A. J. & Kuljis, J., (2006), Aligning Usability and Security: A usability study of Polaris. 2006, SOUPS '06 Proceedings of the second Symposium on Usable privacy and Security pp. 1-7
- DeWitt, A. J. & Kuljis, J., (2006) Is usable security an Oxymoron? 2006, Interactions – A contradiction in terms?, vol.13, iss 3; pp.41-44
- D'Hertefelt S. (2000) Trust and the perception of security; 2000. Online, Available at http://www.interactionarchitect.com/research/report20000103shd.htm. (Accessed 20 June 2010)
- Dix, A., E., Finlay, J. D., Abowd, G. & Beale, R. (1998); Human-Computer Interaction, (2nd Edition), Prentice Hall.Dix, A., E. Finlay, J., D. Abowd, G. & Beale, R. (2004). Human-Computer Interaction, (3rd Edition), Prentice Hall.
- Drucker, P. F. (1993), Post Capitalist Society, Harper Business,
- Drury, J. (2000). Extending usability inspection techniques for collaborative systems. *CHI '00 extended abstracts on Human factors in computing systems.* The Hague, The Netherlands: ACM.
- DSDM Consortium (1995) [Online] Available at: http://www.dsdm.org/
- Dumas, J. S. (2003). User-based evaluations. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications.* L. Erlbaum Associates Inc.
- Edwards, H. & Benedyk, R. (2007). A comparison of usability evaluation methods for child participants in a school setting. *Proceedings of the 6th international conference on Interaction design and children.* Aalborg, Denmark: ACM.
- Edwards, C., Ward, J. and Bytheway, A., (1991); The essence of Information systems. Prentice Hall, 1991
- Ein-Dor and Segve, J., (1993), Information systems Research. Fragmented adhocar, 3rd ISS Conference 1993
- Ellison, B. N. & Boyd, M. D. (2007). Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication,* 13, Article 11.
- Ericsson, K. & Herbert, S. (1984). *Protocol Analysis: Verbal Reports as Data,*

Cambridge, MIT Press.

- Federal Aviation Administration (FAA): Standard Terminal Automation Replacement System, Human Factors Team Report of the Computer–Human Interface Re-Evaluation (1998).
- Federation of American Scientists (FAS), Available online at: http://www.fas.org/sgp/library/nispom/change_ch8.htm. (Accessed 25/03/2011)
- Ferraiolo, D.F., and Kuhn, D.R. (1992) "Role Based Access Control"15th National Computer security Conference. Oct 13-16, 1992. Pp 554-563 available online at http://csrc.nist.gov/rbac/ferraiolo-kuhn.pdf (Accessed 15/04/2010).
- Fidas, C., Voyiatzis, A., and Nikolaos, M., (2010) When Security meet Usability: A User Centric Approach on A crossroads Priority Problem. 14th Panhellenic Conference on Informatics: Patras, Greece. 2010. [Online] Available from; http://www.artemiosv.info/papers/PCI2010.pdf [Accessed: 29/04/2011]
- Fisher J (1999), *'Improving the Usability of Information Systems: The Role of the Technical Communicator'*, European Journal of Information Systems 8, 294-303.
- Fisk, A.D, Rogers, WA, Charness, N, Czaia, SJ and Sharit, J (2009). Designing for Adults: Principles and Creative Human Factors approaches. CRC press, 2009
- Flechais, I., Sasse M.A. and Hailes, S.M.V. (2003), Bringing Security Home: a process for developing secure and usable systems. In NSPW'03: Proceedings of the 2003 workshop on new security paradigms. 49-57. NY: ACM
- Flechais, I., Sasse, M.A. (2009). Stakeholder involvement, motivation, responsibility, communication: How to design usable security in e-Science. *International Journal of Human-Computer Studies*, 67(4), 281-296.
- Flechais, I., Mascolo, C., Sasse, M. A., (2007) 'Integrating Security and Usability into the Requirements and Design process'. Int. J. Electronic Security and Digital Forensics vol. 1, No.1, pp.12-26, 2007. Inderscience Enterprises Ltd.
- Flynn. D. (1992), *Information Systems Requirements - Determination and Analysis*, McGraw-Hill Publishing Company, Berkshire, UK
- **Folmer**, E. & Bosch, J., (2004) Architecting for usability; a survey. Journal of Systems and Software Volume 70, Issue 1. Pages 61-78. January 2004
- Forrester Research (2010) Forrester Forecasts [Online] Available at: http://techcrunch.com/2010/03/08/forrester-forecast-online-retail-sales-will-grow-to-250-billion-by-2014/ [Accessed, 20/10/2010]

- Fox, D. & Naidu, S. (2009). Usability Evaluation of Three Social Networking Sites. 11, 11.
- Freetutes.Com (2007-2011), Systems Analysis and Design: Object-oriented Methodology Life Cycle [Online] Available from; http://www.freetutes.com/systemanalysis/sa2-object-oriented-methodology.html/ [Accessed, 21th April 2011]
- Gabrielli, S., Mirabella, V., Kimani, S. & Catarci, T. (2005). Supporting cognitive walkthrough with video data: a mobile learning evaluation study. *Proceedings of the 7th international conference on Human computer interaction with mobile devices & services.* Salzburg, Austria: ACM.
- Gaonjur, P. (2008). *Scribd.* [Online], Available at: http://www.scribd.com/doc/8191121/Usability-of-Social-Networking-Sites?autodown=pdf [Accessed 25/10/2010].
- Garlik Report (2009). UK cyber-Crime report 2009 (Fafinski, S., and Minassian, N., (ed.)). Invenio Research, September 2009. [Online] Available at: http:// http://www.garlik.com/file/cybercrime_report_attachement [Accessed: 29/05/2011]
- Gibson. Brian (1992), *Managing Computer Projects*, Prentice Hall International (UK) Ltd, Hemel Hempstead
- Gonzalez, J.J., (2002) Modelling Erosion of Security and Safety Awareness. Proceedings of the Twentieth International Conference of the System Dynamics Society July 28 - August 1, 2002 Palermo, Italy.
- Gonzalez, J.J. and Sawicka A., (2003) Origins of compliance – An instrumental conditioning perspective. Submitted to Fifth International Conference on Cognitive Modelling (ICCM 2003). Bamberg, Germany.
- Gonzalez, J.J and Sawicka, A. (2002) A Framework for Human Factors in Information Security. WSEAS Int. Conference on Information Security. Rio de Janeiro 2002
- Goodland. M (1995), *SSADM: A Practical Approach*, McGraw-Hill Publishing Company, Berkshire, UK
- Greenfield, A. (2007). The trouble with Computers. The Economist, Technology Quarterly, Q3, 2007 [Online] Available at: http://www.economist.com/node/9719037?story_id=9719037 [Accessed, 29/10/2010]

- Gregor, S. and Jones, D. (1999) *Web Information Systems Development: Some Neglected Aspects*, Faculty of Informatics and Communication, Central Queensland University, Australia
- Grossman, T., Fitzmaurice, G. & Attar, R. (2009). A survey of software learnability: metrics, methodologies and guidelines. *Proceedings of the 27th international conference on Human factors in computing systems.* Boston, MA, USA: ACM.
- Guan, Z., Lee, S., Cuddihy, E. & Ramey, J. (2006). The validity of the stimulated retrospective think-aloud method as measured by eye tracking. *Proceedings of the SIGCHI conference on Human Factors in computing systems.* Montreal & Quebec 233; Canada: ACM.
- Haag, S., Cummings, M. and McCubbrey, D. J., (2004) Management Information Systems for the Information Age (4^{th} edn) McGraw-Hill, 2004
- Hafner, K. And Markoff, J., (1995) Cyberpunk: Outlaws and hackers on the computer frontier (Revised). Simon & Schuster pub. 1995.
- Hafner, K. And Markoff, J., (1991) Cyberpunk: Outlaws and hackers on the computer frontier. Simon & Schuster pub. 1991.
- Harley D. (2007), Anti-Virus Information Exchange Network malware defense guide for the Enterprise, Published by Burlington, MA: Syngress.
- Hawryszkiewycz, I.T, (1998), 'Introduction to Systems Analysis and Design', Prentice Hall Australia Pty Ltd., fourth edition.
- Heather, D. & J. C, T. (1993). Enhancing the Performance of Interface Evaluators Using Non-Empirical, Annual Meeting, 1993 Santa Monica, CA. 1132-1136.
- Hinson, G., (2003) Human Factors in Information security. White paper, IsecT Ltd, [Online] Available at: http://www.noticebored.com [Accessed 11/10/2011]
- Hivari H. and Klein (1987), Beyond Methodologies- *Keeping up with Information Systems Development Approach through Dynamic Classification*
- Hofmann (1988), *Formalised Systems Development Methodologies, A Critical Perspective*
- Holzinger, A. (2005). Usability engineering methods for software developers. J Communication. ACM. 48, 71-74.
- Hom, J. (2009). *Usability Evaluation* [Online]. Available: http://www.usabilityhome.com/ [Accessed 29/09/2009].

- Hopkins B (1997), '*Brave New World? Towards a Programme to Free us from the Methodological Straitjacket in Information Systems Development'*, BIT 7th Conference, Manchester Metropolitan University
- Hornb, K., 230, Fr, *E., 248 & Kj (2008). Making use of business goals in usability evaluation: an experiment with novice evaluators. Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems.* Florence, Italy: ACM.
- Howard, J. D. (1997) *An Analysis of Security Incidents on the Internet 1989-1995*. Carnegie Mellon, University Ph.D. Thesis, 1997.
- Huang, S. C., Bias, R. G., Payne, T. L. & Rogers, J. B. (2009). Remote usability testing: a practice. *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries.* Austin, TX, USA: ACM.
- Hung, V. O., (2007) Software Construction [Online].Available from http://cnx.org/content/m14739/latest/ [Accessed, 15/04 2011]
- Hughes, Ireland et al. (2004) Project Management for IT-Related Projects: Swindon. The British Computer Society (BCS)
- IEEE Symposium on research in security and privacy (1989): The Chinese Wall Security policy, 206-14: IEEE: Oakland, California [Online]. Available at; http://www.gammassl.co.uk/topics/chinesewall.html [Accessed: 25/04/2011]
- IEEE Standard 610-12-1990. Standard on Software Engineering. 1990
- Infinite Computing Systems (2011), Computing Systems: Innovation Solutions on Demand. USA. Infinite Computing Systems Inc [Online] Available from http://infinite-usa.com/Home/tabid/166/language/en-US/Default.aspx/ [Accessed, 20th April 2011]
- Ingram, P. (1995). The World Wide Web. *Comput. Geosci.*, 21, 799-816.
- Institute of Electrical and Electronics Engineers (IEEE): 1061-1998 IEEE Standard for a Software Quality Metrics Methodology (1998).
- IEEE Symposium on research insecurity and privacy (1989): The Chinese wall security policy, 206-14: IEEE: Oakland, California [Online]. Available from http://www.gammassl.co.uk/topics/chinesewall.html[Accessed: 25/04/2011]
- International Organisation for Standardization: ISO/IEC 9126-1:2001 Edition 1; Software product Evaluation – Quality Characteristics and Guidelines for the User, Geneva (2001).

- International Organisation for Standardization, 1999. ISO 13407: Processes for Interactive Systems, Geneva, Author.
- International Organisation for Standardization (1998) ISO 9241-11: "Ergonomic requirements for office work with visual display terminals (VDTs - Part 11: Guidance on Usability".
- ISO 9241-210 (2007) Human-centred design process for interactive systems (formerly known as 13407). Working Draft. ISO.
- ISO/TR 16982 (2002) Usability methods supporting human centred design. ISO.
- ISO/IEC CD 25010 (2007) Software product Quality Requirements and Evaluation (SQuaRE) – Quality model. ISO.
- International Standards Organization. (2008a). ISO 9241-20: Ergonomics of human-system interaction – Part 20: Accessibility guidelines for information/communication technology (ICT) equipment and services. Geneva: International Standards Organization.
- International Standards Organization. (2008b). ISO 9241-171: Ergonomics of human-system interaction. Part 171: Guidance on software accessibility. Geneva: International Standards Organization.
- International Standards Organization. (2008c). ISO DIS 9241-210: Ergonomics of human system interaction. Part 210: Human-centred design process for interactive systems (formerly known as 13407). Geneva: International Standards Organization.
- International Standards Organization. (2008d). ISO/IEC 10779: Office equipment accessibility guidelines for elderly persons and persons with disabilities. Geneva: International Standards Organization.
- International Standards Organization. (2006). ISO 20282-2: Ease of operation of everyday products - Part 2: Test method for walk-up-and-use products. Geneva: International Standards Organization.
- Ivan Tirado (2008), Business oriented information security requirements development. Published by ACM, Available at: http://portal.acm.org/citation.cfm?id=1456625.1456642&coll=ACM&dl=ACM&CFID=89775628&CFTOKEN=69361880
- Ivory, M. Y. & Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.*, 33, 470-516.

- Jacobson. I. (1995),*Object-Orientated Software Development* Hill Publishing Company, Maidenhead
- Jayaratna. N. (1989), *Understanding and Evaluating Methodologies: A Systematic Framework*
- Jayaratna. N and Fitzgerald. B (1996), *Lessons Learned from the use of Methodologies* British Computer Society Fourth Conference 14th September 1996, UCC, Cork Ireland
- Jayaratna, N. (1994) Understanding and Evaluating Methodologies, NIMSAD, A Systemic Framework, McGraw-Hill, 1994.
- Jayaratna, N. (1996) Understanding and Evaluating Methodologies, NIMSAD, A Systemic Framework, McGraw-Hill, 1996.
- Jessup, L. M.; Valacich, J. S. (2008). *Information Systems Today* (3rd ed.). Pearson Publishing. Glossary p. 416
- Jessup, L and Valacich, J., (2003) Information Systems Today. Pearson Education. ISBN: 0130094145
- Johnston J, Eloff J, and Labuschagne L. (2003) Security and human computer interfaces. IEEE Computers & Security 2003;22 (8).
- Jordan, P.W (2002), Human factors for pleasure in product use: [Online] Taylor and Francis. Available at. (Accessed 20/4/2011)
- Jøsang, A. & Patton, M. (2001), UI Requirements for Authentication of Communication, White Paper, Distributed Systems Technology Centre, QUT, Brisbane, Australia (2001).
- Karger, P., Austel, R., and Toll, D., (2000). IBM Research Report: A New Mandatory security policy combining security and integrity: IBM Research Division: York Town Heights: NY10598 [Online]. Available at: http://domino.research.ibm.com/library/cyberdig.nsf/a3807c5b4823c53f85256561006324be/3a60ddc27f4729768525694600665d6/$FILE/RC21717.pdf [Accessed: 23/04/2011]
- Kaner, C., Falk. J., Nguyen, H., (1999) (2nd edn) Testing computer software. Canada. Wiley
- Kantrowitz, J. (2011) Education research report: the rise of K-12 blended learning. Available online at: www.educationresearchreport.blogspot.com/2011/02/rise-of-k-12-blendedlearning (Accessed 21/3/2011)

- Karat, Clare-Marie (2009), Iterative Usability testing of security Applications: Human Factors and Ergonomics Society. Annual Meeting Proceedings pp 273-277 (5) ISSN: 10 71-1813
- Keen, P. (1996) Evolution of the IT Management: [Online] Available at: http://www.perterkeen.com [Accessed: 15/09/10]
- Kelly S, Holland C and Light B (1999), *'A Departure from Traditional Systems Development Methodologies: Enterprise Resource Planning (ERP) Systems and the Use of Process Modelling Tools'*, 9th Annual BIT Conference, Manchester Metropolitan University
- Kelly, S.; Gibson, N.; Holland, C.; and Light, B., (1999). "Focus Issue on Legacy Information Systems and Business Process Engineering: a Business Perspective of Legacy Information Systems". *Communications of the AIS* **2** (7): 1–27. July 1999
- Kim, D., and Solomon, M. G., (2012) Fundamentals of Information Systems Security, Jones & Bartlett Learning, Sudbury, MA. USA 2012. ISBN:978-0-76379025-7
- Kirakowski, J. (1998), Human Computer Interaction: from Voltage to Knowledge. Chat-well Bratt, Pub: England, 1998. ISBN: 0-86238-179-7
- Krause, M., and Tipton, H., (1997) Handbook of Information Security management: Confidential Models: Integrity Model: CRC Press LLC [Online]. Available from; http://www.ccert.edu.cn/education/cissp/hism/023-026.html/ [Accessed: 24/04/2011]
- Kuniavsky, M. (2003), Observing the User Experience (A practitioner's guide to user research). Morgan Kaufmann Pub: -Elsvier, USA. 2003. ISBN: 1-55860-923-7
- Landauer, T. K. (1995). The trouble with Computers: usefulness, usability and productivity. MIT Press Cambridge MA. ISBN: 10:0262-62108-8
- Landreth, B., (1989), Out of the Inner Circle: the true story of a Computer intruder capable of cracking the Nation's most secure computer systems. Tempus books, 1989. Microsoft corporations, USA
- Langefors, B. (1966), Theoretical Analysis of Information Systems. Lund: Student literature. 1996
- Laudon, K.C. and Laudon, J. P., (2010) Management Information Systems: Managing the Digital firm (11th edn) Pearson
- Lecture Notes on Information System security (n.d.); Available online at: http://www.fas.org/sgp/library/nispom/change_ch8.htm [Accessed 25/3/11]
- Lester, S. (2008) Soft Systems Methodology [Online] Available at:

http://www.humanecology.com.au/SSMeth.pdf [Accessed: 20/05/2011]

- Locasto, M. E., Cretu, G. F., Stavrou, A., and Keromytis, A. D., (2007) A Model for Automatically Repairing Execution Integrity. Technical report, 2007.
- Mack, R., and Nielsen, J. (1993). Usability inspection methods. *ACM SIGCHI Bulletin* **25**, 1, pp.28-33.
- McEwan, T., (2010) Digitally Enabled. The BCS IT Now Magazine, November 2010, BCS UK.
- Mclean, J., Schell, R. R., & Brinkley, D. L. (2002). Security Models: Encyclopaedia of software Engineering: Wiley Online Library: Published Online: [Online]. Available from; http://onlinelibrary.wiley.com/doi/10.1002/0471028959.sof297/full/ [Accessed: 24/04/2011]
- Mclean, E. and Swanson, F. (1980) Management Information Systems, An Academic perspective, UCLA Working papers no 5-80
- Mead, N. R., Viswanathan, V., Padmanabhan, D., and Raveendran, A., (2008), Incorporating Security Quality Requirements Engineering (SQUARE) into Standard Life-Cycle Models. SEI, TECHNICAL NOTE CMU/SEI-2008-TN-006 Carnegie Mellon University, USA 2008
- Mikko T. Siponen, Hari Oinas-Kukkonen (2007), A review of information security issues and respective research contribution, Available from ACM Digital library:
- Minati, G. and Pessa, E., (2006), Collective Beings – Contemporary Systems Thinking. 2006
- Monk, A., Wright, P, Haber, J. and Devonport, L. (1993) Improving your Human-Computer Interface: A Practical Technique. Prentice Hall, London, 1993, ISBN: 0-13-010034X
- Moynihan, E., (1993) Information System Series: Business Management and System Analysis: Henley-On-Thames: Alfred Waller Ltd.
- Muñoz-Arteaga, J., González, R. M., Martin, M.V., Vanderdonckt, J., and Álvarez-Rodríguez, F., (2009) A methodology for designing information security feedback based on User Interface Patterns. Advances in Engineering Software, Elsevier, 40 (2009) 1231–1241
- NCC (2004) National Computing Centre, UK. Information System Risk Management, NCC Publications 2004
- Nguyen, H., Johnson, B., and Hackett, M. (2003) Testing Applications on the Web.

Test planning for mobile and Internet- based System; Indiana. Wiley pub

- Nielsen, J. (2010) iPad Usability: First Findings From User Testing. Jakob Nielsen's Alertbox, July 2, 2010. http://www.useit.com/alertbox/ipad.html [Accessed, 25/10/11]
- Nielsen, J. (2009) Usability Engineering. Kaufmann, 2009, ISBN: 0125184069
- Nielsen, J. (1993) Usability Engineering. AP Professional, 1993. ISBN: 0125184069Nielsen, J. (1992) Finding Usability Problems through Heuristic Evaluation. In the Proceedings of ACM Computer Human Interaction 1992 (CHI'92) Monterey, CA, (US) 3–7 May 1992
- NIST publications; [Online] Available at: http://csrc.nist.gov/rbac/ferraiolohuhn-1992 [Accessed, 04/10/2011]
- NIST Computer Security Resource Centre (CSRC), available online at: http://csrc.nist.gov/ [Accessed 10/04/2011]
- NIST Special publication, 800-18, Rev 1 [Online] Available at: http://csrc.nist.gov/publications/nistpubs/800-18-Rev1/sp800-18-Rev1-final.pdf [Accessed; 20/04/2011]
- Norman KL, Panizzi E, (2006). "Levels of automation and user participation in usability testing." Interact Comput 18, 246-264.
- Norton. K (1999), *'Applying Cross Functional Evolutionary Methodologies to Web Development'*, ICSE Workshop on Web Engineering.
- NRC (2009) Towards better usability, security and privacy of information technology: Report of a workshop: National Research Council: The National Academic press, Washington, DC.
- Olle. W T. (1991*), Information Systems Methodologies: A framework for understanding*, Addison-Wesley Publishing Company, Wokingham.
- OGCIO.gov (2008) (version2.6) Office of the Government Chief Information Officer: An introduction to Structured Systems Analysis and Design Methodology: The Government of Hong Kong Special Administrative Region: [Online] Available from; http://ogcio.gov.hk/eng/prodev/download/s3a_pub.pdf/ [Accessed: 29/04/2011]
- Olsina. L and Godoy. D, Lafuente. G and Rossi. G (1999), *'Specifying Quality Characteristics and Attributes for Web Sites'*, ICSE 99, Web Engineering Workshop, USA

- OMG (2011) Introduction to OMG's Unified Modelling Language: TM (UML (R): Object Management Group Inc, Needham, MA. USA) [Online]. Available from; http://www.omg.org/gettingstartred/what_is_uml.htm [Accessed: 28/04/2011]
- Open University (2011), Models and Modelling: Use cases and activity diagrams: The Open University, Milton Keynes. UK [Online]. Available from; http://openlearn.open.ac.uk/mod/oucontent/view.php?id=397581&section=6.1 [Accessed: 28/04/2011]
- Orlikowski, W. and Baroudi, J. (1991; Studying Information Technology in Organisations: Research Approaches and Assumptions, Information Systems Research Vol. 2. No.1. pp 1-28
- OTS solutions (2008-2010) Outsourced Software Development. [online] Available at: www.otssolutions.com [Accessed: 25/10/11]
- Parallels (1999-2009): YSI Security Standard: Ypsilon: Luxembourg [Online] Available from; http://ypsilon-it.com/gpage2.html [Accessed: 25/04/2011]
- Patching, D., (1990) Practical Soft Systems Analysis. Pitman. ISBN:0273032372:
- Patrick, AS., Long, AC., and Flinn, S (2003). HCI and Security Systems. National Research Council Canada. Institute for Information Technology, 2003
- Paul R.J (1993), *Why Users Cannot 'Get What They Want'*, ACM SIGIOS Bulletin, December 1993, Vol.14, No.2, Pg. 8-12
- PCI data security standards, Available at: https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml [Accessed: 23/03/2010]
- Pearson M and Paynter J (1999), *An Analysis of WWW-based Information Systems*, Department of Science and Information Systems Development, BIT 9th Conference 1999, Manchester Metropolitan University
- Petrie, H., Hamilton, F., King, N. and Pavan, P. (2006). Remote usability evaluations with disabled people. In the Proceedings of CHI 06: ACM Annual Conference on Human Factors in Computing Systems, 1133-1141. New York: ACM Press.Piedad, F and Hawkins M. (2001) High Availability Design, Techniques and Processes. Prentice Hall, Inc. NJ. ISBN 0-13-096288-0.
- Polack, J. (2009). "Planning a CIS Education Within a CS Framework". *Journal of Computing Sciences in Colleges* **25** (2): 100–106. ISSN 1937-4771
- Pooley R. and Stevens P. (1999), *Using UML*, New York, Addison - Wesley.

- Pressman, R (2010) (7th edition) Software Engineering: A Practitioner's Approach. New York: NY10020, McGraw-Hill
- Pressman. R. S. (2001), 'Software Engineering: A Practitioners Approach', McGraw-Hill Publishing Company, London
- Punch, F. K. (2005). Introduction to social research: quantitative and qualitative approaches, London, Sage Publications Ltd.
- Red Hat(R), inc., (n.d.) [Online]. Available from; http://www.centos.org/docs/5/html/Deployment_Guide-en-US/sec-mls-ov.html/ [Accessed: 25/04/2011]
- Roger, Y.; Sharp, H., and Preece, J., (2011) Interaction Design: beyond human-computer interaction. (3rd Edn) John Wiley, 2011. ISBN 978047066576-3
- Rubin, J. (1994). Handbook of usability testing: how to plan, design, and conduct effective tests. Wiley, 1994.
- SansGUI (2000-2003) SansGUI modelling and Simulation environment (version1.2) [Online] Available from; http://protodesign-inc.com/doc/SGfeat/object-oriented_development.htm/ [Accessed, 22th April 2011]
- Saunders, M., Lewis, P., and Thornhill, A., (2000) Research Methods for Business Students 2nd ed. FT Prentice Hall. 2000, ISNE: 0-273-63977-3
- Sasse, M. A. (2011), Designing for Homer Simpson – D'oh!. Usable Security. Interfaces, The Quarterly magazine of the BCS Interaction group. 86 Spring 2011. ISSN 1351-119X
- Sasse, M.A. (2010). Not seeing the crime for the cameras? Communications of the ACM, 53 (2), pp.22-25.
- Sasse, M A, Brostoff, S. and Weirich, D (2001). Transforming the Weakest link: a human-computer interaction approach to usable and effective security. BT Technology Journal, 2001, 19, 122-131
- Schimidt, T. (2006), Literature Review of Soft Systems Methodology. nimrod@mip.sdu.dk? [Online] Available at, http://docs.google.com/viewer?a=v&q=cache:x5m9ZRo8HY0J:thesis.msc-cse.com/pdf/article_ssm.pdf [Accessed :20th April 2011]
- Schneier, B., (2003) Beyond Fear thinking Sensibly about Security in an Uncertain World, Copernicus Books.

- Schneier, B., (2000) Secrets and Lies: Digital Security in a Networked World. New York: John Wiley & Sons, Inc., 2000.
- Schumacher, M. (2001), The use of SSADM (Structured System Analysis and Design methodology) as a standard methodology on Information systems projects. GMbh. GRIN Publishing [Online]. Available from; http://www.grin.com/e-book/106034/the-use-of-ssadm-structured-systems-analysis-and-design-methodology-as/ [Accessed 19/04/2011]
- Schultz, E., Proctor, R., Lien M and Salvendy, G., (2001): Usability and security: An Appraisal of Usability Issues in Information Security Methods: Computing and security vol.20, no7, pp620- 634, 200. Great Britain: Elsevier Science Ltd.
- Seffah A., Donyaee M., Kline R., Padda H.K. (2006): Usability Metrics: A Roadmap for a Consolidated Model. Journal of Software Quality, Volume 14, Number 2 (2006).
- Shallit, J (1995) A very brief History of Computer Science Lecture Notes, University of Waterloo, Ontario, Canada. [Online] Available at. (Accessed, 01/5/2011)
- Shneiderman, B. and Plaisant, C., (2010), Designing the User Interface: Strategies for Effective Human-computer interaction, 5th Edn. Pearson.
- SmartDraw (2011), What are SSADM Diagrams? [Online] Available from http://www.smartdraw.com/resources/tutorials/ssadm-diagrams/ [Accessed :19th April 2011]
- SOFFTERRA (1999-2011), Custom web and Software Solution: Object Oriented Analysis and Design. [Online] Available from http://www.softerra.com/skillset_ooad.htm/ [Accessed, 22th April 2011]
- Sommerville, I (2001; 2011) Software Engineering. Boston, Massachusetts. Pearson Education Ltd.
- Sophos Security Threat report (2009), Available at: http://www.sophos.com/sophos/docs/eng/marketing_material/sophos-security-threat-report-jan-2009-na.pdf [Accessed: 13/03/2010]
- SparxSystems (2011) (version8): Enterprise Architect: The Use Case Model: Sparx Systems pty Ltd. Victoria, Australia.
- Stallings, W and Brown, L. (2012) Computer Security: Principles and Practice (2nd Edn) Pearson, London,
- Stallings, W and Brown, L. (2008) Computer Security: Principles and Practice Pearson, London,

- Stallings, W. (2000) Network Security Essentials: Applications and Standards. Prentice Hall, London, Chapter 9. 2000Stapleton. J. (1999) *DSDM- Dynamic Systems Development Methodology- The Method in Practice*
- Stowell, F. (1995), *Information Systems Provision, The Contribution of SSM,* McGraw-Hill Book Company, London
- Times Online (March 11th 2010), HSBC admits huge scale of Swiss data theft, Available online at: http://business.timesonline.co.uk/tol/business/industry_sectors/banking_and_finance/article7057981.ece [Accessed: 13/03/2010]
- Thai Indian News (May 8th 2008), HSBC admits huge data loss, Available at:http://www.thaindian.com/newsportal/world-news/hsbc-admits-huge-data-loss-in-hong-kong_10046281.html (Accessed, 25/03/2010)
- Thomsett, R. (1998), *Third Wave Project Management- A Handbook for managing the complex Information System*
- Thuraisingham, B. M. (2005), Database and applications security : integrating information security and data management, Published by Boca Raton, Fla, London.
- Trepper C (2000), *'Methodologies that Vary According to the Project'*, Information Week.com, 21st August, 2000.
- Trung, H. VO., (2007) Software Construction [Online] Available at: http://cnx.org/content/m14739/1.2/ 2007. [Accessed, 24/11/11]
- UK Online for Business, (UK government Online support for Business). Available at: webarchive.nationalarchives.gov.**uk**/+/www.direct...**uk**/.../dg_100177. [Accessed, December 5th 2010]
- Users in on the Design. *Information Technology and Libraries,* 19, 141-151.
- UTC (2010) [Online] CS Venn Diagram. Available at: https://commons.wikimedia.org/wiki/File:CS_Venn_Diagram.pdf [Accessed, 15/5/2011]
- Veenendaal, E. V., (2002), 3rd edn. The Testing Practitioner: The Netherlands, UTN publisher
- Wang, J., (2001) Information System Analysis: [MS1S488]: School of Business Administration: University of Missouri. St Louis [Online] Available from; http://www.umsi.edu/~sauterv/analysis/488_f01_papers/wang.htm [Accessed: 28/04/2011]

- Ward P (1991), 'The Evolution of Structured Analysis' Part 1, Pg 4-16, Michigan, U.S.A
- Web Security and Usability. [online] Available at: Accessed: 29/4/2011)
- Website Outsourced Software Development (2008-2010) Agile Methodology – Changing ways of software development [Online] Available from http://www.otssolutions.com/blog/?p=99 [Accessed, 17th April 2011]
- Wharton C., Rieman, J., Lewis C. and Polson, P. (1994) The Cognitive Walkthrough Method: A Practioner's Guide. In *Usability Inspection Methods*, John Wiley & Sons, Inc., 1994.
- White D and Guailemos M (1999), *'How Organisations Ensure Successful Information Systems Development'*, BIT 9th Conference 99, Manchester Metropolitan University
- Whitman, M.E. and Mattord H.J., (2009) Principles of information security (3th ed. Course Technology, Cengage Learning, China, 2009
- Whitman, M.E. and Mattord H.J., (2012) Principles of information security (4th ed. Course Technology, Cengage Learning, China, 2012. ISBN: 13: 978-1-111-13823-3
- Whitten, A and Tygar J.D., (1998) Usability of Security: A Case Study. CMU-CS-98-155, December 18, 1998
- Wilbert O. Galitz (2007) The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques. Wiley Publishing Inc
- Wilson, B. (1999) *Systems, Concepts, Methodologies and Applications,* Wiley and Sons Publications, Chichester and New York.
- Winschiers, H. & Fendler, J. (2007): Assumptions Considered Harmful. HCI (10) 2007: pp.452-461
- Workman, M., Phelps, D.C., and Gathegi, J.N., (2013) Information Security for Managers; Jones & Bartlett Learning, 2013, Sudbury, MA, USA. ISBN: 978-0-7637-9301-2
- Yan, J., Blackwell, A., Anderson, R & Grant, A., (2000) The memorability and security of passwords . some empirical results. Technical report (University of Cambridge. Computer Laboratory); no. 500. ISSN 1476-2986
- Zabriskie, N. And Huellmantel, B. (1994); Marketing research as a strategic tool. Journal of Long Range Planning. Vol. 27, No.1. pp. 107-118

- Zurko, M. E. &. Simon, R. T. (1997). User Centric security . *The pen Group Research institute cambridge* .

Bibliography

- Avison DE and Fitzgerald G. (1995), *Information Systems Development Methodologies, Techniques and Tools Second Edition,* Alfred Waller Ltd Publishers, Henley on Thames
- BCS. (2011): Usable Security: Decades of confusion and no closer to solving the conundrum. The Quarterly magazine of BCS interaction group: Interfaces: 86 Spring 2011: [Online]. Available from http://www.bcs.org/upload/pdf/interfaces86.pdf [Accessed: 06 May 2011]
- Kainda, R., Flechais, I. & Roscoe, A., (n.d): Security and Usability Analysis and Evaluation [Online]. Available from http://www.comlab.ox.ac.uk/files/2859/ares_main.pdf [Accessed: 06 May 2011]
- Booch. Grady (1989) *Object-Orientated Analysis and Design with Applications*
- Bocij. Chaffey, Greasley and Hickie (1999) *Business Information Systems - Technology, Development and Management for the E-Business,* Pearson Education Ltd, Harlow, Essex
- Bell. S. and Harper, T. (1998). *Rapid Information Systems Development, McGraw-Hill Publish*ing, UK
- Crinnion. J. (1995), *Evolutionary Systems Development*, Pitman Publishing, London
- Davenport. T. (1999), *Information Ecology: Mastering the Information and Knowledge Environment*
- Flynn. D. (1992), *Information Systems Requirements - Determination and Analysis,* McGraw-Hill Publishing Company, Berkshire, UK
- Gibson. B. (1992), *Managing Computer Projects,* Prentice Hall International (UK) Ltd, Hemel Hempstead
- Goodland. M (1995), *SSADM: A Practical Approach,* McGraw-Hill Publishing Company, Berkshire, UK
- Hivari H. and Klein (1987), Beyond Methodologies- *Keeping up with Information Systems Development Approach through Dynamic Classification*

- Jacobson. I. (1995),*Object-Orientated Software Development* Hill Publishing Company, Maidenhead
- Jayaratna. N. (1989), *Understanding and Evaluating Methodologies: A Systematic Framework*
- Nielsen J. (1982) *Usability Methodologies. Available on at: www.usabilitynet.org*
- Preece, J. (ed) (1993) A Guide to Usability: human factors in computing. Addison Wesley, Open University. 1993
- Stapleton. J. (1999) *DSDM- Dynamic Systems Development Methodology- The Method in Practice*
- Stowell. F.(1995), *Information Systems Provision, The Contribution of SSM,* McGraw-Hill Book Company, London
- Thomsett. Rob (1998), *Third Wave Project Management- A Handbook for managing the complex Information System*
- Gerald Forest Hice, (1999), *Systems Development Methodologies 1999*
- Hofmann (1988), *Formalised Systems Development Methodologies, A Critical Perspective*
- Checkland. P. and Scholes. J. (1997), *Soft Systems Methodology in Action*
- Olle, W T. (1991*), Information Systems Methodologies: A framework for understanding,* Addison-Wesley Publishing Company, Wokingham.
- Pressman. R. S. (2001), *'Software Engineering: A Practitioners Approach',* McGraw-Hill Publishing Company, London
- Roger, Y.; Sharp, H., and Preece, J., (2011) Interaction Design: beyond human-computer interaction. (3rd Edn) John Wiley, 2011. ISBN 978047066576-3
- Wilson. Brian (1999) *Systems, Concepts, Methodologies and Applications,* Wiley and Sons Publications, Chichester and New York.
- http://portal.acm.org/citation.cfm?id=1216224&dl=ACM&coll=ACM&CFID=89775628&CFTOKEN=69361880
- Ivan Tirado (2008), Business oriented information security requirements development, Published by ACM, Available at: http://portal.acm.org/citation.cfm?id=1456625.1456642&coll=ACM&dl=ACM&CFID=89775628&CFTOKEN=69361880

- An article from Thai Indian News dated May 8th 2008, HSBC admits huge data loss, Available at:http://www.thaindian.com/newsportal/world-news/hsbc-admits-huge-data-loss-in-hong-kong_10046281.html
- An article from Times Online dated March 11th 2010, HSBC admits huge scale of Swiss data theft, Available at: http://business.timesonline.co.uk/tol/business/industry_sectors/banking_and_finance/article7057981.ece [Accessed: 13/03/2010]
- Sophos, Security Threat report 2009, Available at:http://www.sophos.com/sophos/docs/eng/marketing_material/sophos-security-threat-report-jan-2009-na.pdf [Accessed: 13/03/2010]
- An article from BBC Online dated March 11th 2010, HSBC admits huge Swiss bank data theft, Available at: http://news.bbc.co.uk/1/hi/business/8562381.stm [Accessed: 13/03/2010]
- An article from The Register Online dated March 20th 2009, Indian call centre credit card scam exposed: Symantec renewal details end up on black market, Available at: http://www.theregister.co.uk/2009/03/20/call_centre_credit_card_fraud/ [Accessed: 23/03/2010]
- PCI data security standards.[Online] Available at: https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml [Accessed: 23/03/2010]
- Ferraiolo D.F. and Kuhn, D. R (1992) "Role Based Access Control" 15th National Computer Security Conf. Oct 13-16, 1992, pp. 554-563. Available at: http://csrc.nist.gov/rbac/ferraiolo-kuhn-92.pdf [Accessed: 16/05/2010]
- Kelly, S.; Gibson, N.; Holland, C.; and Light, B., (July 1999). "Focus Issue on Legacy Information Systems and Business Process Engineering: a Business Perspective of Legacy Information Systems". *Communications of the AIS* **2** (7): 1–27.
- McEwan, T., (2010) Digitally Enabled. The BCS IT Now Magazine, November 2010, BCS UK.
- Pearson Custom Publishing & West Chester University, Custom Program for Computer Information Systems (CSC 110), (Pearson Custom Publishing, 2009) Glossary p. 694
- Jessup, L. M.; Valacich, J. S. (2008). *Information Systems Today* (3rd ed.). Pearson Publishing. Glossary p. 416

- Rubin, J. (1994). *Handbook of usability testing: how to plan, design, and conduct effective tests*. Wiley, 1994.
- John, B. E., & Mashyna, M. M. (1997) Evaluating a Multimedia Authoring Tool with Cognitive Walkthrough and Think-Aloud User Studies. In *Journal of the American Society of Information Science*, 48 (9).
- Howard, J. D. (1997) *An Analysis of Security Incidents on the Internet 1989-1995*. Carnegie Mellon, University Ph.D. Thesis, 1997.
- Pretty Good Privacy, Inc. *PGP 5.0 Features and Benefits*. Available online at: http://pgp.com/products/PG*P50-fab.cgi, 1997. (Accessed* May 2010)
- The Open Group Research Institute. *Adage System Overview*. Available online at http://www.osf.org/www/adage/relatedwork.htm, July 1998. (Accessed August, 2010)
- Wharton C., Rieman, J., Lewis C. and Polson, P. (1994) The Cognitive Walkthrough Method: A Practioner's Guide. In *Usability Inspection Methods*, John Wiley & Sons, Inc., 1994.
- PGP. (1997-2008). *User's Guild for the PGP for personal privacy version* . http;//pgp.com/product/pgp50-gab.cgi,1997.
- Rubin, J. (1994). *Handbook of usability testing , how to plan, design and conduct effective test* . john wiley.
- Whitten, A. &.Tygar, J. D. (1998). Usability of security a case study . *school of computer science EECs Carnegie mellon University* .
- Zurko, M. &. (1997). User Centric security . *The pen Group Research institute cambridge* .

Web Links

- www.usablebrands/bbc
- www.usabilitynet.org
- www.samsvb.co.uk/.../development_life_cycle.gif
- www.cw360.com
- www.eds.com/case_studies/cs_home.shtml
- www.yourdon.com
- www.sdmagazine.com

- www.research.ibm.com/journal/
- www.intranetjournal.com/itadvisor/
- www.cio.com
- www.e-businessreview.co.uk
- www.e-consultancy.com
- www.nua.ie/surveys
- www.findarticles.com
- www.moreover.com
- www.ecommercetimes.com
- www.web-ee.com
- http://en.wikipedia.org/wiki/Information_systems

Appendix A

Publication List

- Arreymbi, J and Dastbaz, M. (2002) *Issues in Delivering Multimedia Content to Mobile Devices*. Proceedings of the 6th International Conference on Information Visualisation, IEEE Computer Society, London, UK. July 2002.
- Arreymbi, J (2008), Investigating Issues in Mobile Network Security. Journal of Modern Applied Science. Vol. 2 (3) May, 2008. ISSN: 19131844 & ISSN: 19131852
- Arreymbi, J., Dastbaz, M. (2008), "Issues with Information Visualisation and Mobile Devices" In Proceedings of IEEE IV08, London 2008
- Arreymbi, J. and Dastbaz, M. (2008), *Mobile Handset Constraints in Designing for Multimedia Interactivity*. Proceedings of the International Conference on Information Visualisation, IEEE Computer Society, London, UK. 2008 (Awaiting publication)
- Arreymbi, J and Draganova, C. (2010) User Requirements analysis for use of mobile phones in learning and teaching. In Claes, T. and Preston, D. S. (eds), Frontiers in Education. Rodopi, Amsterdam, 2010
- Arreymbi, J. Agbor, E, Dastbaz, M. (2008), "Mobile-Education - A paradigm shift with Technology", in the Proceedings of ED MEDIA 2008, Vienna
- Arreymbi, J. (2005), *Online Banking: Spoofing Scams exposes Security Loopholes*. In Securing Electronic Business Processes. Paulus, S., Pohlmann, N., and Reimer, H.; (Eds). ISSE 2005, Vieweg publishers. Germany 2005. ISBN:3-8348-0011-2 http://www.ecampus.com/book/9783834800114
- Arreymbi, J. and Williams G. (2005), *Assessing the Economics of Electronic Security.* In Securing Electronic Business Processes. Paulus, S., Pohlmann, N., and Reimer, H.; (Eds). ISSE 2005, Vieweg publishers. Germany. ISBN:3-8348-0011-2
- Williams, G. and **Arreymbi. J**. (2007), I*s Cyber tribalism winning Online Information Warfare?* In Securing Electronic Business Processes. Paulus, S., Pohlmann, N., and Reimer, H.; (Eds). ISSE/SECURE 2007, Vieweg publishers, Germany (ISBN: 978-3-8348-0346-7)

- Arreymbi, J. (2007), *Mobile Handset Constraints in Designing for Interactivity and Usability*. Research in Interactive Design, Springer Verlag, 2007 (Awaiting publication).

- Arreymbi, J. (2005), *Phishing Attacks – A Socially Engineered Threat to e-Business*. Proceedings of the International Conference on Internet Computing. Las Vegas, Nevada, 2005
- Arreymbi, J. (2005), *Risk and Trust in Online Banking*. Proceedings of the 1st IEEE Annual International Conference on Advances in Information and Communication Engineering. Accra, Ghana, 2005. ISBN: 988643136
- Arreymbi, J. (2006) *Modelling to enhance GSM Network Security*. Proceedings of the International Conference on Security and Management. Las Vegas, USA, 2006.
- Arreymbi, J. (2006), *Enhancing the Logistics of Product Identification and product Security using Radio Frequency Identification (RFI*D). Proceedings of the 2ND IEEE Annual Conference on Advances in Information and Communication Engineering. Accra, Ghana, 2006. ISBN: 988643136 - Volume 2
- Arreymbi, J. and Al-Zakwani, A. (2006) *Modelling Service Delivery System for Wireless Handsets using Instant Alert and Frequency Scanning Technologies: An approach for Bus Service application*. Proceedings of the International Conference on Internet Computing & Computer Games Development, Las Vegas, USA, 2006.
- Arreymbi, J. and Gachanga, E.W. (2006), *Interactive Design and Delivery Challenges for wireless Handheld Multimedia Systems*. Proceedings of the International Conference on Internet Computing & Computer Games Development, Las Vegas, USA, 2006.
- Honsy, W. and **Arreymbi, J**. (2007), *Adapting mobile Technologies for Education and Learning (m-Learning)*. Proceedings of the International Conference on Internet Computing & Computer Games Development, Las Vegas, USA, 2007.
- Arreymbi, J. (2007), *Examining Security in mobile Communication Networks*. Proceedings of the 2nd Annual International Conference on Advances in Computing and Technology, London, UK, 2007
- Arreymbi, J. and Al-Zakwani, A. (2007), *A Performance Analysis of Mobile Network Gaming Architecture: Measuring Handset OS functionality and Reliability in real-time*. Proceedings of the International Conference on Internet Computing & Computer Games Development, Las Vegas, USA, 2007.

- Adnan, A., Williams, G., Arunachalam, S. Cazan, A. **Arreymbi, J**. (2007), Analysing *Communication System for Successful Outsourcing – A Games Theory Perspective*. 3rd Annual International Conference on Advances in Information and Communication Engineering. London UK, 2007
- Arreymbi J. (2007) *Evaluating the impact of Mobile Devices in Human-Computer-Communication for Sustainable Development*. 3rd Annual International Conference on Advances in Information and Communication Engineering. London UK, 2007
- Adnan, A. Arunachalam, S. Cazan, A. Arreymbi, J. & Webb, P. A. (2008) *Developing an Outsourcing Questionnaire: Validation Study*. CITE AC& T Conference, London, January 2008.
- Arreymbi, J. Agbor, E.A. & Adnan, A. (2008), Why ICT uptake is slow in Developing Economies: A case of Cameroon. CITE AC & T Conference, London, 2008

Appendix B

Some Usability models for quality evaluation

Here we will be analysing ISO quality model standards on usability. The diagrams below are a simple overview of ISO 9126 standard. The ISO 9126 is a standard for software production evaluation: It specifies quality characteristics and guidelines for the use. The standard defines model of quality. ISO 9126 defines quality as totality of features and characteristics of a software product that bears on its ability to satisfy implied or stated needs (www.chrisbunney.com, 2009). Quality according to the standard is divided into the following six attributes:

- Functionality
- Reliability
- Efficiency
- Usability
- Portability
- Maintainability

The standard proposes how the quality factors can be further sub-divided into quality attributes as illustrated in the diagrams

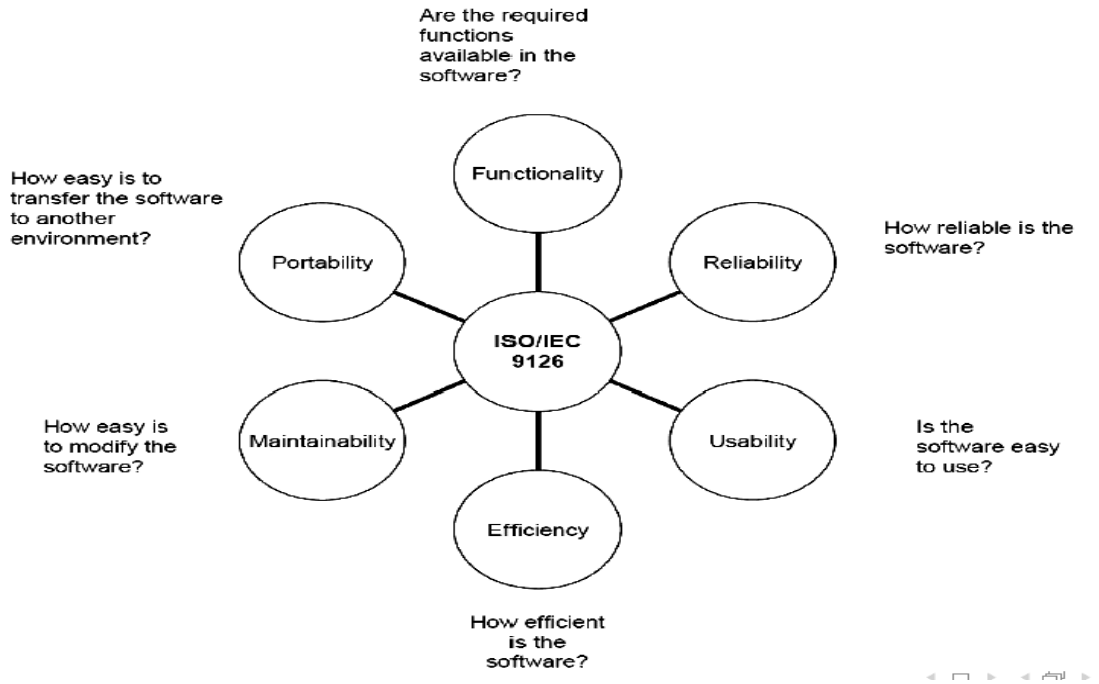


Figure B1: Quality factors and attributes (ISO 9126 - 2001).

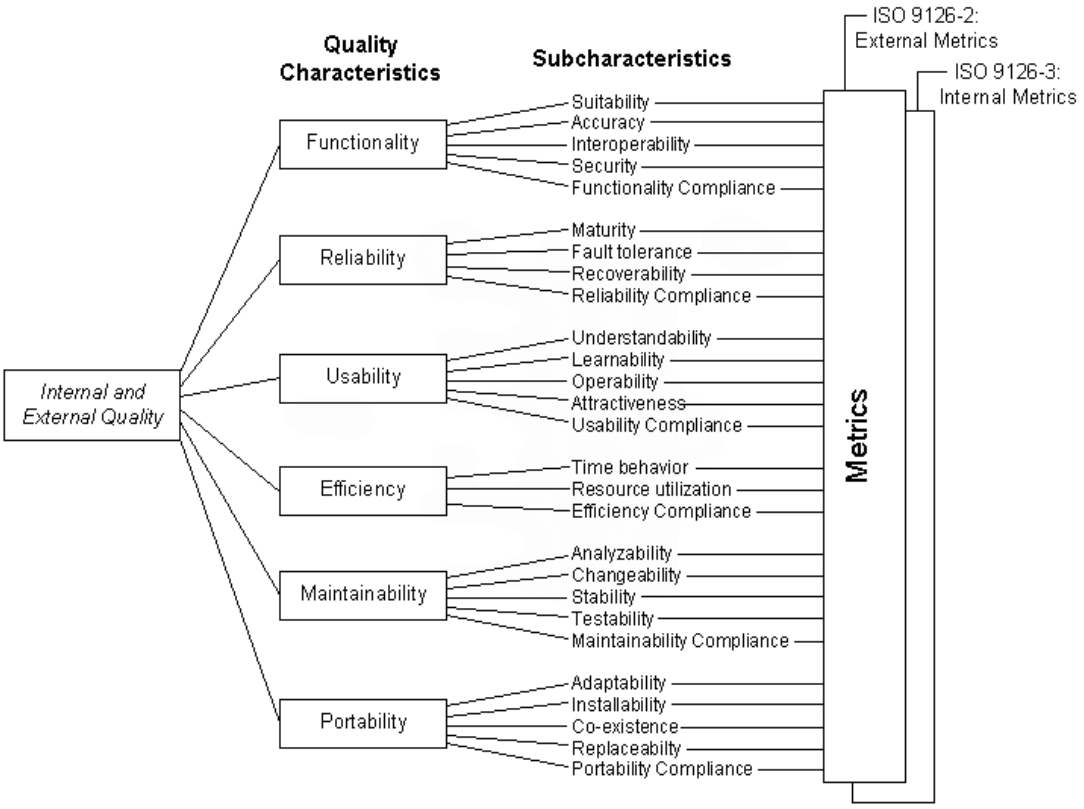


Figure B2: Quality Characteristics metrics (ISO 9126 -2001).

Standards related to usability: Usability according to usability net (2000), can be classified into the following:

- Use of the product (effectiveness, efficiency and satisfaction in a particular context of use.
- The user interface and interaction
- Processes used in developing products
- Capability of organisation in applying user-Centred-Design

ISO – 9241-11 Guidance on Usability (1998); and ISO- 9241 -210 Human-Centred Design Process for Interactive Systems. (2007)

The ISO 9241-210 (2007) has replaced the ISO 13407 (1999)

The ISO: 9241-11: Guidance on usability (1998)

The standard defines usability – as applicable to previous ergonomic related standards. Usability is the extent to which a product can be used by specified users in achieving specified goals in a specified context of use. The standards explain how to identify information: that which is needed to take into consideration when specifying or evaluating usability in terms of measures of user performance and satisfaction. Guidance is given on how to describe context of use of the product and measures of usability in an explicit way. It explains how usability of a product can be specified and evaluated as part of quality standards for example one that conforms to ISO 9001 standards. The standard explains how measures of user performance and satisfaction can be used in measuring how components of a work system affect the quality of the whole work system in use.

Criticism of ISO: 9241-11

Despite the emphasis of ISO: 9241-11 standards on usability, it has received criticism from IS specialist and industry experts an example of such criticism as explained in an article from (UserFocus, n.d), the quality model standards introduces the concept of quality, but does not make specific recommendations in terms of product attributes.

Another critic of the model is regarding the word 'satisfaction', specialists in the industry consider satisfaction as similar to weak, inadequate or just good enough. However, in the current English dictionary, satisfaction is defined as feeling of pleasure that comes when a need or desire is fulfilled. **ISO/IEC 9126: Software Product evaluation** –Quality Characteristics and guidelines for their use (1991)

In software engineering, usability has been more narrowly linked with User Interface design (usability net, 2000). The ISO/IEC 9126 was implemented separately as a software engineering standard on quality. It defines usability as a relative contribution to software quality, which is associated with the design and evaluation of user interface and interaction. The standard looks usability as a set of attributes that bear on the effort, needed for use, and on the individual assessment of such use by a stated or implied set of use.

ISO/IEC FDIS 9126-1: Software Engineering-Product quality-Part I: Quality Model (2001)

The ISO/IEC 9126 (1991) has been replaced by a four part standard that reconciled two aspects of usability. ISO/IEC 9126-1 depicts the same six categories of software quality that are relevant during product development. These are as follows:

- Functionality
- Reliability
- Efficiency
- Usability
- Portability
- Maintainability

This is illustrated in figure below (usability net, 2000)

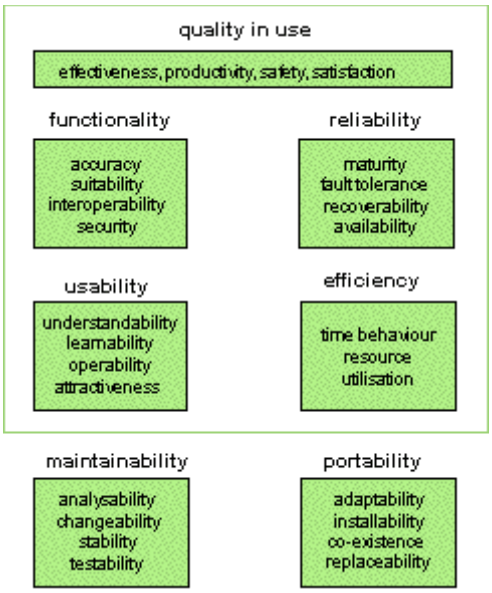


Figure B3 Six categories of software quality

The standard elaborates that usability plays two roles: a detailed software design activity (implied by the definition of usability) and an all-inclusive goal that software meets user needs as illustrated by the diagram- ISO/IEC 9126-1 uses the term quality in use for the broad objective.

Quality in use is the combined effect of six classification of software quality when the product is in use. The overall objective is to achieve in use for the end user and support user. Functionality, reliability, efficiency and usability determine quality in use for an end user in a particular context; the support user is concerned with quality in use of maintenance and portability tasks.

ISO/IEC DTR 9126-4: software Engineering –Product quality-Part 4: Quality in use metrics.

This is a technical report that specifies examples of metrics for effectiveness, productivity, safety and satisfaction. The standard suggests metrics for effectiveness, productivity and satisfaction that can be used to verify quality in use. The results can be documented using the industry standard template for usability test reports, which is attached as an annex to the standard.

Software Interface and interaction: The following standard according to (Usability net, 2000) can be used to support user interface development:

- To specify details of the appearance and behaviour of the user interface, ISO 14915 and IEC 61997 specify recommendations for multi-media interfaces. Specific guidance for icons can be found in ISO/IEC 11581, PDAs in ISO/IEC 18021 and cursor control in ISO/IEC 107421
- To provide detailed guidance on the design of user interfaces (ISO - 9241 parts 12-17)
- To provide criteria for evaluation of user interfaces(ISO/IEC 9126 parts 2 and 3)

Documentation

ISO/IEC 15910 provides a detailed framework for the development of user documentation (Paper and on-line help), however, ISO/IEC 18019 provide more guidance on how to produce documentation that user needs.

ISO/IEC 15910-software user documentation process (1999)

The standard cover details of the minimum process required for creating user documentation for software that has a user interface which includes printed documentation (e.g. user manuals and quick-reference cards) online documentation help text and on-line documentation systems.

BS 7649: Guidelines to the design and preparation of documentation for users of application of software (1993)

BS 7830: guidelines to the design and preparation of on-screen documentation for users of application software (1996). The standard is intended to endorse ISO /IEC 9127-user

documentation and covers information for software packages and ISO /IEC 15910 software user documentation process.

The Development Process

ISO 9241-210 (2007) formerly ISO 13407 (1999) explains activities required for developing user centred designs. ISO /TR 16982 (2002) outlines the types of method that can be used. ISO/IEC 14958 provides a general framework for evaluating software products using the model in ISO/IEC 9126-1

ISO 9241 -210 Human centred design processes for interactive systems (2007). The standard specifies guidance on human-centred design activities throughout the life cycle of interactive computer based systems. It depicts human centred design as a multidisciplinary activity which include human factors and ergonomics knowledge and techniques with the objective of enhancing effectiveness and efficiency which improves working conditions and counteract possible adverse effects of use on health and safety and performance as illustrated as the following process (usability net, 2000)

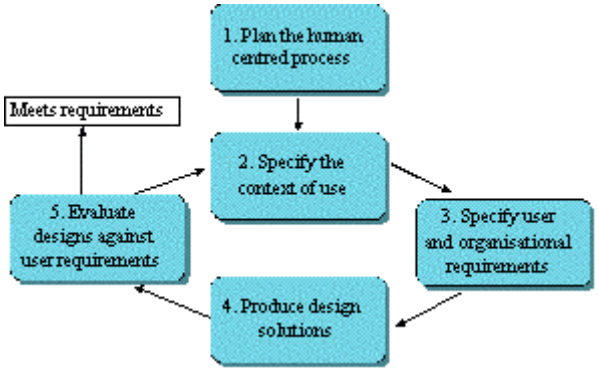


Figure B4 Human-centred design process (source: usability net, 2000)

Capability of the organisation: The Usability model in ISOTR 18529 contains an organised set of processes derived from ISO13407 and a survey of good practice. It can be applied to assemble the extent to which an organisation is capable of implementing user-centred design. Each of the HCD process (such as specify the user and organisation requirements can be rated on ISO 15504 software process assessment scale which are the listed as follows:

- Incomplete
- Performed

- Managed
- Established
- Predictable or optimising

Criticism of ISO quality standard model

Despite the usefulness of the ISO quality standard model in providing a framework for usability, IS specialist have expressed their concern on shortcomings of the model. In a recent report from QualitySIG, Newsletter (2002), computing specialist and users explain that ISO system provides appearance of quality through documentation, without quantitative demonstration of output or outcome.

On the shortcomings of the standard, it is argued that there is less emphasis on the Plan and Acts parts of the PDCA cycle of Plan-Do-Check-Act ,although ISO standards addresses the need for design planning with considerations for specific particular design inputs. The cycle in the standard is not associated with the design stage where planning is crucial and acting must take place.

Another criticism of the quality model is that the system does not discuss the needed practice. Documentation of corrective action is emphasised over determination of cases. There is no reference to examining the design. Focus is on quality system rather than examining component processes such as Six Sigma or systematic continual improvement and associated tools are rarely documented as preventative measures in a quality system