The University of Pardubice
Faculty of Economics and Administration
Department of System Engineering and Informatics

The complexity of business information systems

Frank Febiri

Master Thesis

2019

University of Pardubice
Faculty of Economics and Administration
Academic year: **2018/2019**

DISSERTATION ASSIGNMENT

(PROJECT, ART WORK, ART PERFORMANCE)

First name and surname:	**Frank Febiri**
Study program:	**N6209 System Engineering and Informatics**
Identification number:	**E17920**
Specialization:	**Regional and Information Management**
Topic name:	**The complexity of businesses information systems**
Assigning department:	**Institute of System Engineering and Informatics**

Rules for elaboration:

Objective: To describe the current ways of assessing the complexity of information systems to assess the complexity of the information system of the selected company.

Content:

- analysis of selected information systems,
- analysis of the current ways of assessing the complexity of information systems,
- assessment the complexity of the information system of the selected company.

Scope of graphic works:

Scope of work report
(scope of dissertation): **Approximately 60 pages.**

Form of dissertation elaboration: **printed/electronical**

List of specialized literature:

BAŻEWICZ, Jacek. Handbook on data management in information systems. Berlin: Springer-Verlag, c2003. ISBN 3-540-43893-9.
STAIR, Ralph M a George Walter REYNOLDS. Information systems. 9th ed. Boston: Course Technology, c2010. ISBN 978-0-538-47425-2.
STAIR, Ralph M a George Walter REYNOLDS. Principles of information systems: a managerial approach. 7th ed. Boston: Thomson Learning, c2006. ISBN 0-619-21525-9.

Tutor for dissertation: **doc. Ing. Miloslav Hub, Ph.D.**
Institute of System Engineering and Informatics

Date of dissertation assignment: **3 September 2018**
Date of dissertation submission: **30 April 2019**

doc. Ing. Romana Provazníková, Ph.D.
Dean

L.S.

doc. Ing. Pavel Petr, Ph.D.
Department Manager

In Pardubice, dated: 3 September 2018

DECLARATION

I hereby declare:

This thesis was prepared separately. All the literary sources and the information I used in the thesis are listed in the bibliography. I got familiar with the fact that the rights and obligations arising from the Act No. 121/2000 Coll., Copyright Act, apply to my thesis, especially with the fact that the University of Pardubice has the right to enter into a license agreement for use of this thesis as a school work pursuant to § 60, Section 1 of the Copyright Act, and the fact that should this thesis be used by me or should a license be granted for the use to another entity, the University of Pardubice is authorized to claim a reasonable contribution from me to cover the costs incurred during making of the thesis, according to the circumstances up to the actual amount thereof. I am aware that my thesis will be accessible to the public in the University Library and via the Digital Library of the University of Pardubice in agreement with the article 47b of the Act No. 111/1998 Coll., on Higher Education Institutions, and on the Amendment and Supplement to some other Acts (the Higher Education Act), as subsequently amended, and with the University Pardubice's directive no. 9/2012.

In Pardubice on April 30, 2019 **Frank Febiri**

ACKNOWLEDGMENTS

Firstly, I am most grateful to our creator, Almighty God. He is the source of my wisdom and strength. I thank Him for His special protection and Grace throughout this thesis work.

I also would like to appreciate my supervisor doc. Ing. Miloslav Hub, Ph.D. for impacting me with his academic excellence through his knowledge transfer, constant assessment, and inputs. I have personally benefited enormously from his soft skills such as time management, communication skills and words of encouragement. I want to say thank you. I want to thank Mr. Emmanuel Jefrey - Manager and Head of SWD (Foxconn.cz), Mr. Radek Cerman; VENDORWEB Manager (Foxconn cz) and the team members; Petr Suchy, Petr Klouček, Radek Novotny, Petr Dusek, all members of Web and Internal apps Team. God bless you all for your massive contribution to this thesis work.

Finally, I want to thank some special people in my life who always pray with and for me and also gave me moral guidance throughout my study; Miss Doreen Kusi, Miss Lois Agyei-Mensah Pokuah. To my parents, the backbone behind my studies Mr. and Mrs. Febiri, God bless you so much.

ANNOTATION

A great deal of controversy exists about the impact of information systems on a firms performance. While some authors have reported positively that today's business success is solely built on a business's information system, others argue that information systems are gradually taking over and making businesses to lose their control. Business Information systems appear to be a dream come true to solve the increasing demand by customers for efficient and quality delivery of goods and services. This has necessitated many businesses and organizations in the 21st century to go through a radical transformation in building quality and highly integrated information systems to cater for and solve all the needs of the customers whilst keeping track of internal and external business processes and records. These information systems packages which are currently tailored to suit organizational purposes promise the all-in-one amalgamation of all the information curving through an organization. Different literatures admit information systems development (ISD) to be a complex activity. This complexity is magnified by the daily and continuous changes in user requirements due to changing business needs which are triggered by the speed of customer needs in a fast-changing external competitive environment. Many researchers have concluded that if this increasing complexity is not managed appropriately, information systems may fail and turn to impact organizations more than it's anticipated benefits.

Writers of various kind of literature have proposed various methods of assessing and measuring the complexity of an organizations information system. The main aim of this thesis was to identify the current ways of assessing the complexity of business information systems and using that as a basis to measure the complexity of the information system of Foxconn – a manufacturing company in the Czech Republic. The thesis uses the function point method mainly in measuring the information system of Foxconn. From the analysis, the derived level of complexity of the company and its impacts are shown.

KEYWORDS

Business information system, Function point, Information technology.

ANOTACE

V současné době existuje velký spor o dopadu informačních systémů na výkonnost firem. Zatímco někteří autoři hodnotí pozitivně, že dnešní obchodní úspěch je postaven výhradně na podnikových informačních systémech, jiní tvrdí, že informační systémy postupně přebírají kontrolu nad obchodními procesy, kterou pak samotné podniky ztrácejí. Obchodní informační systémy se zdají být splněním snu pro vyřešení rostoucí poptávky zákazníků po efektivním a kvalitním dodání zboží a služeb. To si vyžádalo, aby v 21. století mnoho podniků a organizací prošly radikální transformací v budování kvalitních a vysoce integrovaných informačních systémů, které budou zajišťovat a řešit všechny potřeby zákazníků a zároveň sledovat vnitřní a vnější obchodní procesy a záznamy. Balíčky informačních systémů jsou v současné době přizpůsobeny tak, aby vyhovovaly organizačním účelům a slibují sloučení všech informací v organizaci. Různá literatura připouští, že vývoj informačních systémů je komplexní činností. Tato složitost je umocněna každodenními a neustálými změnami požadavků uživatelů v důsledku měnících se obchodních potřeb, které jsou vyvolány rychlostí změn potřeb zákazníků v rychle se měnícím externím konkurenčním prostředí. Mnozí výzkumníci dospěli k závěru, že pokud tato rostoucí složitost není řádně řízena, informační systémy mohou selhat a obrátit proti organizaci, na kterou mají dopad, více, než se předpokládalo.

Různí autoři navrhli různé metody hodnocení a měření komplexnosti informačního systému organizace. Hlavním cílem této práce bylo identifikovat současné způsoby hodnocení komplexnosti podnikových informačních systémů a využít je jako základ pro měření komplexnosti informačního systému společnosti Foxconn – výrobní společnosti v České republice. Práce využívá metodu funkčního bodu hlavně při měření informačního systému společnosti Foxconn. Z analýzy je ukázána odvozená úroveň složitosti společnosti a její dopady.

KLÍČOVÁ SLOVA

Podnikový informační systém, Funkční bod, Informační technologi

CONTENT

LIST OF TABLES 10

LIST OF FIGURES 10

LIST OF ABBREVIATIONS 11

1. INTRODUCTION 12

1.1 Current State of Information Systems 14

1.2 Review of Current State 15

1.3 Objective of the Study 16

1.4 Justification of the Study 16

1.5 Significance of the Study 16

1.6 Organization of the Study 17

2. LITERATURE REVIEW 18

2.1 Defining Data and Information 18

2.2 Defining Systems 19

2.3 Defining Information Systems 19

2.4 Business Information Systems 20

2.4.1 Types of Business Information Systems 21

2.4.2 Components of Business Information Systems 21

2.4.3 The Architecture of Information Systems 23

2.5 Developing Information Systems 24

2.5.1 The Traditional Waterfall Model 24

2.5.2 Agile Software Development 24

3. EVALUATION OF COMPLEXITY 26

3.1 Defining Complexity 27

3.2 Components of Information System Complexity 27

3.3 Locus of Complexity: Organization versus Informational Technology 29

3.4 Complexity versus Simplicity of Information Systems 30

3.5 The Need for Estimation 31

3.6 Methods of Estimating Complexity 31

3.7 Function Point Method..32

3.7.1 Using Function Point Analysis 32

3.7.2 Function Point Analysis..33

3.7.3 Using Function Point Values 34

3.7.4 Function Point Analysis Caveats 35

3.8 Data on Document Method 36

3.8.1 Design Complexity Based on Quantity of Data on Documents 36

3.8.2 Estimation Elements in Data on Document Method 36

3.8.3 Stages of Data on Document Method Application 40

3.9 Constructive Cost Model..40

3.10 Database Complexity Method..43

3.11 Why Function Point as the Chosen Method...43

4. COMPLEXITY OF ORION USING FUNCTION POINT METHOD 45

4.1 The Business of Foxconn 45

4.2 The Architecture of Foxconn Information System 46

4.3 Calculating Crude Function Point...48

4.4 Calculating Relative Complexity Adjustment Factor 52

4.5 Calculating Value Adjustment Factor..53

4.6 Calculating Function Point...53

4.7 Estimated cost and Project time 53

5. CONCLUSIONS AND RECOMENDATIONS 55

5.1 CONCLUSION 55

5.2 RECOMMENDATIONS FOR FURTHER STUDIES 57

REFERENCES 58

LIST OF TABLES

Table 1: Factors in FPA 34
Table 2: CoCoMo Cost Drivers 41
Table 3: Project Modes 42
Table 4: Complexity standard of ORION 48
Table 5: Complexity matrix for FP function components 49
Table 6: Function component complexity weight assignment 49
Table 7: Complexity Values for inputs 50
Table 8: Complexity values for Logical Files 50
Table 9: Complexity values for outputs 51
Table 10: Complexity values for Inquery 51
Table 11: Complexity values for interface 51
Table 12: General system characteristics 52

LIST OF FIGURES

Figure 1: Basic Sources of Complexity 28
Figure 2: Interrelated system 29
Figure 3: Model of Data on Documents method 38
Figure 4: Architecture of Orion 46

LIST OF ABBREVIATIONS

BIS	Business Information System
BPR	Business Process Re-Engineering
BPM	Business Process Modelling
COCOMO	Constructive Cost Model
CFP	Crude Function Point
DC	Database Complexity
DOD	Data on Document
DSS	Design Support Systems
EAF	Effort Adjustment Factor
EDI	Electronic Data Interchange
EI	External Input
EIF	External Interface File
EO	External Output
FP	Function Point
IS	Information Systems
IT	Information Technology
IFPUG	International Function Point User Group
MIS	Management Information System
OIS	Operating Information System
SFC	Sequential Function Chart
TPS	Transaction Processing System
VAN	Value Added Network

1. INTRODUCTION

An increasingly competitive and dynamic business world has forced managers of large organizations to be in an endless search to strengthen their organizations on the one hand and to identify weaknesses of their competing rivals to arrive at new opportunities (Jofre, 2011).

One major tool organizations adopt today, focuses on, and spends enormous sums of money is to build competitive and robust information systems. A big competing challenge such organizations face as they grow in size and complexity is how to keep their systems up to date, integrated with current business processes and in sync with each other so that information can flow, be shared, provide intelligence and support for decision making while minimizing duplication of records (Walrad, 1993).

Company engineers and systems administrators specialize in optimizing the information flow and the supporting technology of the organization so that the organization can meet their business objectives at a lower cost and with a foundation not just to operate but also to anticipate and accommodate future requirements (Tait, 1988).

Information systems (IS) receives, process, store, and output the processed information to satisfy the needs of their beneficiaries. The architectural view of an IS represents high-level components of a specific attributes of the system. An essential aspect of the IS is seen from the architecture of the IS. The architecture allows users and interested parties envision how transfer of information among system components is achieved in the IS. Understanding and evaluating the complexity of this information architecture provides stakeholders with the best and suitable architecture from an information standpoint.

The complexity of IS design stems from the fact that, besides, a technological basis, it implies and covers business processes, and reflects on the final organizational goal, time, costs, and quality (Banker, 1993). Furthermore, the design complexity indicates on managing human resources, such as a team, users and top management support. However, not much work and time has been invested into measuring the complexity of IS, what is spent mainly applies to measure the complexity of the software which is just one aspect of the whole IS. Even the function point method has a primary focus in measuring the complexity of the software itself. The question is whether the complexity of an IS can be estimated before building such IS? Here, the emphasis is on estimation, using the available business operations and requirements of the business. Many extensive works have been done on the problem of estimating the complexity of Business information systems (BIS), yet little has been achieved. .ISs can be categorized into many forms. These include; transaction processing system (TPS) or management information system(MIS), ranging from a simple application on a smartphone, desktop applications, and enterprise applications, to the World-Wide Web itself (Hardcastle, 2008)

Complexity is a vital non-functional requirement for software architectures as this gives a good indication of the difficulty of maintaining the system. Typically, the more complex the design, the more difficult it is to keep the system. Since maintenance is the most expensive part of the IS, it will be of interest to software practitioners to have a measure of the complexity at the architecture level itself. Few methods exist in the literature for evaluating IS complexity.

This work analyses some methods of evaluating the complexity of ISs with the central focus on the information systems of Foxconn, a multinational manufacturing company in Pardubice - Czech region.

1.1 Current State of Information Systems

A continuing stream of IS, innovation, combined with new ways of doing business and changes in the manner in which we transact business with different organizations, generate revenue, and how customers receive products and services have brought about the need of superb IS. The growth of enterprise-wide IS that provide extraordinarily rich data to managers, customers, suppliers, and employees, ensures that managers no longer have to work in a haze of perplexity but rather, have on the web, instant access to the vital information they need to make accurate and timely decisions that influence organization performance.

IS are essential components of every business. Much of a business's investment is in the IS and associated technologies (O'brien & Marakas, 2005). This is because there is a growing interdependence on a firm's ability to develop strong, competitive IS and capitalizing on the developed IS to achieve its objective as well as execute its strategies. The rapid growth of the personal computer industry, substantial decreases in computer unit cost, and the simultaneous increase in computer capabilities have made the vast amount of information readily available to individuals in organizations (Venkatesh & Davis, 2000).

The goal of every IS, in any organization, is to improve job performance, and this performance efficiency is only achieved when Information Technology (IT) is accepted and used warmly by the concern employees in organizations (Venkatesh & Davis, 2000). Performance of an organization is the priority of managers of today. Organization performance is defined as continuous and action-oriented with a focus on improving performance by using objective, standards, appraisal, and feedback (Hunger & Wheelen, 2003). An Organization's performance comprises the actual output of an organization‘s achievement compared to its intended goals and objectives. Organizations adopt performance measurement because it creates accountability, provides feedback to operations, and result in more effective planning, budgeting, and evaluation using it IS (Ammons, 2001).

As quoted from (Walrad & Moss, 1993), ‘‘Efficiency and effectiveness do not mean the same thing. Often one can have one but not both (Unless one is lucky or one wants to spend a lot of money). Being efficient means spending little time on an activity, one spends less money on something or one spends fewer efforts (or the number of workers) on something”.

Most organizations today rely on IS as an integral part of their operations to achieve efficiency and effectiveness in their business processes. Critical elements of every organization are its people, structure, business processes, politics, and culture. From the preceding, it is evident that, for a firm to achieve its business objectives and survive today's competitive business environment it needs significant investment in its ISs. Such investment is in hardware, software, data management, and telecommunications technologies (AbuDoleh & Weir, 2007).

The strategic objectives and business processes of the organization will, in part, depend on the ISs available to it. There is thus a two-way relationship between business and IS. Businesses rely on IS to help them achieve their goals, and IS are also products of the firms that develops them. Therefore businesses build ISs, and ISs helps businesses to function

1.2 Review of Current State

One trend with the 21st - century business world, particularly the act of conducting business, is the increasing and interconnections between economic agents and actors (consumers, suppliers, banks, markets participants, etc.) (Tagra, 2011). An essential development is taking place in the nature and application of technology in modern business, a change with reflective and fast impact on every company irrespective of its size and shape. A recent research conducted by the DMR Group, Inc., which studied over 4,500 organizations across the globe to find the nature and impact of changes in technology on business was evident of the impact caused by changes in technology on companies. The fusion and analysis of this information indicate that information technology is going through its second paradigm shift (Vasconcelos & Ramirez, 2011). Driven by the demands of the competitive business environment and intense changes in the nature and technologies used in conducting business, the information age is climbing speedily on high technological ladder.

Every computing platform in most organizations today is striving to catch up with the current trend for corporate reawakening. Managers and professionals in the business field are thus learning and finding new ways on how to take instantaneous action for the immediate benefits through the new technological trend while positioning their organizations for a sustainable growth and transformation (Xia & Lee, 2004).

This implies that such agents are all interacting and consequently giving rise to enormous degrees of non-linearity, hence the complexity of the current trend. Developing systems and models to solve the complex nature of the trend has, therefore become inevitable (Bazewicz, 2003). This thesis thus focuses on the complexity of business, information systems and how such complexities can be measured. It further applies the function point method in analyzing the complexity of the IS of Foxconn, a chosen company solely for the purposes of the readers understanding.

1.3 Objective of the Study

In line with the issues discussed above, this study aims at describing the current ways of assessing the complexity of the IS of Foxconn, a manufacturing company using a highly integrated information system. The specific objectives of the study include:

- Analysis of selected information systems.
- Analysis of the current ways of assessing the complexity of information systems.
- Assessing the complexity of the information system of Foxconn.

1.4 Justification of the Study

In today's business world, the size of the market is not limited to the local market or the organization's environs but expands broadly across the entire globe (Amaratunga, 20001). The broadened market size requires businesses to develop and manage effectively complex information systems that will factor all the activity of the organization both local and international. This brings about a high demand between organizational structures and systems as against market standards and requirements. To resolve this conflict, measures on how to develop a capable and complex organizational IS has becomes a necessity. The Complexity of IS thus has become an inevitable part of modern organizations. The need to access and measure the complexity serves as a winning tool for organizations in today's highly competitive business environment (Atim, Tati & Widianti, 2018).

1.5 Significance of the Study

To get the investment put into IT, organizations first need to improve their ability to deliver projects using their information systems (Avison & Fitzgerald, 2002). The complexity of IS determines how the IS can capture all the businesses of the organization and produce quantitative and qualitative results to the organization. The study will focus much on factors that go in measuring and determining the complexity of an organizational IS to enable organizations and IS providers to understand the effect of complexity on the business of the organization. This thesis examines in details some already existing methods of measuring complexity of IS. It also takes a practical approach of measuring the complexity of IS of a company.

The findings of this study will help understand a practical method for measuring the complexity of IS. Additionally, this work will reveal how multinational companies build an active, yet complex IS to factor their entire business and sub-businesses into such systems to strategically win a large market size.

1.6 Organization of the Study

This work is based on point by point analysis of various methods of estimating information system complexity. This work is divided into five parts. The first part gives a general Introduction to the project. It goes further to explain the objective and the significance of this project. The second part focuses more on information systems and their interrelationships with business, a significant emphasis on the concepts of simplicity and complexity.

The third part discusses the complexity of information systems. It presents an in-depth review of some methods of accessing and measuring information systems complexity. The fourth part reveals how the complexity of Foxconn can be analyzed and measured using the Function Point method discussed in chapter three. The last section covers a summary of the essential techniques and findings, conclusions and recommendations in relation to the complexity of information systems.

2. LITERATURE REVIEW

ISs are now ubiquitous in nearly all large companies and organizations (Stair et al, 2010). They provide a permanently available online transaction and information to customers. They automate an ever-growing share of business processes and tasks, thus contributing to the rationalization effort and cost reduction required by the globalization of competition. Managers and higher level executives use IS to perform business activity monitoring that allows them to react quickly in fast- moving markets, where reducing the time to market is more important than ever. ISs have thus truly become an essential tool for sound decision-making as well as for selling or providing goods and services (Stair & Reynolds, 2006).

2.1 Defining Data and Information

It is essential to strike a difference between data and information when structuring business IS. Data as commonly described is an unprocessed fact and can take different form and shape, such as a date or quantity. It is necessary for businesses to put in place appropriate procedures to ensure data are recorded (Hardcastle, 2008).

A standard definition of information is that it is data that have been processed so that they are meaningful (Hardcastle, 2008). This invokes a process that is used to produce information which involves gathering data and processing the collected data to create the report.

As stated above information is generated through the transformation of data. This can be accomplished using many different methods. Some examples of how data can be process are by using the aggregation method which summarizes data by taking averages of group of numbers. Classification, another way of processing data places data into categories such as on-time and late receipts. Sorting organizes data so that items are placed in a particular order, for example, listing receipts by delivery date or delivery time. Calculations can be made on data such as calculating an employee’s bonus by multiplying the number of hours worked in excess by the hourly bonus rate of the company. Lastly, data can be selected based on other selection criteria, such as the geographical location of customers. Although it can be concluded that information is a useful resource for individuals and organizations, not all data can be considered valuable. The differences between good and bad information can be identified by considering whether or not it has some or all of the characteristics of information quality. Characteristics can be related to the timing, content, and form of the information (Hardcastle, 2008).

Timeliness refers to the availability of information when needed. If the information is provided to the organization too early, it may not be useful. Likewise, when the information is supplied too late, it will be of no use with the business trend. Information should, therefore, cover the correct period. For example, a production might include information concerning Material master, capacity requirement, current performance and predicted performance so that the production Controller and scheduler have a fair idea of all requirement and their availability before real-time work is started of the order. The content of the information points to the accuracy of the information in relation to what the intended use of such information is. Information is only relevant when it has direct accuracy with its intended use and the circumstances of the user. Again, clarity of the information with regards to its recipients can be referred to as the form of the information. The user of the information should be able to identify and locate key variables easily and quickly. These are general characteristics of good information. One can say that good information has high level of detail.to meet the recipient's information needs.

2.2 Defining Systems

According to Bertallanffy (1977), a system can be defined as a collection of components that work together to achieve a set goal or a common purpose. A system aims at receiving inputs and transforming these into outputs (Bertallanffy, 1977). In the earlier section where I defined data and information, the use of the conversion process was used to explain how data is transformed into information. It must also be emphasized that not every system has a single goal. Some systems may have several subsystems with subjective sub goals, all contributing to meeting the overall system goal (Bourgeois, 2014). For example, the Marketing, Operations and Human Resource Managers of an organization should all have goals geared towards achieving the overall organizational objectives. It is evident that in systems, data are used as the input for a process that creates information as an output to monitor the performance of the system, some feedback mechanism is required (Chenhall, 2003). Also, arrangements must be put in place to correct any errors that occur and ensure that the system is fulfilling its goal.

2.3 Defining Information Systems

IS serves as a platform for management to access and make decisions to ensure that the organization is controlled (Jorgensen & Kjetil, 2006). The organization will be in control if it meets the needs of the environment. Many forms of information systems are used to support business processes in a firm. As a business may have tens or even hundreds of different business processes, they will require different types of software and support systems. Sometimes business ISs are particular to one area or group of people. At other times they may provide services to the whole organization. One thing that organizations are sure of is that no single system can give all of the information that an organization needs (Wood, 1986). Business information systems can be viewed from a functional perspective whereby the business function defines them; or from a constituency perspective whereby the system is defined in terms of the organizational groups that it serves.

2.4 Business Information Systems

With the earlier explanation of a system and information, we can put these two together and deduce that business information system is a group of interrelated components that work together to carry out business processes and control actions in order to transform raw data into meaningful use to support planning, forecasting, monitoring, co-ordination, decision making and operational activities in an organization (Hardcastle, 2008). Regarding the components that perform this activity, they can be classified into five important resources which are People, hardware, software, Process, and Technology. People as a resource include the developers and users of an organization IS who help operate and maintain the system. These people may include the technical support staff and managers of an organization (Brehm & Lynne, 2000). Hardware resources may include the physical and tangible items such as printers and computers. Software resources refer to the entire computer based program and associated instruction manuals that helps manipulate the hardware. Technology resources refer to all the systems and tools required to effectively produce or create a product or service (Bocij, Greasley & Hickie, 2008).

In most organizations, Business Information Systems (BIS) make widespread use of information technology, such as personal computers. The reasons why computerized BIS have become common among many organizations these days are as a result of their advantages to business. Such benefits include speed, accuracy and reliability. These BIS have high degree of flexibility due to their ability to be programmed to perform different task. There are, however some demerit that come along with computerized BIS. They lack creativity that human possess and it makes it difficult to incorporate decision making such as innovation and intuition (Arteta, 2004).

2.4.1 Types of Business Information Systems

Information systems may be categorized into two groups. They are the systems that support an organizations day to day operation of the business and systems that support managerial decision making (Hardcastle, 2008). Operation information systems (OIS) are generally concerned with process control, transaction processing and communications which happen within the organization on daily basis. However, Management information systems (MIS) aim at supporting management indecision making. They have a primary goal of providing support to managerial decision making (Boehm, 1981). Although these systems are useful for managers, reviewing the type of BIS and OIS that an organization adopts will reflect the reality of the entire systems within the organization, particularly with the current increase in inter –organizational e-commerce and electronic data interchange (EDI). For example e-business systems and enterprise resource planning systems cut across both operational and management systems to provide businesses with more integrated information systems.

2.4.2 Components of Business Information Systems

As stated earlier, many users have come to the intuition that an IS has a direct relationship with databases and spreadsheets. Others link IS to computers and e-commence. They are all right, at least in part: ISs are made up of various individual units that work collectively to provide value to an organization. These components have their role to play in achieving the broad organizational goal (Bourgeois, 2014). The broad components of every IS includes;

- **Technology:** Technology is the application of knowledge scientifically for practical purposes. The daily improvement in the ways of doing things better, technology is a part of our lives in so many ways. As discussed earlier, the first three components of an IS: hardware, software, and data all fall under the category of technology (Bourgeois, 2014). Thus, technology is an inevitable part of every IS.
- **Software:** Software is a set of instructions that tells the hardware what to do (Bourgeois, 2014). When programmers create software programs, they do so by merely typing out lists of instructions that tell the hardware what to do. These instructions may be business processes that the software will have to emulate and also task that need to be solved. There are several kinds of software, with the two main groups being operating-system software, which makes the device usable, and application software, which does something useful (Bourgeois, 2014). Example, Calculating employees monthly salary.

- **People:** Information systems just like any other computer and information technology has behind its operation the key value of people. Technology unlike human being needs human assistance to function effectively. It is therefore important to factor human or people as a key component of every information system. From the front-line help-desk worker to the system analyst, programmers, all the way up to the chief information officer, the people involved with ISs are an essential element that must not be overlooked (Bourgeois, 2014).
- **Networking:** Besides the components of hardware, software, and People, which for ages have been seen as the core technological components of every IS, it has been suggested that one other element that should be added when talking about the technological components of an information system is communication (Bourgeois, 2014). Even though the first kind of computers were stand-alone machines that did not access the internet and didn't communicate with other computers or devices over any network (Bourgeois, 2014). However, in today's hyper-connected technological world where business communicates each day with various partners across the globe, it is an extremely rare that computer does not connect to another device or to a network. Technically, the networking components made up of hardware and software but it is such a core feature of today's information system.
- **Process:** The final component under discussion of the components of IS is Process. A process is a series of activities followed to achieve a desired outcome or result (Bourgeois, 2014). Information systems have become very tied to organizational processes, bringing more productivity and better control to those processes. But simply programming IS to be in tune with businesses processes which in other words is referred to as the automation of IS is not enough. Businesses' looking forward to derived maximum benefits from IS are doing more than just automation (Bourgeois, 2014). Technology buzzwords such as "business process re-engineering, and "enterprise resource planning" all have to do with progressive measures geared towards getting the maximum from IS and its integration with technology. Businesses hoping to excel and be the market leader over their competitors are highly focused on the components of IS (Bourgeois, 2014).

2.4.3 The Architecture of Information Systems

We define the architecture of a system as a set of related models that describe the essentials of a system (Stair et al., 2010). The variety of models describes different components and different aspects of the system. Elements are building blocks: a system can be constructed by gluing together the components according to some rules. Opinions differ from components in the sense that they do not occur as a system on their own (Cardoso, et. al, 2006). To illustrate this, let’s consider the building industry. Each floor of a building can be seen as a different component and the water supply system, or the electricity system is examples of views. The latter systems are realized in parts that belong to elements. Usually, we distinguish several standard aspects of a system: a business view, a functional view, and a technical view. Each of these views can be split into parts. The professional view is divided into a software view and a network view. So we divide the architecture of an information system into four levels (Chaffey & White, 2010):

- Business architecture: Business processes and the object classes that play a role considered from the perspective of the IS.
- Functional architecture: The logical decomposition of the system into (logical) components and the assignment of processes and object classes to these components.
- Software architecture: Software components that realize the functional architecture, e.g. the database management system, the workflow engine and the connectivity software (middleware).
- Network architecture: A computer and communications network together with their operating systems.

In case of a system that is implemented on a stand-alone computer, the network architecture is trivial: just one node (Chaffey & White, 2010). In some other lifecycle models, business architecture is considered to be part of the requirements analysis.

Architecture should have two essential properties: consistency and completeness. "Consistency" means that all models are steady internally and that they do not imply any contradiction or conflict when they are put together (Chaffey & White, 2010). Internal consistency should be defined regarding the modeling framework. One simple but important consistency property is that a model is syntactically correct.

With "completeness" it means that all models together provide sufficient information for constructing a system with the same functionality as the modeled system. A practical test for completeness is that from the set of models the external behavior of the system is fully defined and that a simulation model of the system can be generated and tested in the environment where the real system should operate (Chaffey & White, 2010):

2.5 Developing Information Systems

Developing a new system or application can represent a major investment of time and money for an organization (Cong & Romero, 2013). Often, such systems are driven by business process reengineering to achieve improvements in cost, time, service, and or quality. These projects require careful planning and clear communication. Development frameworks for new software have moved away from the waterfall framework towards agile development. Software development lifecycles can be considered as an abstract representation for the process of creating software and provide an overall strategy for development and project planning (Kock, 1996).

2.5.1 The Traditional Waterfall Model

The waterfall model defines the various activities that should occur when building BIS. These activities usually occur in a predefined order with a review at the end of each stage before the next step can be started (Martin, 2002). The purpose of the waterfall model of BIS development is to divide the development process up into a series of manageable parts that relate to each other in an organized way. Also, some tasks will have to be completed before others can commence. For example, it will not be possible for a programmer to start writing a program until the design specification for that program is complete. The waterfall model is a simple representation of what actually happens during a systems development project, but it provides a useful framework for introducing information systems development since all of the activities that are identified in the model occur in a typical project (Martin, 2002).

2.5.2 Agile Software Development

Traditional sequential, method of software development “the waterfall model”, discussed above holds that complex systems can be develop in a single pass (Laurie, 2003). This eliminates revisiting requirements or design ideas in light of dynamic business or technology situations. It stems from the fact that complex software systems can be built in a consecutive, phase-wise means where all of the requirements are collected at the initial stages of the design, all the collected design is completed in the next stage, and finally the master design which will be implemented in a production environment is delivered (Gharajedaghi, 2011). It was first proposed in an article by Winston Royce in 1970, with its primary intention focus on government projects. This method therefore equates software development to the manufacturing assembly line; defined processes can be established that, when used chronologically, result in a successful project each time.

However, current trends of information development erase the notion by the advocates of the waterfall model. Almost, if not all information systems are not so simple and its development could be accurate from the beginning to the end (Martin, 2002). The inherent uncertainty and complexity in all current IS projects requires some adaptive development plan to cater for changes that may occur and other unknown variables. Agile makes up for the lapses within the traditional waterfall model. With agile, the development lifecycle is cut up into increments also referred to as “iterations” and each iteration focus on each of the traditional “phases” of development. With this principle, each face is seen as developing a usable unit to the customer which can be tested (Martin, 2002). This approach serves as a platform for improvements in each phase of the development. Complexity of each phase can be used to access the complexity of the total project since each phase is seen as a usable product to be delivered to the customer.

3. EVALUATION OF COMPLEXITY

IS development projects are becoming more and more complex each day. Project team leaders and managers are under intense pressure to deliver a finished complex IS in agreed timeframe and budget (Sterman, 2000). Often, this is not possible due to bad judgment about the needed timeframe and complexity of the individual component of the IS. Over the years organizations have invested a lot of money into developing IS, hoping this will better their operations and strategic position within the market, however, many organizations have failed in their intent (Sterman, 2000). To return the investment put into IT, organizations first need to improve their ability to analyze and understand the complex needs of their organization. The complexity of IS design stems from the fact that, besides a technological basis, it implies and covers business processes, and reflects on the final project goal, time, costs and quality. Furthermore, the design complexity reflects on managing human resources, such as a team, users and top management support (Speier, et. al., 2003).

Still, too little money and time are invested into measuring the complexity of IS development, and what is invested mainly applies to measure the complexity of the software (Wood, 1986). Many writers have written on the problem of measuring the complexity of an information system. Although organizations invest heavily in developing IS with the primary goal of making better their operational and strategic positions, however, many organizations fail to obtain this goal because of high failure rate of development projects. The Standish Group reported in 1994 that U.S companies spend more than $25billion each year during the 1990s for developing IS. Out of these companies there was only 16.2percent success rate. (Rubinstein, 2007). The 2001 reports showed that even though the investments had increased by four times compared to the 1990s yet only 28 percent of the projects were successful (Rubinstein, 2007). To enhance the return on investment in IT, organizations must first understand their IS needs, how businesses are interrelated and the complex needs of their operations (Jorgensen et. al., 2006). This will not only help build a well- structured IS but also improve and set a clear measure of the individual component that is required to be in the IS so that organizations can achieve their objectives .

Estimation in the development of IS product is a tool that helps in the decision-making process and ensures information needed for defining the various components and negotiation with the users. It is assumed that a larger quantity of different measuring elements (like in the function point method) results in a better estimation. However, processing a large quantity of input data is time-consuming and demanding, and sometimes a quick estimation of IS development is required.

The concept of complexity of IS as explained in (wood, 1986) states that in conceptualizing project complexity; there is the need to use complexity as a basis. Wood (1986) defined three major type of task complexity: He emphasizes on component complexity, coordinative complexity, and dynamic complexity. Component complexity of a project refers to the number of individual distinct functions of the task that needs to be performed in other to complete the task. He also explains coordinative complexity to refer to the nature of relationship that exist between task input and its resulting output i.e. the input to output relationship.

The strength of the relationship and the form that acts between information cues act and product are components of coordinative complexity. Finally dynamic complexity is caused by the changes in the states of the world which have an effect on the relationship between tasks and IS (Wood, 1986).

3.1 Defining Complexity

The complexity of a system is composed of different parts connected with each other in order to exhibit one or more behavior that is not clear from the properties of its individual parts. In other words, a complex system has various individual subsystems that come together to perform a specific function (Alamoudi & Kumar, 2017).

The interrelationship between the IS components enhances communication and creates a mutual understanding between businesses (Gruhn & Laue, 2006). Complexity creates new ways for the organization to communicate, automate, and reduce the business process management problems in globalized economies. Information Systems on their own do not have any value but they have a tremendous role in creating value by supporting business (Stair, et al, 2006).

3.2 Components of Information System Complexity

At the elemental level, all complex systems can be broken down into the three interacting groups of components i.e. peoples, tools or Technology and processes.

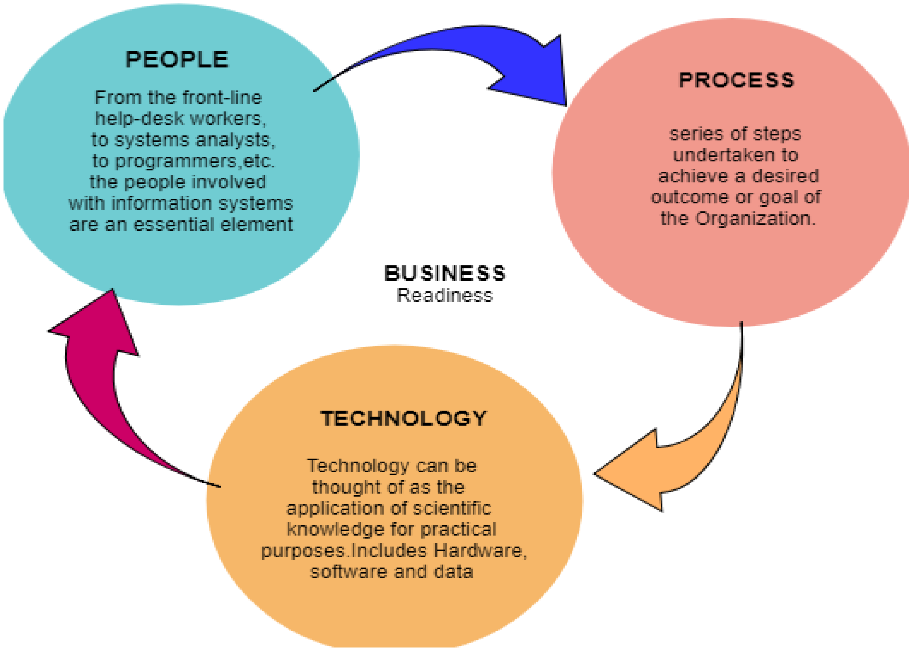

Figure 1: Basic Sources of Complexity

Source: Alamoudi D, et al. 2017

The broader components of every IS stem from these three components:

- **Processes:** The series of individual activities that a business performs each day in their quest to achieve their objectives (Alamoudi D, et al. 2017). These activities may include but not limited to issuing purchasing and sales orders, production processes, labor times calculated on a product, shipping of products, purchase requisitions, etc.
- **Technology:** This component of the information system is the application of scientific knowledge for practical purposes within the organization (Alamoudi D, et al. 2017). This includes hardware components such as computers, scanners, work-centers machines, etc. Technology also encompasses the software used to manipulate the hardware. Data of the organization is also a very important technological component of an information system. Organizations stores thousand and millions of data about its activities each year, this data may stem from customers data, suppliers records, future demand estimates, and environmental data such as legal requirements of the country in which the organization operates. This data is very important for the future prospect and sustainability of the organization.
- **People:** At the center of technology and processes is a third component, people. People are a very important component of every information system (Alamoudi D, et al. 2017).

From the front-desk attendant, the factory workers, the departmental manager, the accountant, and the driver all play an important part in the business information system.

Finally, the organization should be ready for the task and cost involved in implementing and maintaining the IS. Over the years, organizations have invested a lot of money in building and maintaining IS. Irrespective of the high failure rate of these IS to achieve their objectives, organizations should keep improving their IS each day to take advantage of the growing global market. Example of more complex interrelated system is shown in the next figure below:

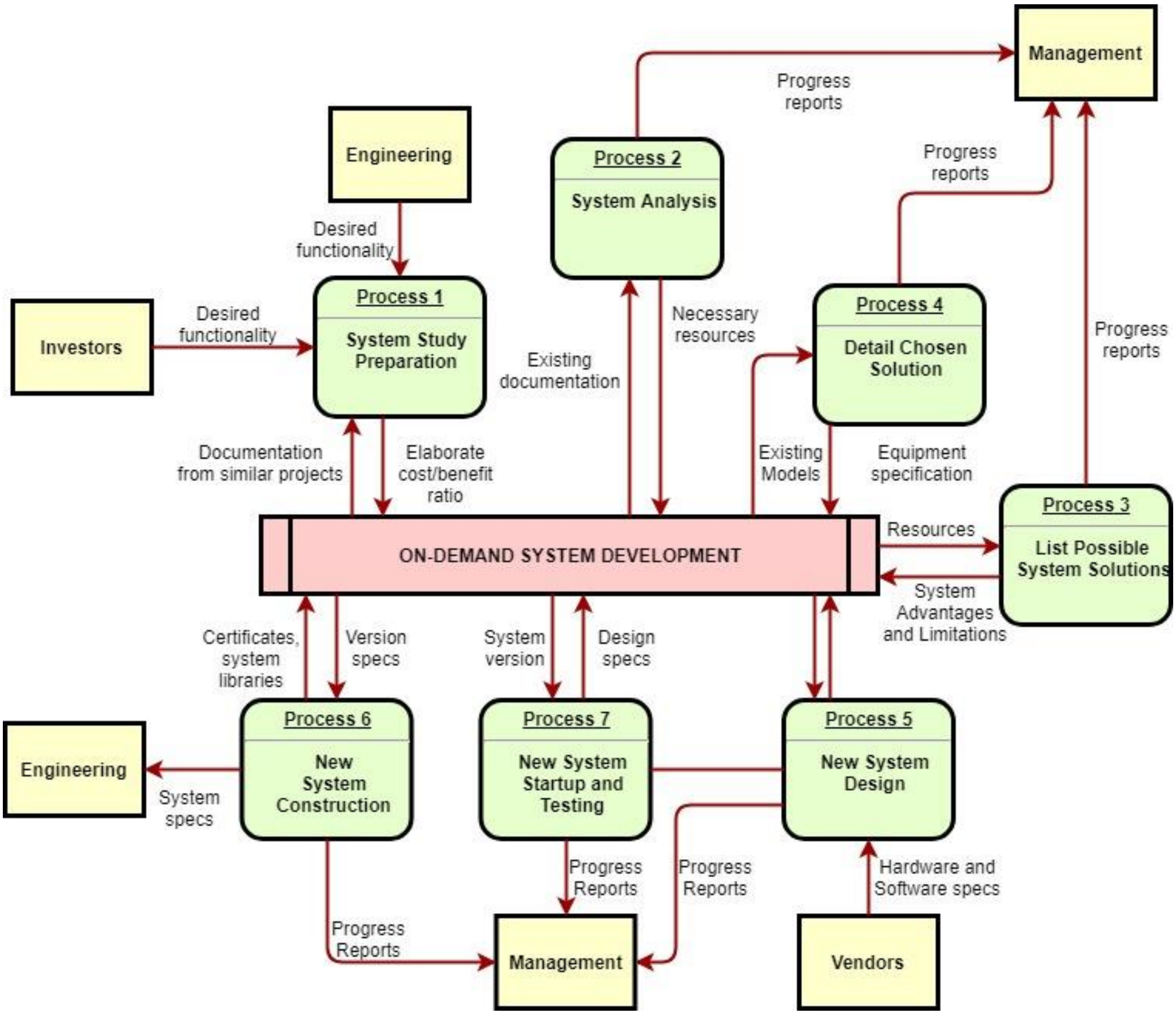

Figure 2: Interrelated system

Source: Internet (Draw Io templates, 2018)

3.3 Locus of Complexity: Organization versus Informational Technology

The locus of IS complexity defines the angle from which an organizations is measuring its complexity (Mukhopadhyay, et. al., 1995). Weather the complexity of an organization spins from the organizational factors or IT factors. Striking the difference between the two ends of measuring complexity is a very important basis to understanding the complex nature of an organization. This as explain by Baccarini (1996) in his book, he explains organizational complexity of an IS to include the structure of the organization, the processes involve in everyday business, information needs of the organization, user involvement, support of top management, and project personnel capabilities that goes into executing a project or task.

The IT components not only include the hard technology elements such as hardware, software and network, but also soft technology element such as the knowledge of the operators of the IT, their skills and experiences that are put in operation to complete a task (Qureshi, et. al., 2015). Since different project capabilities are required to deal with organizational complexity as opposed to IT complexity there is the need to strike these loci of complexity to understand the nature of individual task complexity.

3.4 Complexity versus Simplicity of Information Systems

Societal changes and Technology are dramatically changing the business environment and how businesses are been done. Changes to the ways of doing business have forced organizations to move from the simplicity of business information systems to complex and more integrated business information systems (Rao, et. al., 2002). Companies and organization have gone beyond their comfort zones to have a fair share of the global market. Each day, companies are forced to interact and discuss different business-related activities with supplies from other countries, buyers from other continents, and subsidiary companies from other regions. Companies are thus forced to have their information systems to take care of all their inter-business and intra-business relationships. From the top management, through to the production lines, every minute of work recorded by machine time to the payroll and cost centers, businesses need information from each of these activities. Companies will need to build different simplified IS at each stage of operations (Rao, et. al., 2002). To avoid the cost and segmentation of companies IS, companies have resulted in building a single IS to cater for all its business needs hence the need for complex business IS.

Complexity has become a very important tool that companies use to compete these days. From attaining high customer satisfaction to getting the best deal from their buyers to keeping an up-to-date record of each activity, to making maximum profit, all depends on the complexity of the company’s IS to capture every single business activity.

3.5 The Need for Estimation

It is essential for a project manager to know the effort needed, schedule and functionality of a project in advance. This is essential for the effective completion of every project. It is not just enough to know about the project without knowing the feasibility of the project. Factors such as the time needed to complete the project and resources needed are very useful to creating a marketable project (Kavoussanakis & Sloan, 2001). However, irrespective of the amount of information that one gathers before the start of the project, project factors may change during the execution of the project and the change can affect the delivery of the project. One effect of this change is managers' inability to predict the quantum of change that is to be expected hence the project manager will need to gather ample information before the start of the project. This is why we estimate information system projects complexity. It is always difficult to estimate in advance and get accurate values at the end, yet estimation techniques help managers to glace a view about these changes and anticipate for them (Kavoussanakis & Sloan, 2001). Better and quality estimation techniques yield more accurate results and serves as a guide for manages to yield the desired results and thus effective contingency plans. This is the answer to the project manager to the ever-changing conditions of the project. It is the norm for project estimates to be quite off the final figures and become better as the project progresses and the information becomes more solid. In the early stage of a project, feasibility study, one can underestimate the size of a project by up to 4 times its final size, or overestimate it by the same range (Kavoussanakis & Sloan, 2001). However, the numbers can drop to 25% by the time the design document is finalized. Estimation thus has become a powerful competing too for most organizations.

3.6 Methods of Estimating Complexity

If there is an objective to develop an IS, there is also a very logical request: how could one measure IS complexity (Kim, Lively & Simmons, 2006). IS complexity could be measured or estimated by different methods. An Essential factor in measuring or estimating process is also an IS type. IS type defines its relation toward inputs and outputs, interaction with the customer, customer interface, independent learning, and so on. In other words, IS type defines its placement in three-dimensional vector space by Genetic taxonomy. There are so many methods which could measure or estimate IS complexity, some of which include: Functional point analyses (FP), Data on Documents method (DOD), Constructive Cost Model (CoCoMo), and Database complexity method. Some methods could measure, but some of them could only estimate software complexity. Accepted referent method for measuring IS complexity is the method of functional points analyses (Morris, 2001). The other methods are in generally reclined, or they are derived on FP. Propriety of some methods is proved by correlation with FP (Kavoussanakis & Terry, 2001).

Methods for estimating software system development and its complexity can be generally put into two groups. These are Direct Estimation Methods and Derived Estimation Methods. Direct Estimation Methods are also known as Expert Opinion Methods. These imply the cooperation of one or more experts which directly estimate required elements of the estimation of function points, basing their estimation on experience and intuition. Derived Estimation Methods or Algorithmic Model Methods provide with the estimation of complexity as a function of more variables which relate to certain attributes of a software project. In this project, the focus will be placed on three of the well-known methods which different writers and organizations have applied in the past. They include the function point method, Database Complexity model, Data on Document Method and Constructive Cost Model.

3.7 Function Point Method

Function Points were introduced in 1977 by A. Albrecht of IBM to measure the size of computer applications and the projects that build them (Southard, 2000). The objective was to give, as a result, a number which will represent software complexity, and that number should be of importance (Kemerer, 1992). In other words, for two different software, a number of functional points could be given, and it could represent a real and objective difference between them (Southard, 2000). FP is a number without dimension defined in functional points that represents an effective relative measure of functional value delivered to the customer regarding the International Function Point Users Group (IFPUG); the prevalent group of users.

3.7.1 Using Function Point Analysis

As stated in the introduction, a very core reason why function point analysis exists and is used most frequently by many organizations is to address the issues of measuring and assessing productivity and costs related multi-tasked and multi-complex applications. As stated by Grupe (1991), users of the function point analysis aim at achieving one or more of the following:

- Forget the code running behind and focus on the application's functionality
- Apply good analysis to any Information system or application
- Apply analysis to any source code
- Calculate the cost of an IS using cost per function point
- Determine the number of function points in a project to estimate the complexity of a project
- Measure software firm or department's productivity.
- Estimate the complexity of the IS using the function points.

Therefore it can be said that function point is one of the techniques for providing analysis and a measure of the system's size and complexity which helps to determine the effort and cost required to develop and maintain a system (Symons, 1988).

3.7.2 Function Point Analysis

FPA is a methodology for measuring IS Complexity and the cost associated with development and maintenance. One FP is one end-user requested business function. The following defines the five characteristics of function points as stated by Grupe (1991):

- **External Input (EI)**: These are the actions that are executed by the end –user during the operations of the information system. Examples include clicking the mouse.
- **External Outputs (EO)**: This represents the results the end-user obtains due to the input passing through a process. Examples include the GUI displayed after a mouse click on a button or reports that are displayed on the screen
- **Internal Logical Files (ILF)**: These are the logical or sometimes referred to as the master files that the system operates with on daily basis or during its session.
- **External Interface Files (EIF)**: The external logical files, unlike the ILF which are used solely for the purpose of the application or the system, the EIF are shared among different applications.
- **External Inquiries (EQ)**: These are typical functions that are initiated by the end user to other systems outside their IS. Example includes a click for online help. Usually a developer creates these functions for the end-user assistance.

One all these characteristics and components are identified and gathered, a complexity of the functional value for the end–user is determined. Atin et al. (2018) in his work list 14 factors that determine the function point of an IS. These factors as listed in the table below are used in assigning weights when using the function point method.

Table 1: Factors in FPA

1	The level of data communication complexity
2	The level of complexity of data processing
3	Level of performance complexity
4	The level of configuration complexity
5	Software user frequency level
6	Data input frequency level
7	Level of ease of use for the user
8	Data frequency update rate
9	The level of complexity of data processing
10	Level of possible reuse / reusable program code
11	Level of ease in installation
12	Level of ease of operational software (backup, recovery, etc.)
13	The software level is made for multiple organizations /companies /clients
14	Level of complexity in following changes /flexible

Source: Southard, 2008

The next operation to be done after function points are assigned to the various factors, the function points are added and assumed unadjusted. The specialist then will assign a weighted value from the scale of zero (minimum) to five (maximum) on each function Points according to the degree each component has of the entire system (Grupe, 1991). During this stage, the analyst assigns degree of influence to each function point, inputs and inquiries are rated lowest on a zero to five scales, while interacting with files or the databases have the highest weights assigned (Banker et al, 1993).

3.7.3 Using Function Point Values

The power of using the FP is for one organization to compare one project to another, or to compare one application to another, and finally to compare an organization performance to that of another (Grupe, 1991). Function points allow this comparison by striking differences among project size from the effort and the technology required. On the other hand, if the need is geared to estimate the development time, programmer work output, process improvement or minimizing defects in software, Size of the project not to be affected by its technology alone (smith, 1997). Rates used for comparison under the function point as stated by smith (1997). The rates used in these comparisons include:

- Delivery Rate - Project function point per work effort (FP/Hours)
- Support Rate - Work effort that is ben supported per application function point (Hours/FP)
- Cost Rate – The cost that is incurred per function point (CZK/FP)
- Defect Density - Defects per application function point (Defect/FP).

Therefore there are two measures mostly obtained from measuring complexity using the function point method. They include project development time and the project cost (Smith, 1997). Other methods can be used along the function point method. FP values are normalized across applications, supports, projects and quality. The decision to either purchase an IS off the shelf or outsource it can be made using the FP method (Smith, 1997).

In measuring the cost of developing a software project , time is of great importance to businesses and software development companies which internalizes IS projects to develop business applications. If projects are not the accurate size during the requirements gathering phase of the software development life cycle, businesses will lose out to those who do it more efficiently.

3.7.4 Function Point Analysis Caveats

Function point analyses as already mentioned is one of the methods for estimating software. It helps to measure the cost and development time of software application. In the next section, other methods for estimating or assessing the complexity of an information system will be discussed. Notwithstanding the popularity and wider application of the function point method, the list below provides additional assumptions when using the function point analysis.

- Function point analysis gives a consistent approach for measuring the size of a software development project during the initial stages of the software development cycle.
- The complexity of the system increases with the growth in the number of function points and this translate to more time and cost in the developing of the information system.
- Miscalculating data items, reports and perimeters, which serves as inputs to the function point unknowingly can distort the accuracy of the function point analysis which can renders it useless.

- The FPA is not an alternative to experience.
- Experienced users of function point analysis have a great responsibility to train new users in usage and accuracy.
- Non-business applications that use tons of logical functions than business-type applications are not a good fit for FPA estimation.

Considering these assumptions and caveats, the next section implores the use of alternatives methods to the function point.

3.8 Data on Document Method

Data on Document (DOD) is a method which could be used for IS designing complexity estimation (Georgescu et al., 2005). In this method, all system documents have to be collected, and then continue the computation of relevant data on each document. Summation of different data types gives a number which represents a system complexity. This method statistically correlates with referent measuring method FP. DOD could very fast accomplish system complexity estimation, and foresee measuring results by using FP method.

3.8.1 Design Complexity Based on Quantity of Data on Documents

The method proposed in this section is primarily assigned to the estimation of IS design complexity. The method concentrates on measuring the quantity of data on documents and is accordingly called the DOD method. During the method definition process, the main aim is to simplify the estimation and, by doing that, speed up the process of estimation. The point was to obtain certain indicators, as simply as possible, which could be compared to function points, since the FP method is the most widespread and the most accepted method (Pavlic et al., 2008).

3.8.2 Estimation Elements in Data on Document Method

In the process of defining estimation criteria, a lot of thought is given to the function of an IS, i.e. answering the question of what all information systems have in common. The purpose of every system, including IS, is to result in needed information based on input data and its processing. Years of experience in developing IS resulted in understanding that the number of documents and the complexity of a given document inside a business system somehow determine the complexity, not only of designing but also developing an information system (Pavlic et al., 2008). In most cases, to fully understand them, more time and effort is needed for systems with numerous documents, than for those with fewer documents. Proportionally, the project development is more complex and time-consuming (Pavlic et al., 2008).

When documents are concerned, we can divide these into two groups: basic (original) documents, and classic reports documents (derived, calculable, summary, preview documents).DOD method is based solely on measuring one element, and that is the number of data on basic documents. Basic documents are those which connect the environment to the system or the sole processes in the system. Classic report documents mostly contain data which belong to some other document or are calculated from previously familiar data (Pavlic et al., 2008). Some basic documents in business systems can be put into this category, based on the preceding report description. These are mostly documents which serve as an output to the environment (like an insurance policy created out of the insurance offer, a note of receipt created out of supplier's note of delivery, a receipt created out of the order, etc.). The data on mentioned documents are copied without any changes or with little change. For example, a decreased quantity on the note of receipt, document date, added data from a coding table, etc. according to the proposed method, such documents are not considered classic report documents and they enter the analysis for the complexity estimation by DOD method. Unlike such outputs from the IS, there are so called classic reports which are just a view to the condition of data in the data base, and their function is to inform or manage a business system (various statistic reports). These do not belong to the group of basic documents of a business organization (Pavlic et al., 2008). It is difficult to say how many reports like these are in a business system since there are continuous inquires for new reports with the same or different data, according to the same or different criteria. Therefore, classic reports are not considered in the proposed method and are not counted. The DOD method does not count or analyze documents like various regulations and laws which determine system operations. The basis of the method is data and processes over this data do not interest us in this method. All processes represent a "black box" (Poščić et al. 2008). What are important are process inputs and outputs, and the algorithm itself is ignored. The sole number of documents in a system cannot be sufficient criteria for the estimation of design complexity because these cannot significantly differ. Each document needs to be analyzed, but not in detail like in the process of its modeling, but it is necessary to count a different kind of data on the document. Therefore, according to the DOD method, there are two criteria for the estimation of IS development complexity: document and the number of data on the document (Poščić et al. 2008). Since the aim of the proposed DOD method is to estimate the complexity of developing business information systems in a quicker and simpler manner, weight values are not assigned to criteria, but every observed data on the document has the same weight factor. It is hard to determine the weight of a given data at the very start of the project, i.e. its influence on further complexity of the project. We could possibly consider the importance of a certain document for the system, and based on that, assign a weight to a certain document (Poščić et al, 2008). The same applies to data on documents. Assigning weight factors to documents and data on them would have a few drawbacks. The quantity of documents circulating in a business system determines the complexity of the design, and the building of the IS itself (Poščić et al, 2008). A Model of Data on Documents method (DOD) proposed by Neven Vrcek in (Vrcek, 2008) as illustrated in figure 3 above gives a detail about the DOD method.

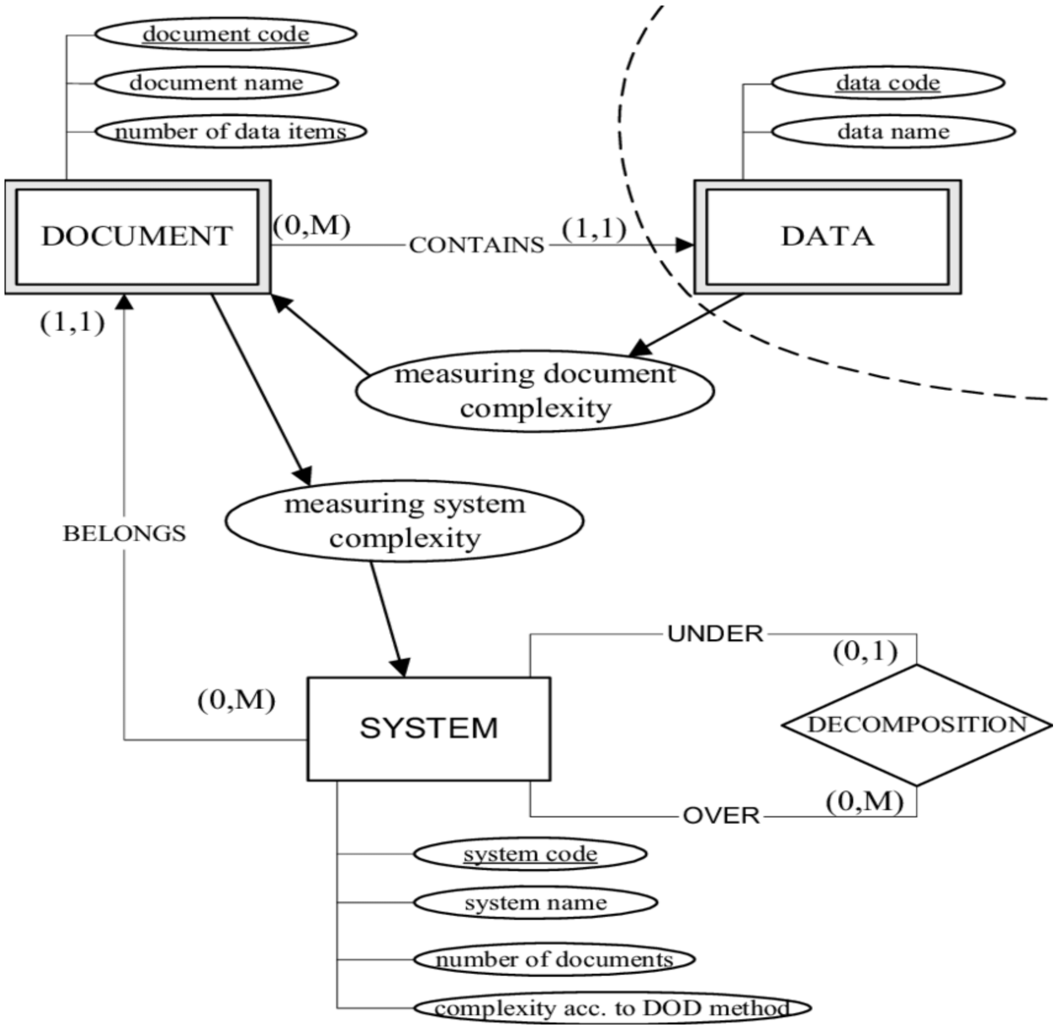

Figure 3: Model of Data on Documents method

Source: (Neven Vrcek, 2008)

In the center of the DOD method in the model shown above in figure 3, are document and data, and in the model, they are shown by entity types Document and Data. For each document, its code and name are recorded. The Document is shown as a weak entity type in relation to System, which means that a certain document “interests” us as a part of a given system, i.e. that one document belongs to the given system. A document can hold more data items, which is shown by relationship cardinality contains (0, M). Data is a weak entity type in relation to Document. Each data has its code and name. One data item can belong to exactly one document cardinality (1, 1). The cardinality should be explained in more detail: when we observe a data, e.g. a person’s name and surname, it appears on more documents in a given system. Therefore, we could have cardinality (1, M) at entity type data, so the relationship contains would become an aggregation as shown in figure 4 below.

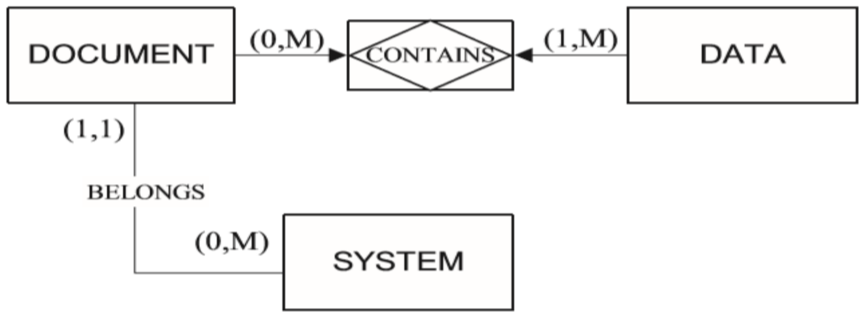

Figure 4: The relationship of entity type documents

Source: Vrcek, 2008

For each data on a document we would be checking whether it is an entity type or an attribute of an entity type. Such an analysis would demand much more time and would make the method more complex, and that is not the aim. Estimation of system complexity using the DOD method is conducted before the system modeling stage, according to the life-cycle of the information system development. Therefore, each data on a document is separately counted in the DOD method and belongs only to the given document. The accepted DOD model is shown in Figure 4. Besides documents and data, the model shows another entity type: System. The estimation of complexity is conducted for a system that has its code and name. Systems are, if they are complex, decomposed into smaller subsystems; this is shown by feedback relationship decomposition in Figure 4. Each system can be a subsystem of a given system, and, on the other hand, can be a super system of a larger number of subsystems.

For each system, the total number of documents flowing through that system is also recorded. After applying the DOD method, system complexity is determined.

3.8.3 Stages of Data on Document Method Application

The application of DOD method according to Pavlic et al. (2008) follows these stages:

- Decomposition of the system into subsystems
- Data collecting
- Creating a list of documents
- Determining the number of documents
- Linking documents and the system
- Counting data on each document
- Creating a data list
- Data processing

3.9 Constructive Cost Model

Constructive cost model (CoCoMo) has become one of the more common software estimation techniques in the software industry (Boehm, 1981). CoCoMo, developed by Boehm in the early 80s, comes in three flavors: basic, intermediate, and detailed. The CoCoMo models are directed more toward effort estimation than size and complexity estimation. However, the models do use complexity estimation results to obtain the effort estimations. Most companies implement either the intermediate or detailed model. Case studies have shown the basic models provide inaccurate results because the estimated costs and complexity are based solely on the size of the software (Boehm, 1981). The intermediate model estimates cost and complexity based on other drivers besides size. These drivers include reliability, complexity, execution time, storage constraints, etc. The detailed CoCoMo model removes the limitations placed on the basic and intermediate models. Complexity is considered to be a major driver of cost, reliability, and functionality of systems. Both the inherent complexity of the problem and any additional complexity of the implementation are important aspects. Software size is included in this interpretation of complexity because the complexity of software usually increases with the size of the software. These limitations include estimated distribution of effort by project phase may be inaccurate and the cumbersome calculations for larger software projects with many components. The detailed model provides phase sensitive effort multipliers for each cost driver attribute. These cost drivers, as mentioned with the intermediate model are shown in the Table 2 below.

The 15 attributes each gets rates on a six-point scale ranging from “extra high” to “very low” (in order of importance). The rating from the table below applies to the effort multiplier. The effort multiplier produces and effort adjustment factor (EAF). The values for EAF usually range from 0.9 to 1.4.

Table 2: CoCoMo Cost Drivers

Cost Drivers	**Ratings**					
	Very Low	**Low**	**Nominal**	**High**	**Very High**	**Extra High**
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of the application database		0.94	1.00	01.8	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	01.11	1.30	1.66
Memory constraints			1.00	01.6	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	01.7	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	01.10	1.00	0.90		
Programming language experience	1.14	01.7	1.00	0.95		
Project attributes						
Application of software engineering methods	1.24	01.10	1.00	0.91	0.82	
Use of software tools	1.24	01.10	1.00	0.91	0.83	
Required development schedule	1.23	01.8	1.00	01.4	01.10	

Source: Carol Poole, 2015

The Intermediate CoCoMo formula now takes the form:

$$E = a_i (KLoC)^{(bi)} (EAF) \quad (1)$$

From the equation 1 above, E is the effort applied in person-months; **KLoC** is the estimated number of thousands of delivered lines of code for the project. **EAF** is the factor calculated above. The coefficient $\mathbf{a_i}$ and the exponent $\mathbf{b_i}$ can be seen in the next table below.

Table 3: Project Modes

Software Projects	$\mathbf{a_i}$	$\mathbf{b_i}$
Organic	3.2	1.05
Semi-detached	3.0	1.12
Embedded	2.8	1.20

Source: Poole, 2015

CoCoMo can be measured based on three modes of software projects as defined by Carol Poole (2015)

- Organic CoCoMo: This is relatively small teams operating in a highly familiar Environment

$$\mathbf{Effort = 2.4 \text{ x } kSLOC^{1.05}} \quad (2)$$

- Embedded CoCoMo: The team has to operate within strongly coupled, complex, hardware, software and operational procedures such as air traffic Control:

$$\mathbf{Effort = 3.6 \text{ x } kSLOC^{1.20}} \quad (3)$$

- Semidetached CoCoMo: This is an intermediate stage between the two extremes of Embedded and Organic:

$$\mathbf{Effort = 3.0 \text{ x } kSLOC^{1.12}} \quad (4)$$

COCOMO models basically depend on the two main equations (equation 2 & 3) as discussed above.

- Development effort: In development effort, one month of effort by one person is man-month /person month/ staff-month. In COCOMO, there are 152 hours per person month. These values can change from the standard from 10% to 20% depending on the organization.
- Effort and development time: The higher the value of effort requires, the longer the development time and hence the more complex the project or information system. Again, the lower the effort required, the lower the development time and hence the less complex the IS.

3.10 Database Complexity Method

Database Complexity (DC) method can be used for database complexity measuring. Every business IS is composed of database and software. DC doesn’t measure IS complexity, but it only estimates it (Pavlic, 2008). When measuring IS complexity, all its composed elements have to be measured. IS is composed of hardware components, software, organizational ware, lifeware, and NetWare. DC measures only logical structure of physical database used in the IS. Size of database itself will not be performed in the measuring process. DC can, from a data point of view, estimate software complexity by measuring database complexity, DC method can foresee IS complexity. Physical database development is only one step within IS designing and developing methods. Before database construction, there have to be made at least two different data models which represent some data relations (Stair et al., 2010). These models are the entity-relationship model and the relational model. Entity Relationship represents appearance types on a semantic level, with their properties (attributes), and each other connections (Stair et al., 2010). The Relational model could be made by using some defined translation rules from the entity-Relationship model. Relational model, beside attributes, also represent keys and foreign keys in each relation. Mentioned models will ensure as less (minimum) redundancy as possible in future database. If the relationship has some redundant attributes, there are anomalies also present. Anomalies could be occurred in the insert, update and delete processes with tuples. If IS has database modeled and normalized to minimally 3NF, DC method could easily estimate an IS complexity. Un-normalized relations in the database generally will not be measured by DC (Stair et al., 2010).

3.11 Why Function Point as the Chosen Method

Throughout this chapter, a number of methods used for estimating information system complexity have been discussed. Some of these methods as already discussed in chapter two focuses on estimating the complexity of the software which is a major part of every information system. Most of these methods are derived from the function point method. Function Point Analysis has been proven over the years as a reliable method for measuring the size of IS complexity. In addition to measuring the output of IS, FPA is extremely useful in estimating projects size, managing the change of scope of a project, measuring productivity and individual output, and communicating functional requirement (Roger, 2000).

Our major goal in the next chapter will thus focus on using the function point method in analyzing the complexity of the IS of a manufacturing company - Foxconn.

4. COMPLEXITY OF ORION USING FUNCTION POINT METHOD

It's already discussed in the chapters 3 above, various features of software projects. It has been stated that software can be measured base on the size, complexity, adherence to process, quality, reliability and how profitable the software is. Most of these characteristics depend on the other. For example, the size of software can be a basis for measuring the complexity of the software. Also, the quality of an IS can determine the profitability of the system.

This chapter focuses much on how we can assess the size and complexity of the IS of Foxconn using the Function Point method. This estimation will be used to determine the size, the effort needed and the finally the complexity measured per the resulting estimation. Opinions of programmers and developers within the software department of the company are mostly used in determining the Metrix and rating of essential components to be measured. Expert's opinion as already measured forms a very important basis for assigning values to individual components of an IS.

4.1 The Business of Foxconn

Started in the year 2000 in the region of Pardubice in the Czech Republic, Foxconn has been applauded for being the 5th largest employer in the year 2018 within the Czech Republic. The company originally originated from its parent company Hon Hai Precision Industry Co. Ltd., trades as Foxconn Technology Group. Foxconn is a Taiwanese multinational manufacturing company with headquarters in Tucheng, New Taipei, Taiwan, which produces electronics. In Europe alone, Foxconn has factories in Slovakia, Turkey, Hungary and the Czech Republic.

It can be deduced from the little history that the company has other plans operating in various countries. Again all these plants intercommunicate with each other, with a large pool of suppliers.

4.2 The Architecture of Foxconn Information System

The information system of Foxconn just like any other company is a useful way by which the company speed and automate business processes and make decisions. In today's dynamic business world, companies are forced to interrupt with subsidiaries and other organizations that they trade with (Eccles, 2011). These may include suppliers, customers, government agencies, etc. The IS architecture as provided by the project manager of the IS as shown in the figure below

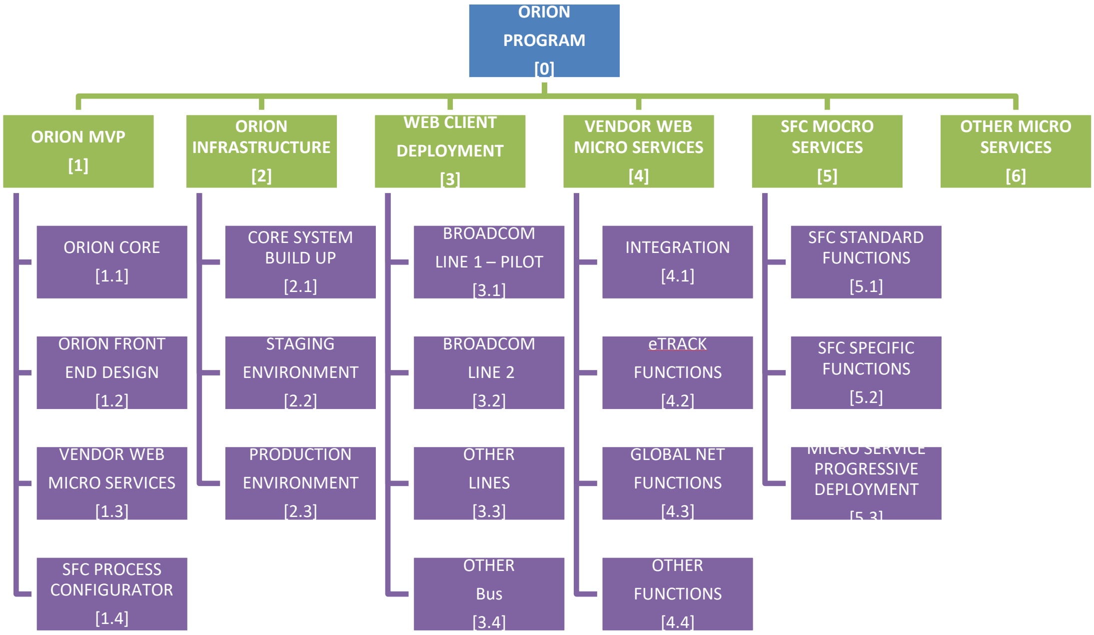

Figure 4: Architecture of Orion

Source: Foxconn Information system Documents, 2019

The layout showcases some fundamental components that most IS will have as part of its implementation components. Brief explanations of the various components of the architecture as explained below will help us to understand the various ratings that were assigned by the senior developers of the company.

In decomposing the IS in Figure 4 above, we have the following components which are fundamental and will be explained for the purposes of better ratings.

- **Orion *(*MVP*)*** – This primary component is focused on developing early and enough characteristics to satisfy early adopters. Finally, the full set of the features is designed and developed after feedback from the initial users of the product has been considered. With major subcomponents as shown in Figure 4 above.

- **The core** - The core of the information system manages document control, corrective action, training records, calibration, and most ISO 9001 required processes. The traditional approach to running software was to install the application on your local computer or company server and access it through your local company network.
- **The front end design** – This focuses on the User interface or that part of a IS that a user sees on the display, and acts on to enter commands or to access other parts of the IS.
- **Vendor- web microservice** - This component focuses on the core of the business of the company. This is a section of the information system that allows the company to interact with its customers and supplies. As the name implies the vendor or suppliers of the company are able to create sales orders and also interact with the company through this component of the System.
- **SFC process configurator** - Sequential function chart (SFC) is a language for programmable logic controllers (PLCs). It is one among the five languages defined by the International Electrotechnical Commission standard. It can be used to program processes that can be split into steps. This is essential for the purposes of splitting business processes in the IS.

THE CORE INFRASTRUCTURE

- **Staging Environment** – This is application environments for testing that exactly bear a resemblance to the production environment. It's a complete but independent copy of the production environment, including the component of database. Staging provides a true basis for QA testing because it precisely replicates what is in production.
- **Production Environment** - Developers usually refers the setting where software and other products are actually put into operation as the production. This environment can be thought of like a real-time setting where programs are run, hardware setups are installed and relied on for organization or commercial daily operations.

WEB-CLIENT DEPLOYMENT

This is the user side of the Web. It typically refers to the Web browser in the user's machine. It may also refer to plug-ins and supporting applications that enables the browser to support special services from the site. This may imply the whole user machine or refer to a handheld device that provides Web access.

VENDOR WEB MICROSERVICE

- **Integration**: System integration is the process of bringing together the component and sub-systems into one system. It is a combination of subsystems cooperating so that the system is able to deliver the predominant functionality and ensuring that the subsystems function together as a system.
- **E-track functions** - This is a component of the application that focuses on keeping tracks of all electronic communications with vendors and customers. Its primary aim is to keep data and files received from vendors.
- **Global net functions** - Global net is a component of the application that is a new buildup of the eTrack component. It expands the old interface of the eTrack application and extends its functionality to provide a platform or a section of the IS for the suppliers and buyers of the organization to use for their businesses.

With the summary and vital elements explained about the nature of the IS. The next paragraphs focus on using the function point to assess the complexity of the IS which for the purposes of this thesis we will refer to as ORION. In using the function points, we will follow these steps;

STEP 1

4.3 Calculating Crude Function Point

The first step in estimating project complexity using function point is to calculate Crude Function Point (CFP). There are several components involved in CFP calculations. These components have "simple", "Average/Medium" or "complex" categories depending on the characteristics of their complexity. Simple, medium and complex categories are derived from the complexity standards set by Foxconn. As for the standard complexity as set for the purposes of this thesis can be seen in the table below.

Table 4: Complexity standard of ORION

Complexity Level	Point FP
Simple	Point < 400
Medium	400 point < 700
Complexity	> 700

Source: Scale from company manuals

In addition, depending on the project complexity standard as in Table 4 above, Counting FP requires the identification of five types of functional components (XIA et al., 2009): Internal Logical Files (ILF), External Inputs (EI), External Interface Files (EIF), External Inquiries (EQ) and External Outputs (EO). Each functional component is classified as a complexity factor based on its associated file numbers such as Data Element Types (DET), File Types Referenced (FTR) and Record Element Types (RET) (W. XIA et al., 2009). The complexity matrix for the five components is shown in Table 5 and Table 6 illustrates the value of each component.

Table 5: Complexity matrix for FP function components

ILF/EIF	DET			**EI**	DET			**EO/EQ**	DET		
RET	1-19	20-50	51+	FTR	1-4	5-15	16+	FTR	1-5	6-19	20+
1	Low	Low	Avg	0-1	Low	Low	Avg	0-1	Low	Low	Avg
2-5	Low	Avg	High	2	Low	Avg	High	2-3	Low	Avg	High
6+	Avg	High	High	3+	Avg	High	High	4+	Avg	High	High

Source: XIA et al., 2009

Table 6: Function component complexity weight assignment

Component	Simple	Average	Complex
External Inputs (EI)	3	4	6
External Outputs(EO)	4	5	7
External Inquiries(EI)	3	4	6
Internal Logical Files (ILF)	7	10	15
External Interface Files(EIF)	5	7	10

Source: XIA et al., 2009

The Unadjusted Function Point (UFP) is calculated with Equation 1, where Wij is the complexity weights and Zij are the counts for each function component

$$UFP = \sum_{i=1}^{5} \sum_{j=1}^{3} zij \, . \, wij \qquad (5)$$

Table 7: Complexity Values for inputs

Explanation System	**Complexity Level**			**Total UFp**
Components	**Simple**	**Average**	**Complex**	
Inputs				
The Core	6 x 3 = 18			18
Vendor- Web Microservice	4 x 3 = 12			12
Sfc Process Configurator	5 x 3 = 15			15
Core System Build Up	7 x 3 = 21			21
Staging Environment			4 x 5 = 20	20
Production Environment			4 x 5 = 20	20
Broadcom Line		7 x 4 = 28		28
Broadcom Line 2		6 x 4 = 24		24
Other Lines		4 x 4=16		16
Other Bus		6 x 4 = 24		24
Integration	5 x 3 = 15			15
E Track Functions	6 x 3 = 18			18
Global Net Functions	5 x 3 = 15			15
Other Function	4 x 3 = 12			12
Sfc Standard Functions		6 x 4 = 24		24
Sfc Specific Functions		3 x 4= 12		12
MS Progressive Deployment	6 x 3 = 18			18
			Total	**312**

Source: function point values provided by Senior Software analyst of Foxconn

Table 8: Complexity values for Logical Files

Explanation System	**Complexity Level**			**Total UFp**
Components	**Simple**	**Average**	**Complex**	
Logical File				
Database File			3 x 15= 45	45
Entity Class		15x7=105		105
Controller Class			4x15=60	60
Interface Class			5x15=75	75
			Total	**285**

Source: function point values provided by Senior Software analyst of Foxconn

Table 9: Complexity values for outputs

Explanation System	**Complexity Level**			**Total UFp**
Components(Outputs)	**Simple**	**Average**	**Complex**	
The Front End Display	6 x 3 = 18			18
Output Of Customer Data	4 x 3 = 12			12
Output Supplier Data	4 x 3 = 12			12
Output EDI Data	5 x 3 = 15			15
Output Sap Data		4 x 4 = 16		16
Output Vendor Web Data		6 x 4 = 24		24
Output Materials Data	4 x 4 = 16			16
Output Processes Data	5 x 4 = 20			20
Output Of Hr. Data	4 x 4 = 16			16
E Track Functions		3 x 4 = 12		12
Global Net Functions		5 x 4 = 20		20
Other Function		8 x 4 = 32		32
			Total	**213**

Source: function point values provided by Software analyst of Foxconn

Table 10: Complexity values for Inquery

Explanation System	**Complexity Level**			**Total UFp**
Components (Inquery)	**Simple**	**Average**	**Complex**	
E Track Functions			4 x 6 = 24	24
Global Net Functions			6 x 6 = 36	36
Other Function			10 x 6 = 60	60
			Total	**120**

Source: function point values provided by Software analyst of Foxconn

Table 11: Complexity values for interface

Explanation System	**Complexity Level**			**Total UFp**
Components(Interface)	**Simple**	**Average**	**Complex**	
The Front End Design		10 x 7 = 70		70
E Track		5 x 7 = 35		35
Global Net		6 x 7 = 42		42
Others		20 x 7 = 140		140
			Total	**287**

Source: function point values provided by Software analyst of Foxconn

The accumulated results of the calculation of the unadjusted function point (UFP) values can be seen in the tables above. Based on the calculation in table 7, 8,9,10 & 11, total value of UFP of 1217 points was accumulated.

4.4 Calculating Relative Complexity Adjustment Factor

Relative Complexity Adjustment Factor (RCAF) is calculated based on the overall complexity of the system (XIA et al., 2009). RCAF is calculated using 14 General System Characteristic (GSC), on a GSC scales which starts from zero up to five (Atim et al., 2018). The zero scale shows no effect and the scale of five indicates a broad influence on the whole project. GSC calculation serves to calculate the conclusions of complexity in which there are 14 points characteristics of the software system (Atin et al., 2018). The 14 criteria for GSC calculation is shown in table 12 below. The assessment of the complexity of 14 criteria for GSC calculations has a scale of zero to five where the value is zero = no effect, 1 = incidental, 2 = moderate, 3 = average, 4 = significant and 5 = essential. After calculating the UFP, we assign values base of the impacts to the IS as done on the table below.

Table 12: General system characteristics

No	General System Characteristic (GSC)	Value of Interest
1	The level of data communication complexity	5
2	The level of complexity of data processing	5
3	Level of performance complexity	4
4	The level of configuration complexity	5
5	Software user frequency level	4
6	Data input frequency level	5
7	Level of ease of use for the user	3
8	Data frequency update rate	4
9	The level of complexity of data processing	4
10	Level of possible reuse / reusable program code	4
11	Level of ease in installation	4
12	Level of ease of operational software (backup, recovery, etc.)	5
13	The software level is made for multiple organizations/companies / clients	4
14	Level of complexity in following changes/flexible	4
	Total RCAF	**60**

Source: GCS Ratings assigned by the project manager of ORION

The results of the assessment of system complexity using GSC can be seen in Table 12. By using the 14 GSC criteria, the RCAF score shows 60 points.

STEP 3

4.5 Calculating Value Adjustment Factor

After performing GSC calculations, the next step in calculating the complexity of a project is to calculate the value adjusted factor (VAF) which takes into account the supposed contribution of technical and quality requirements (Atim et al., 2018). The VAF is calculated from the result obtain from the GSC, using Equation 6 below; The GSC includes the characteristics used to evaluate the overall complexity of the software. *Ci* is the Degree of Influence (DI) rating of each GSC.

$$VAF = 0.6 + 0.01 \sum_{i=1}^{14} Ci \quad (6)$$

VAF = (0.65 + 0.01 x 60) = 1.90

Using the score from calculation of the GSC we calculate the VAF which gives us 1.90.

STEP 4

4.6 Calculating Function Point

As a final step, we calculate the FP by the multiplying of UFP and VAF, as expressed in the Equation 7 below.

$$FP = UFP \times VAF \quad (7)$$

FP = 1217 x 1.90

= 2312.3 FP

Therefore based on the calculation of the function point, the estimated complexity obtained for the ORION project is 2312.3, which exceed the complex mark assigned for the IS. It can thus then concluded that ORION information system Project has a high level of complexity. This figure can then be used in calculating the estimated cost and project time.

4.7 Estimated cost and Project time

After going through several stages in the FP calculation, we obtained the total point function value of 2312.3 points. This value is then used in calculating the estimated time required to finish the project as show in equation 8 below:

$$\text{Estimate Time} = \text{Total FP} / (\text{No. of Developers x 6 FP}) \tag{8}$$

$$\text{Estimate Time} = 2312.3 / (6 \times 6) =$$

$$64.23 \text{ Weeks}$$

$$= (64 \text{ Weeks})$$

With the assumption that, this project has 6 developers who can work on 6 FP per function week. We estimate that the total duration of this project is approximately 64 weeks which accumulate to 2560 working hours taking into accounts that each programmer works 8 hours a day.

As for the estimated cost, it can be calculated by multiplying labor rates per week by the estimated amount of time. The calculation of the estimated cost required for the completion of the project giving that the labor rate is 1200czk/ hour as follows:

$$\text{Estimate Cost} = \text{labor rates x the estimated amount of time} \tag{9}$$

$$\text{Estimate Cost} = 1200 \text{ CZK} \times 2560 = \text{CZK3, 072,000.00}$$

Based on the calculation of the function point we obtained the level of project complexity, the estimated cost and time required for completion of the project. There are several conclusions obtained from the results of this study, the complexity of the project is either simple, medium or complex base on the initial scale define by the company. We can also estimate the cost of the project so that the offer is not too high or too low. Finally, we can estimate the time of the project so that the project doesn’t finish too fast or take too long in its completion. From the base complexity rating given by the company, we can see that the IS demonstrates a high complexity level which intend manifest in its cost and time for its completion. Management that can make a decision to either reduce the requirements or add more depending on the budget allocated for the project. Looking at the time for the completion of the project, management can outsource the IS or develop it taking into account the market changes and obsolete factors that may results after the completion of the project.

5. CONCLUSIONS AND RECOMENDATIONS

In concluding this document, various aspects relevant to the success of the IS complexity and the methods of assessing them were investigated, in accordance with the current trends and methods.

This thesis was set to describe the current ways of assessing the complexity of information systems to assess the complexity of the information system of a selected company. The various aspects of the study have been presented in chapters 1 to 4. This chapter documents the conclusions and recommendations of the project by reviewing the extent to which the aims and objectives of the study have been met. The chapter also makes recommendations for improving the use of the function point method and its relevance as a current approach of assessing the complexity of IS. Various kinds of literature were reviewed to how information systems have affected positively and negatively on organizations in the past years.

The main objective of this thesis as presented in chapter 1.2 was current ways of assessing the complexity of IS. Some well-known methods such as the function point method, the data on documents methods, database complexity and the constructive cost model method were discussed. Again, in chapter 4 of the thesis, using the function point method to assess the complexity of Foxconn IS was discussed. Objectives one to three have been met through chapters 1, 2, 3 and 4.

Through chapter three it was identified that most of the method for assessing the complexity of an information system such as the data on documents methods, the constructive cost model and the database complexity originate from the function point method. Again it was clear that the function point method is regulated by the users of the method which is the international function point user groups.

5.1 CONCLUSION

The aim of this thesis has been to identify the methods that are used to assess the complexity of business information system. This has been achieved by conducting a semi-structured analysis of complexity analysis of the IS of Foxconn. As presented in the previous chapters, IS has become a tool for competition used by businesses. There is a growing interdependence between a firm’s ability to use information systems and its ability to implement corporate strategies and achieve corporate goals. What business would like to do in a few years often rely on what its IS will be able to do. Rising market share, becoming the high-quality or low-cost producer, developing new products, and increasing employee productivity depend more and more on the kinds and quality of information systems in the organization. The need for complexity thus has become an inevitable concept in Business information systems. The higher and more complex an IS, the higher the cost and time incurred by such organizations. The complexity of IS goes a long way to determine factors such as the competing edge of the business, the cost required for maintaining the IS, the effort required to develop the IS, etc. The following conclusions are made by this project;

- Complexity can be used to inform information systems analysis, and how some individuals and organizations are using notions of complexity in predicting the success rate of organizations.
- Complexity has many aspects; some organizations are dealing with technical and physical infrastructure complexity, as well as the application of complexity in specific areas such as supply chain management and network management.
- Accurately projecting and tracking software costs is difficult, and cost overruns often occur. It is very important, therefore, to understand software estimating processes and methods.
- To facilitate complexity measuring process, some methods for software complexity estimation has been created. Estimations are less correct, but the measuring is faster and easier. Estimation methods could be proved by correlation level with FP.
- Even though organizations put a lot of money into developing IS, hoping this will better their operations and strategic position, many fail in their intent. To return the investment put into IT, organizations first need to improve their ability to deliver IS projects on time. The complexity of IS design stems from the fact that, besides a technological basis, it implies and covers business processes, and reflects on the final project goal, time, costs and quality.
- The application of the function point method demands a full, detailed level of documentation, like a functional specification of the software system being measured. There are some situations in which, instead of using the FP method, it would be better to apply a method for the estimation of functional size.
- Methods for estimating software system development can be generally put into two groups. These are Direct Estimation Methods and Derived Estimation Method.
- Data on Documents method concentrates on measuring the quantity of data on documents. The point is to obtain certain indicators, as simply as possible, which could

be compared to function points, since the FP method is the most widespread and the most accepted method.

- There are two measures derived from measuring complexity using the function point analysis: they are project development time and cost.

5.2 RECOMMENDATIONS FOR FURTHER STUDIES

Understanding and effectively managing IS complexity to enhance success rate has become a strategic issue for present-day organizations. IS activities in most organizations are organized on the basis of complexity analysis. Despite the practical importance of IS project complexity, there has been little theoretical and operational understanding of the interrelationship between complexity and success of IS. Although various measures of IS complexity have been proposed in this project, these measures are not adequate to assess the total complexity of ISs. Software is a core part of an IS project; however, there are many other important organizational or environmental factors that contribute to the overall complexity of the IS project. Due to time and resource constraints, this project collected some of the already existing methods of assessing the complexity of IS. To be able to have a deeper insight into the use of and understand well how these methods are used by researchers in the field, this research recommends that further studies are carried out that will involve investigating how organizations use information system complexity as a competing tool.

These additional factors need to be taken into account to accurately assess the degree of complexity of an entire organization's IS. It is therefore important that readers who will want to know more about the entire complexity of an organization IS channel their studies to analyzing other organizational factors that influence the complexity of an organization.

REFERENCES

[1] ABU-DOLEH, Jamal, & David Weir. "Dimensions of performance appraisal systems in Jordanian private and public organizations."*The International Journal of Human Resource Management.* 2007, 18(1), 75-84. ISSN: 0958-5192.

[2] ALAMOUDI Doaa, Kumar Amlendu. Information System Complexity and Business Value. *International Journal of Economics & Management Sciences.* 2017. 12(2),201-298. ISSN: 2162-6359.

[3] AMARATUNGA, Dilanthi, David Baldry, & Marjan Sarshar. "Process improvement through performance measurement: the balanced scorecard methodology."*Work study.* 2001, 50(5), 179-189. ISSN: 0043-8022.

[4] ARTETA, Mikel., & Ronald, Giachetti. "A measure of agility as the complexity of the enterprise system."*Robotics and computer-integrated manufacturing.* 2004(6), 495-503. ISBN: 978-9998176850.

[5] ATIN, Sufa, Tati. Harihayati, & Widianti D. "Utilization of function point method for measuring software project complexity." *IOP Conference Series: Materials Science and Engineering*. 2018, 407(1), 1-5. ISSN: 1757-899X.

[6] AVISON, David, & Guy Fitzgerald. *Information systems development: methodologies, techniques and tools*. McGraw Hill, 2002. pp. 608 ISBN: 978-0077096267.

[7] BANKER, Rajiv , Srikant Datar, Chris Kemerer, & Dani Zweig. "Software complexity and maintenance costs."*Communications of the ACM*. 1993, 36(11), 81-95.

[8] BAZEWICZ, Jacek. 2003. *Handbook on data managementin information systems*. Berlin: Springer-Verlag, 2003. pp. 65-120. ISBN 3-540-43893-9.

[9] BOCIJ, Paul, Greasley Andrew, & Hickie Simon. *Business information systems: Technology, development and management*. Pearson education, 2008. pp 736 ISBN: 978-0273716624.

[10] BOEHM, Barry W. *Software engineering economics*.vol. 197. Englewood Cliffs (NJ): Prentice-hall, 1981.pp. 767. ISBN: 978-0138221225.

[11] BOURGEOIS, David. *Information systems for business and beyond*. The Saylor Foundation, 2014.pp.168. ISBN: 978-1304943484.

[12] BOURNE, Mike, John Mills, Mark Wilcox, Andy Neely, & Ken Platts. "Designing, implementing and updating performance measurement systems."*International journal of operations & production management*. 2000, 20(7), 754-771**.** ISSN: 0144-3577.

[13] BREHM, Lars, & Markus Lynne. "The divided software life cycle of ERP packages." In *Proceedings of 1st Global Information Technology Management (GITM) World Conference.* 2000, pp. 43-46.

[14] CARDOSO, Jorge, Mendling Jan, Neumann Gustaf, & Reijers Hajo. "A discourse on complexity of process models." In *International Conference on Business Process Management*. 2006. pp. 117-128.

[15] CHAFFEY, Dave & White Gareth. *Business information management: improving performance using information systems*. Canada: Pearson Education. 2010 pp.688. ISBN 9780273711797

[16] CHENHALL, Robert H. "Management control systems design within its organizational context: findings from contingency-based research and directions for the future."*Accounting, organizations and society*.2003. 28(2-3) 127-168. ISSN: 0361-3682

[17] CONG, Yu, & Romero Jorge. "On information systems complexity and vulnerability."*Journal of Information Systems.* 2013. 27(2). 51-64.

[18] ECCLES, Robert G., Serafeim George, & Cheng Beiting. "Foxconn Technology Group (B)."*Harvard Business School Accounting & Management Unit Case*. 2011. 112-058.

[19] Foxconn in the Czech Republic, [online]. 2017 Available at: https://www.foxconn.cz/

[20] GEORGESCU, Bogdan, Zhou Xiang, Comaniciu Dorin, Rao Bharat, & Gupta Alok. "*Method of database-guided segmentation of anatomical structures having complex appearances*".2005. 2. pp. 429-436

[21] GHARAJEDAGHI, Jamshid. *Systems thinking: Managing chaos and complexity: A platform for designing business architecture*. Morgan Kaufmann, 2011. pp. 376 ISBN: 978-0123859150

[22] GRUHN, Volker, & Laue Ralf. "Adopting the cognitive complexity measure for business process models." *2006 5th IEEE International Conference on Cognitive Informatics*. 2006. 1, pp. 236-241. ISBN: 1-4244-0475-4

[23] GRUPE, Fritz H., & Dorothy F. Clevenger. "Using function point analysis as a software development tool."*Journal of Systems Management.* 1991. 42(12).23. ISSN: 0164-1212

[24] HARDCASTLE, Elizabeth. *Business Information Systems*. Elizabeth Hnadcastle & Ventus Publishing Aps, 2008. pp 56 ISBN: 978-87-7681-463-2.

[25] SYMONS, Charles R. "Function point analysis: difficulties and improvements."*IEEE transactions on software engineering.* 1988. 14(1) 2-11. ISSN: 0098-5589

[26] HUNGER, J. David, & Wheelen Thomas L. *Essentials of strategic management*. NJ: Prentice Hall, 2010. 9, pp. 216. ISBN: 978-8120348615 .

[27] JOFRE, S. (2011). *Strategic Management: The theory and practice of strategy in (business) organizations.* Kgs. Lyngby: DTU Management.2011 pp.87 , 1. ISBN: 978-87-92706-18-8.

[28] JORGENSEN, Magne, & Kjetil Molokken-Ostvold. "How large are software cost overruns? A review of the 1994 CHAOS report."*Information and Software Technology*.2006, 48(4), 297-301. ISSN: 0950-5849.

[29] KEMERER, Chris F., & Benjamin S. Porter. "Improving the reliability of function point measurement: an empirical study."*IEEE Transactions on Software Engineering*. 1992. 18(11),1011-1024. ISSN: 0098-5589.

[30] KIM, SangEun, William M. Lively, & Dick B. Simmons. "An Effort Estimation by UML Points in Early Stage of Software Development." *Software Engineering Research and Practice*. 2006. pp. 415-421. ISBN: 1-932415-90-4

[31] KOCK Jr, Nereu F., & Robert J. Mcqueen. "Product flow, breadth and complexity of business processes: An empirical study of 15 business processes in three organizations."*Business Process Re-engineering & Management Journal*. 1996. 2(2). pp.8-22. ISSN: 1355-2503

[32] MARTIN, Robert C. *Agile software development: principles, patterns, and practices*. Pearson Education, 2002. pp.552. ISBN: 978-0135974445.

[33] MUKHOPADHYAY, Tridas, Sunder Kekre, & Suresh Kalathur. "Business value of information technology: a study of electronic data interchange."*MIS quarterly*. 1995. 19(2). pp.137-156.

[34] O'BRIEN, James A., & George M. Marakas. *Introduction to information systems*. New York City, USA: McGraw-Hill/Irwin 2005. 13. pp.768. ISBN: 978-0073376882

[35] PATRIZIA, Poščić. "*Method for Estimating the Complexity of Designing Business Information Systems*." PhD diss., Fakultet organizacije i informatike, Sveučilište u Zagrebu, 2008.

[36] PAVLIC, Mile, Marin Kaluza, & Neven Vrcek. "Database complexity measuring method." In *Central European Conference on Information and Intelligent Systems*, p. 1. Faculty of Organization and Informatics Varazdin, 2008.

[37] POŠČIĆ, Patrizia, Mile Pavlić, & Neven Vrček. "Method for Estimating the Complexity of Designing Business Information Systems."*Journal of Information and Organizational Sciences.* 2008. 32(2) 123-136. ISSN 1846-3312

[38] QURESHI, Sheheryar Mohsin, & ChangWook KANG. "Analysing the organizational factors of project complexity using structural equation modelling."*International Journal of Project Management.* 2015. 33(1). 165-176. ISSN: 0263-7863.

[39] RAO, Aibing, Rohini K. Srihari, Lei Zhu, & Aidong Zhang. "A method for measuring the complexity of image databases."*IEEE Transactions on multimedia.* 2002. 4(2). 160-173. ISSN: 1520-9210.

[40] RUBINSTEIN, David. "Standish group report: There's less development chaos today."*Software Development Times.*1. 2007

[41] SPEIER, Cheri, Iris Vessey, & Joseph S. Valacich. "The effects of interruptions, task complexity, and information presentation on computer-supported decision-making performance."*Decision Sciences.* 2003. 34(4).771-797.

[42] STAIR, Ralph M & George, Walter Reynolds. *Information Systems.* 9th ed. Boston: Course Technology, c2010. ISBN 978-0-538-47425-2.

[43] STAIR, Ralph M & George, Walter Reynolds. *Principles of Information Systems: a managerial approach.* 7th. Boston: Thomson Learning, c2006. ISBN 0-619-21525-9.

[44] STERMAN, John D. *Business dynamics: systems thinking and modeling for a complex world*. McGraw-Hill Education.2000.pp.1008. ISBN: 978-0072389159.

[45] SYMONS, Charles R. "Function point analysis: difficulties and improvements."*IEEE transactions on software engineering.* 1988. 14(1). 2-11. ISSN: 0098-5589

[46] TAGRA, Dinesh. "*Cost Estimation For Commercial Software Development Organizations*." LAP LAMBERT Academic Publishing, 2011. pp.92.ISBN: 978-3847334583.

[47] TAIT, Peter, & Iris Vessey. "The effect of user involvement on system success: a contingency approach."*MIS quarterly*. 1988. 12(1), 91-108.

[48] VASCONCELOS, Flávio C., & Rafael RAMIREZ. "Complexity in business environments."*Journal of Business Research*. 2011, 64(3), 236-241. ISSN: 0148-2963.

[49] VENKATESH, Viswanath, & Fred D. DAVIS. "A theoretical extension of the technology acceptance model: Four longitudinal field studies."*Management science*. 2000. 46(2), 186-204. ISSN: 1526-5501.

[50] WALRAD, Charlene, & Eric Moss. "Measurement: the key to application development quality."*IBM Systems Journal*. 1993.32(3). 445-460. ISSN: 0018-8670.

[51] LAURIE Williams & Alistair Cockburn. "Agile software development: it’s about feedback and change."*IEEE Computer.* 2003. 36(6). 39-43. ISSN: 0018-9162.

[52] WOOD, Robert E. "Task complexity: Definition of the construct."*Organizational behavior and human decision processes.* 1986. 37(1). 60-82. ISSN: 0749-5978.

[53] XIA, Weidong, & Gwanhoo Lee. "Grasping the Complexity of Information Systems Development Projects: A Taxonomy and Assessment. " *A communication of the ACM* .2004. 47(5). 68-74.