SEVEN STAGE MODEL OF THE SOFT SYSTEM METHODOLOGY FOR SOFTWARE DEVELOPMENT PROCESS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

GÖKÇE GÜNAYDIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

JUNE 2023

Approval of the thesis:

SEVEN STAGE MODEL OF THE SOFT SYSTEM METHODOLOGY FOR SOFTWARE DEVELOPMENT PROCESS

Submitted by Gökçe GÜNAYDIN in partial fulfillment of the requirements for the degree of **Master of Science in Information Systems Department, Middle East Technical University** by,

Date: **23.06.2023**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Gökçe Günaydın

Signature :

ABSTRACT

SEVEN STAGE MODEL OF THE SOFT SYSTEM METHODOLOGY FOR SOFTWARE DEVELOPMENT PROCESS

Günaydın, Gökçe
MSc., Department of Information Systems
Supervisor: Prof. Dr. Sevgi Özkan Yıldırım

June 2023, 75 pages

Organizations always try to improve processes, optimize workflows and minimize turnover rate in order to enhance efficiency, increase productivity, reduce waste and improve performance. Process improvement is a vital component for organizations and it can be achieved through a variety of quality methods, such as, total quality management, Lean, Six Sigma and etc. Process improvement begins with complete understanding of current processes which requires data analysis, stakeholder engagement followed by an implementation with proper methods. Soft Systems Methodology (SSM) is a perfect problem-solving approach to analyze complex processes and to find suitable implementations.

This paper uses SSM in software development process (SDP) within two interrelated organizations. In this research, SSM applies some problem-solving techniques, such as systems thinking, rich picture drawings, root definition and related systems construction, conceptual model creations and action planning to maintain continuous process improvement.

This thesis study contains four sub-processes of SDP. These sub-processes are called Design Management, Technical Development, Testing and Training processes. Seven-staged SSM is applied to sub-processes in order to enhance system performance. Last stage of SSM, which is action taking, is presented however it is not applied yet. Apart from this, rich pictures of sub-processes are drawn, relevant systems and root definition are identified, conceptual models are created with focus group studies. Lastly, suggestions are offered regarding to real world activities and conceptual models.

Results of study shows effectiveness of SSM for ill-structured problems in SDP. It emphasizes real world activities and proposes needed activities in sub-processes to overcome issues in SDP.

Keywords: Soft Systems Methodology, Software Development Process, Process Improvement, Systems Thinking

ÖZ

YAZILIM GELİŞTİRME SÜRECİ İÇİN SOFT SİSTEM METODOLOJİSİNİN YEDİ AŞAMA MODELİ

Günaydın, Gökçe
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Prof. Dr. Sevgi Özkan Yıldırım

Haziran 2023, 75 sayfa

Organizasyonlar, verimliliği artırmak, üretkenliği yükseltmek, israfı azaltmak ve performansı iyileştirmek için süreçleri geliştirmeye, iş akışlarını optimize etmeye ve işgücü devrini en aza indirmeye çalışır. Süreç iyileştirme; toplam kalite yönetimi, Lean, Six Sigma gibi çeşitli kalite yöntemleriyle gerçekleştirilebilir. Süreç iyileştirme, mevcut süreçlerin tam anlaşılmasıyla başlar ve veri analizi, paydaş katılımı ve uygun yöntemlerle bir uygulama gerektirir. Soft Sistem Metodolojisi (SSM), karmaşık süreçleri analiz etmek ve uygun uygulamaları bulmak için mükemmel bir problem çözme yaklaşımıdır.

Bu makale, birbiriyle bağlantılı iki organizasyon arasında yazılım geliştirme sürecinde (YGS) SSM'in kullanımına odaklanmaktadır. Bu araştırmada, SSM, sistemik düşünme, zengin resim çizimleri, kök tanımı ve ilgili sistem yapılandırması, kavramsal model oluşturma ve eylem planlaması gibi bazı problem çözme tekniklerini sürekli süreç iyileştirmeyi sürdürmek için uygular.

Bu tez çalışması, YGS'nin dört alt sürecini içermektedir. Bu alt süreçler Tasarım Yönetimi, Teknik Geliştirme, Test ve Eğitim süreçleri olarak adlandırılmaktadır. Yedi aşamalı SSM, alt süreçlere uygulanarak sistem performansının artırılması amaçlanmaktadır. SSM'nin son aşaması olan eyleme geçme aşaması sunulmuş, ancak henüz uygulanamamıştır. Bunun yanı sıra, alt süreçlerin zengin resimleri çizilmiş, ilgili sistemler ve kök tanımı belirlenmiş, odak grup çalışmalarıyla kavramsal modeller oluşturulmuştur. Son olarak, gerçek dünya faaliyetleri ve kavramsal modellerle ilgili öneriler sunulmuştur.

Araştırmanın sonuçları, SSM'nin YGS'deki yapılandırılmamış problemler için etkinliğini göstermektedir. Gerçek dünya faaliyetlerine vurgu yapmakta ve SDP'deki sorunları aşmak için alt süreçlerde gereken faaliyetleri önermektedir.

Anahtar Sözcükler: Soft Sistem Metodolojisi, Yazılım Geliştirme Süreci, Süreç İyileştirme, Sistem Düşüncesi

To My Family

ACKNOWLEDGMENTS

Firstly, I would like to express my appreciation to Prof. Dr. Sevgi Özkan Yıldırım for her guidance through the whole process of my master thesis.

I wish to express my special thanks to the members of my thesis committee Prof. Dr. Soner Yıldırım, Asst. Prof. Dr. Banu Yüksel Özkaya, Asst. Prof. Dr. Özden Özcan Top and Asst. Prof. Dr. Tuna Hacaloğlu for their support, criticism, comments and advice.

I would also like to thank all of participants, employees, and managers from investigated company who willingly participated and helped in the process of data collection for the research study.

I would also like to thank me for not quitting in this thesis process and I would like to thank my family for their continuous support.

TABLE OF CONTENTS

ABSTRACT vi

ÖZ........ vii

DEDICATION xx

ACKNOWLEDGMENTS........x

TABLE OF CONTENTS xi

LIST OF TABLES xvv

LIST OF FIGURES........ xvi

LIST OF ABBREVIATIONS xviii

CHAPTERS

1. INTRODUCTION........1

1.1. Introduction 1

1.2. Purpose Statement 2

1.3. Research Questions........3

1.4. Significance of Study........3

1.5. Structure of Thesis........3

2. LITERATURE REVIEW........5

2.1. Software Processes in Organizations........5

2.1.1. Software Process Improvement........7

2.2. Systems Thinking 9

2.2.1. Hard and Soft Systems Thinking 10

2.3. Soft System Methodology(SSM)........11

3. RESEARCH METHODOLOGY 19

3.1. Research Philosopy........19

3.2. Research Method 19

3.3. Research Design 20

3.3.1. Data Collection........20

3.3.2. Data Analysis 21

3.3.2.1. System Thinking Behavior of Processes. 21

3.3.2.2. Soft Systems Methodology. 22

3.4. Trustworthiness and Triangulation 23
3.5. Ethical Considerations 24
4. EMPIRICAL FINDINGS: CASE STUDY APPLICATION 25
4.1. Software Development Process 25
4.2. Stage 1: Finding out about Problem Situation 27
4.3. Stage 2: Analysis 27
4.3.1. Design Management Process 28
4.3.2. Technical Development Process 32
4.3.3. Testing Process 36
4.3.4. Training Supporting Process 40
4.4. Stage 3: Root Definition of Related Systems 43
4.5. Stage 4: Conceptual Models of Sub-Processes 43
4.5.1. Design Management Process 44
4.5.2. Technical Development Process 48
4.5.3. Testing Process 50
4.5.4. Training Process 52
4.6. Stage 5: Comparison 54
4.6.1. Design Management Process 54
4.6.2. Technical Development Process 56
4.6.3. Testing Process 57
4.6.4. Training Process 58
4.7. Stage 7: Debate 59
4.8. Stage 8: Take Action 60
5. CONCLUSION 61
5.1. Discussion 61
5.2. Conclusion 62
5.3. Contribution to Learning 62
5.4. Limitations of the Study 62
5.5. Delimitation of the Study 64
5.6. Future Research 64

REFERENCES 65
APPENDICES 69
APPENDIX A 69
APPENDIX B 73

LIST OF TABLES

Table 1: Systems Thinking Definitions and Key Concepts ... 10

Table 2: Elements of Conceptual Model of Design Management ... 47

Table 3: Elements of Conceptual Model of Technical Development Sub-Process ... 50

Table 4: Elements of Conceptual Model of Testing Sub-Process ... 52

Table 5: Elements of Conceptual Model of Training Process ... 53

Table 6: Comparison of Design Process Conceptual Model and Real World Activity ... 54

Table 7: Comparison of Technical Development Process Conceptual Model and Real World Activity ... 56

Table 8: Comparison of Testing Process Conceptual Model and Real World Activity ... 57

Table 9: Comparison of Training Process Conceptual Model and Real World Activity ... 58

LIST OF FIGURES

Figure 1: Structure of Thesis..4
Figure 2: Steps in Descriptive Software Process Modeling (adapted from Becker-Kornstaedt et al, 2000) 6
Figure 3: People's Contribution to Software Process Improvement (adapted from Korsaa M., Johansen J., Schweigert T., et al, 2013)..6
Figure 4: Lean Thinking Methodologies (adapted from Jiang, W., 2018)........................9
Figure 5: Hard and Soft Systems Thinking(adapted from Checkland, P.,2000).............11
Figure 6: The Purpose of Soft Systems Methodology(adapted from Gasson, S.,1994) .12
Figure 7: The Standard Seven Stage of SSM(adapted from Gasson, S.,1994) 14
Figure 8: Problem Diagram Example (adapted from Gasson, S.,1994)..........................16
Figure 9: Meta Model of BPMN Elements (adapted from Zarour, K.et al, 2020)..........17
Figure 10: Basic Rich Picture of Software Development Process 26
Figure 11: Organizational Charts and Relations of Two Firms 27
Figure 12: Issues and Primary Task Related Areas...28
Figure 13: Sub-Processes of Software Development Process...29
Figure 14: Rich Picture of Design Process...30
Figure 15: Value Chain Diagram of Design Management Sub-Process.........................31
Figure 16: BPMN of Design Management Sub-Process...32
Figure 17: Rich Picture of Technical Development Sub-Process..................................33
Figure 18: Value Chain Diagram of Technical Development Sub-Process....................35
Figure 19: BPMN of Technical Development Sub-Process..36
Figure 20: Rich Picture of Testing Process...38
Figure 21: Value Chain Diagram of Testing Sub-Process 39
Figure 22: BPMN of Testing Sub-Process...40
Figure 23: Rich Picture of Training Process 42
Figure 24: Value Chain Diagram of Trainig Process..43
Figure 25: BPMN of Training Process..43
Figure 26: Conceptual Model of Design Management Sub-Process 46
Figure 27: Conceptual Model of Technical Development Sub-Process 49
Figure 28: Conceptual Model of Testing Sub-Process..51
Figure 29: Conceptual Model of Training Sub-Process..53

LIST OF ABBREVIATIONS

BPMN	Business Process Modelling Notation
CMMI	Capability Maturity Model Integrated
CTQ	Critical-to-Quality
DMAIC	Define, Measure, Analyze, Improve, Control
ERP	Enterprise Resource Planning
EPC	Event-Driven Process Chain
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
PDCA	Plan-Do-Check-Act
QIP	Quality Improvement Paradigm
SPICE	Software Process Improvement and Capability Determination
TQM	Total Quality Management
UML	Unified Modelling Language

CHAPTER 1

INTRODUCTION

1.1. Introduction

Individuals and organizations of all sizes and types use Process Improvement (PI) in order to increase their productivity. PI is a critical aspect of organizations that want to adapt changeable market conditions and stay competitive in the dynamic business environment. PI approach can be used both organizational management-wide and an employee-wide. Organizations can decide the areas to be developed as well as an employee can be developed by trainings, mentoring activities and practices. PI covers continuous improvement in order to achieve steady results in changing nature of business environment.

Enterprise resource planning (ERP) is a software system that most of the organizations use in order to keep track of their data and monitor processes. ERP integrates business processes and provides unified view of operations. Software development is an essential part of an ERP system, however ERP software development does not have so much difference when compared to other non-ERP software development processes.

Software development process (SDP) in organizations requires human involvement, hardware, software, proper project management, clear objectives and continuous improvement to ensure that software product become relevant over time. There are lots of methodology defined in literature for SDP such as Agile, Waterfall, Kanban, Spiral and etc. These methodologies mostly cover how the operations should work with standardized guidelines. In broad terms, SDP starts with requirements engineering, followed by code development, testing and going live. However, PI needs detailed investigation of these processes in order to find bottlenecks and identify inefficiencies that prevent smooth functioning of SDP. To achieve this, systems thinking and PI methodologies become prominent.

Software development and its sub-processes forms a system since system can be defined as "a set of components which are related by some form of interaction, and which act together to achieve some objective or purpose" in broad terms. In SDP, components are simply the individual parts which are sub-processes, relationships are interactions between sub-processes and objective is the desired outcome which the process is attempting to achieve. Hence, using a systems thinking methodology is a preferred way to understand problems of overall process.

There are lots of systems thinking methodologies in the literature such as soft systems thinking, critical systems thinking, viable systems thinking, hard systems thinking and etc. Soft systems methodology is a soft systems thinking approach that helps to understand and solve problematic areas in SDP in order to gain higher quality end products. SSM defines the problematic areas by using rich pictures and current process drawings and proposes improvements to all stakeholders in order to remove problematic areas. By applying this approach all the time, organizations achieve continuous process improvement.

The ultimate goal of both process improvement and systems thinking methodologies is to improve quality of end product and reduce any kind of wastes such as waiting, motion, defects and etc. Herein, end product of software development is developed software and wastes can be named as defects, resources, processing redundant steps and waiting. Hence, PI and systems thinking concepts cannot be separated from each other. PI techniques are some quality methodologies which can be six-sigma, kaizen, lean, hoshin kanri and total quality management (TQM). Among these, TQM mainly focuses on overall process improvement and emphasizes continuous improvement with involvement of all employees in the process.

SDP can use SSM for several reasons. Cause of SSM is a flexible and adaptable approach, it can address complex problems that are encountered in SDP. In addition to that, SSM values stakeholder participation which is also important in SDP to ensure developed software meets the need that is formed by stakeholder's perspectives. Lastly, SSM highlights the continuous improvement which is very suitable for SDP that is an iterative process by nature.

This study intends to use SSM in ERP SDP within organizations. Three sub-processes and one supporting sub-processes of SDP will be examined in detail. Results of the SSM will be shared with stakeholders. But implementations and usage of proposed changes are not within the scope of this study.

1.2. Purpose Statement

The purpose of this research is to define enhancements that will improve overall software development process. Research investigates each sub-process to determine and propose changes in order to have more effective and efficient software development process. Soft systems methodology is used to describe process with gathering data from meetings and face-to-face interviews, drawing rich pictures of sub-processes and identifying root definitions. To propose changes, conceptual model creations and comparisons of conceptual models with real world activities are done with respect to SSM's stages.

This research also serves as a model to stakeholders of the examined process about how to sustain continuous improvement and how to establish proper communication channels within organizations. In order to gather data about current software development processes and its sub-processes, semi-structured interviews and meetings are done with designers, developers and relevant managers. SSM provides a deeper understanding of current processes as well as proposing related changes. However, application and evaluation of these changes are not within the scope of this research.

1.3. Research Questions

There may many questions that addresses process improvement in software development process in ERP environment. Nevertheless, with respect to purpose of the study, the researcher seeks answers to below research questions.

RQ1: How does soft systems methodology helps to improve ERP software development process?

RQ2: In what ways soft systems methodology will help to improve the examined process?

1.4. Significance of the Study

The study has various important implications. Investigating software development process in ERP environment with using soft systems methodology is not a common practice for process improvement. There is a few written literature on this issue. However, there are plenty of applications of soft systems methodology into software development processes in the literature. Hence, applying soft systems methodology into ERP software development process is not that different from usual implementations of soft systems methodology into software development process.

Moreover, organizations that are categorized as medium to big size enterprise often encounters dysfunctional operations that may cause significant amount of value decrease in the economy. In order to avoid such losses, continuous process improvements should be done with involvement of responsible participants. Application of soft systems methodology in such firms gives deeper understanding of their processes and willingness to improve processes further.

1.5. Structure of the Thesis

This study consists of five main parts. In Chapter 1, a brief summary of the study is presented, purpose, research questions and significance of the study are stated. After this, in Chapter 2, a literature review about systems thinking methodologies, process improvement methodologies and systems thinking methodologies in software development process is conducted. An overview of ERP systems are depicted. Chapter 3 discusses research methodologies for SSM in software development process. Chapter 4 involves an application of SSM in an ERP software development process. Final Chapter, the limitations and constraints of the study is specified and potential future studies are explored.

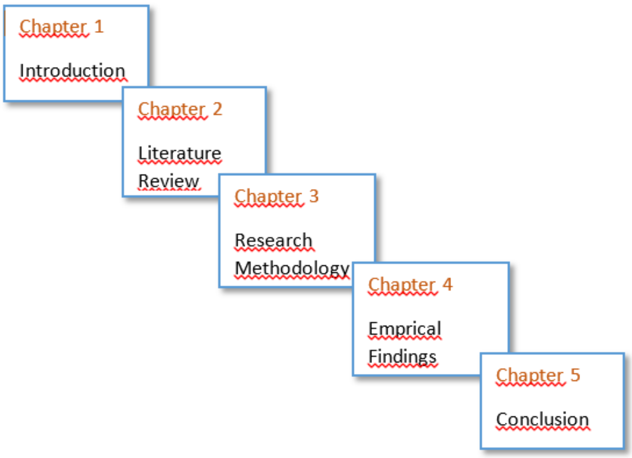


Figure 1. Structure of Thesis

CHAPTER 2

LITERATURE REVIEW

2.1. Software Processes in Organizations

Industrial manufacturing processes aims to achieve repeatability to minimize uncertainty, hence it is reasonable to have separate design and manufacturing processes. However, software is composed of code and this composition brings an uncertainty. This uncertainty can be eliminated by collaborating design and software engineering teams. (Lucena, P. & Braz, A &, Chicoria A. and Tizzei L., 2016) Software has become vital for all sectors such as business and economic initiatives, industrial infrastructure, education, politics, entertainment and etc. Consequently, software development processes gained attention and extended their existing approaches such as agile methods, open source development and etc. (Fugetta, A. & Di Nitto, E., 2014) In addition to software development approaches, software processes do not only involve coding, it needs process modelling, process requirements and process evaluation. (Osterweil, L., 1997) Hence, software development process incorporates other process elements.

Software process has four main activities (Sommerville, 2004)

1. Defining software specification
2. Designing and implementing software
3. Validating and verifying software
4. Enabling evolution of software for future changes.

The software process activities should be understood clearly. In order to do that, software processes should be modeled appropriately and related metrics should be evaluated (Canfora, G., Garcia F., Piattini, M. & Visaggio, C.A., 2004) Software process models can be summarized as descriptive and active models.(Curtis et al, 1992 & Dowson and Fernström, 1994) Descriptive software processes are the processes that define processes and organizational behavior with considering relations between roles, activities and artefacts entities. On the other hand, active models are the executable systems that support the flow of processes. Descriptive software process models uses qualitative analysis in order to handle process modeling. The data sources for process elicitation are consist of documents, interviews with process performers and observations of the process and process performers. (Becker-Kornstaedt et al,,2000) Descriptive process modeling is used in process improvement and main steps are depicted below figure.

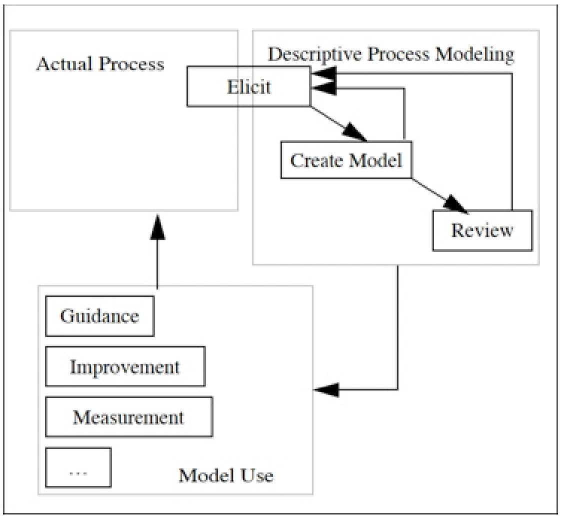


Figure 2. Steps in Descriptive Software Process Modeling (adapted from Becker-Kornstaedt et al, 2000)

Since descriptive process models are more human-centered, a software process cannot be developed without involvement of people. Employees are the valuable assets for sensing urgency, idea creation and idea processing, The below improvability model depicts contributions of people on successful innovation and process improvement. (Korsaa M., Johansen J., Schweigert T., et al, 2013)

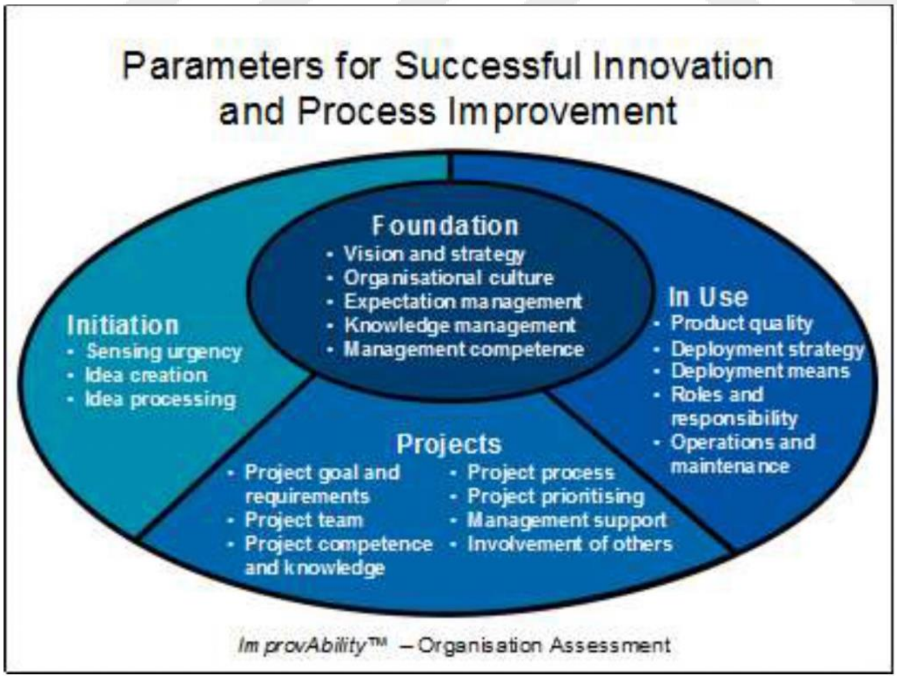


Figure 3. People’s Contribution to Software Process Improvement (adapted from Korsaa M., Johansen J., Schweigert T., et al, 2013)

The model suggests an approach that analyses 20 human related parameters that can be risk for any process improvement projects (which are also suitable for software processes)

2.1.1.Software Process Improvement

Software process improvement can be affected by various factors. According to Perry E.D et al (1994), in order to understand the structure of software development process and reduce the software development time to achieve process improvement, organizational and social context of software process environment should be investigated. In the paper, three reasons are determined:

- Human element is crucial for software process
- Usage of various software tools can be complicated. Instead of using various software tools, use what is suitable for employees that are involved in software process. Thus technology will be incorporated with organizational and social infrastructure.
- Most of the software development project issues come from outside of the programming. Eliminate unnecessary meetings, machine downtimes, paperwork and other company works if it is possible.

Software process improvement does not have different principles than process improvement. Basically, process improvement has four main steps that can be executed iteratively. (Cheng, C.K., Permadi, R.B., 2009)

- Determining the current process
- Making a detailed improvement plan
- Implementing related developments with respect to plan
- Measuring the developed plan

While performing above software process improvement steps, software process improvement pays attention to three foundations which are frameworks, practices and tools. Frameworks are guiding models to improve current processes. For instance, capability maturity model integrated (CMMI), software process improvement and capability determination (SPICE), quality improvement paradigm (QIP), six sigma, lean thinking and etc. Practices are development activities that are planned and performed to achieve specific software process. Examples are inspections, test-driven development and etc. Tools are applications like Bugzilla, IBM rational requisite pro, Telelogic Doors and etc. that support software development activities.

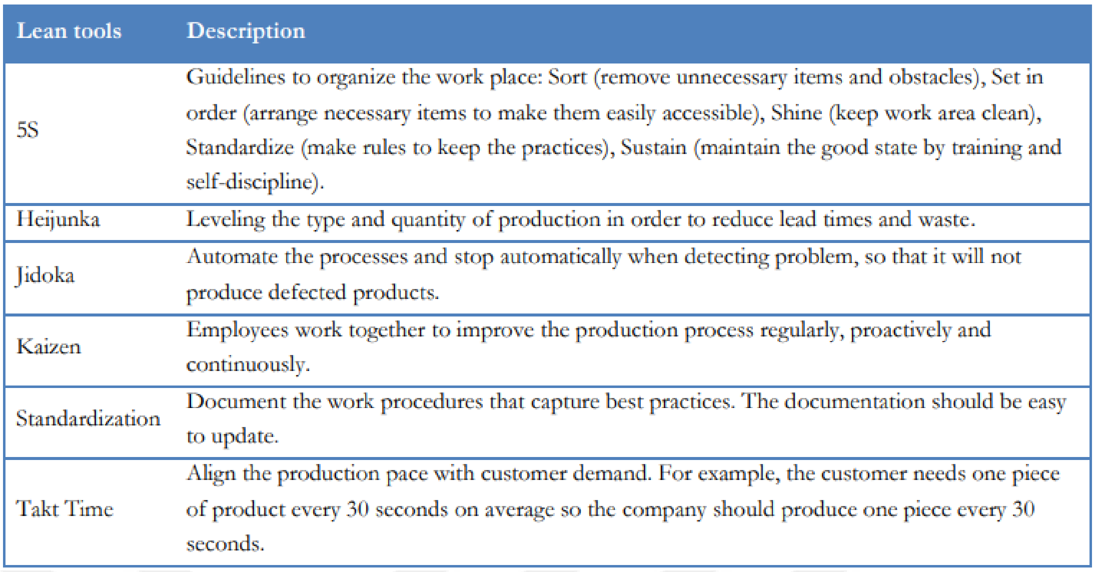
Most of the process improvement tools are also applicable for software process improvement tools. Quality of processes, end products are directly related with process improvement. Some of the main process improvement methodologies are Total Quality Management (TQM), Six Sigma Methodology, Lean Thinking, Kaizen, Plan-Do-Check-Act(PDCA) cycle and moreover, some software process specific guides are defined in order to improve software processes such as capability maturity model integrated (CMMI), software process improvement and capability determination (SPICE) and etc. (Cheng, C.K., Permadi, R.B., 2009)

Definition of TQM is important cause it aims overall quality improvement which needs a broader meaning. Ross, J.E. (1994) defines TQM as "…the integration of all functions and processes within an organization in order to achieve continuous improvement of the quality of goods and services. The goal is customer satisfaction." Nonetheless, Boaden, R.J (1997) needed a list that TQM involves to make proper definition. The list includes;

- "Customer focus, with emphasis on the customer-supplier relationship, internally and externally.
- The commitment of everyone to quality of improvement, especially managers.
- Training and education considered as an investment.
- The involvement of everyone within the organization in quality improvement.
- A focus on processes.
- The use of teams and teamwork
- The use of appropriate tools and techniques, reviewed regularly.
- Goal-setting, measurement and feedback for all aspects of the business.
- Continuous improvement as a philosophy.
- A change in the culture of the organization, i.e. the way people think and behave.
- The inclusion of quality principles into product and service design."

Even though TQM seems more close to qualitative approach and needs organizational effort to establish continuous improvement, six-sigma methodology is more statistical tool that is widely used. It uses define, measure, analyze, improve and control (DMAIC) cycle by considering critical-to-quality (CTQs) characteristics. It expects at most three-sigma deviations over mean from both sides in order to have more quality on end products (Green, F.B., 2006)

Lean systems are designed to eliminate waste. Womack et al. (1990) defines lean systems as: "… uses less of everything compared with mass production – half the human effort in the factory, half the manufacturing space, half the investment in tools, half the engineering hours to develop a new product in half the time. Also it requires keeping far less than half the needed inventory on site, results in many fewer defects, and produces a greater and ever growing variety of product."
Lean systems thinking uses below methodologies in order to reduce waste.



Lean tools	Description
5S	Guidelines to organize the work place: Sort (remove unnecessary items and obstacles), Set in order (arrange necessary items to make them easily accessible), Shine (keep work area clean), Standardize (make rules to keep the practices), Sustain (maintain the good state by training and self-discipline).
Heijunka	Leveling the type and quantity of production in order to reduce lead times and waste.
Jidoka	Automate the processes and stop automatically when detecting problem, so that it will not produce defected products.
Kaizen	Employees work together to improve the production process regularly, proactively and continuously.
Standardization	Document the work procedures that capture best practices. The documentation should be easy to update.
Takt Time	Align the production pace with customer demand. For example, the customer needs one piece of product every 30 seconds on average so the company should produce one piece every 30 seconds.

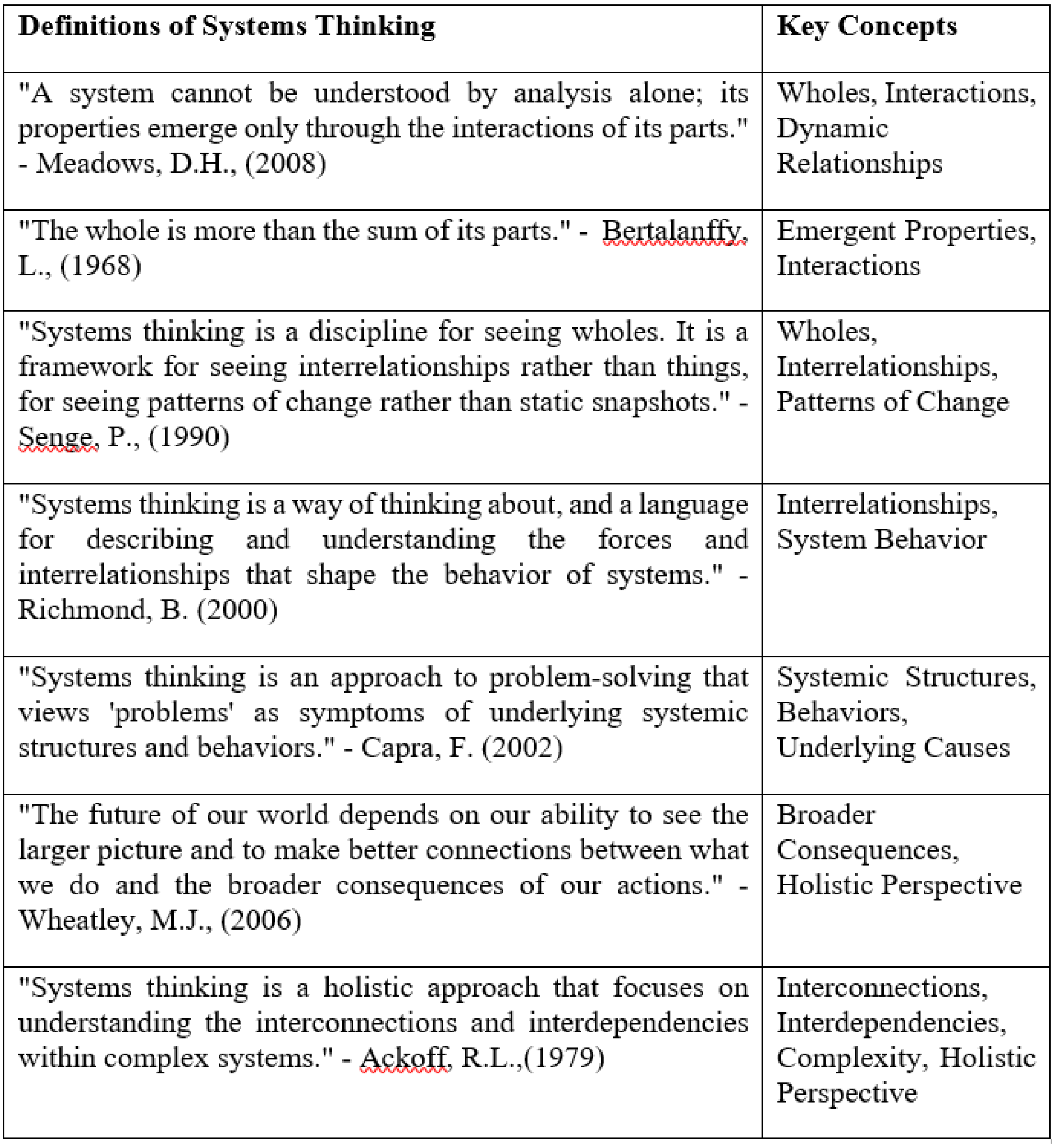
Figure 4. Lean Thinking Methodologies (adapted from Jiang, W., 2018)

2.2. Systems Thinking

Origin of systems thinking probably begun as humans started to categorize distinctive characteristics of objects according to common features. This classification continuous to this day and perceived as useful process. Another aspect was that objects are composed of components which are interrelated and interdependent. However, characterizing objects and identifying relations between objects do not enough to define behaviors of objects. Objects' behavior do not comprise of simply adding its components behavior. The objects' behavior can only be understood by analyzing impact of the components to each other and system's behavior can only be evaluated by considering net influence of all components. (Turner, W.C., Mize, J.H., Case, K.E., Nazemetz, J.W., 1993) Wright, R. emphasizes the importance of systems thinking as "…enriches our awareness of nearly all dimensions of life. It explains complexity in a more comprehensive way."

Most of the definitions about systems thinking agrees with that systems compose of components that are interact with each other and collectively constitutes a system with a purpose. Richmond, B. (2000) defines systems thinking as "a way of thinking about, and a language for describing and understanding, the forces and interrelationships that shape the behavior of systems." Moreover, Senge, P. (1990), sees systems as a whole and states that "systems thinking is a discipline for seeing wholes. It is a framework for seeing interrelationships rather than things, for seeing patterns of change rather than static snapshots." Below is the some systems thinking definitions and important aspects summarized into few keywords that describes systems thinking.

Table 1.Systems Thinking Definitions and Key Concepts



Definitions of Systems Thinking	**Key Concepts**
"A system cannot be understood by analysis alone; its properties emerge only through the interactions of its parts." - Meadows, D.H., (2008)	Wholes, Interactions, Dynamic Relationships
"The whole is more than the sum of its parts." - Bertalanffy, L., (1968)	Emergent Properties, Interactions
"Systems thinking is a discipline for seeing wholes. It is a framework for seeing interrelationships rather than things, for seeing patterns of change rather than static snapshots." - Senge, P., (1990)	Wholes, Interrelationships, Patterns of Change
"Systems thinking is a way of thinking about, and a language for describing and understanding the forces and interrelationships that shape the behavior of systems." - Richmond, B. (2000)	Interrelationships, System Behavior
"Systems thinking is an approach to problem-solving that views 'problems' as symptoms of underlying systemic structures and behaviors." - Capra, F. (2002)	Systemic Structures, Behaviors, Underlying Causes
"The future of our world depends on our ability to see the larger picture and to make better connections between what we do and the broader consequences of our actions." - Wheatley, M.J., (2006)	Broader Consequences, Holistic Perspective
"Systems thinking is a holistic approach that focuses on understanding the interconnections and interdependencies within complex systems." - Ackoff, R.L.,(1979)	Interconnections, Interdependencies, Complexity, Holistic Perspective

2.2.1. Hard and Soft Systems Thinking

Hard systems thinking emphasizes the use of formal models, quantitative methods, and rigorous analysis to understand and solve well-defined problems within clearly defined boundaries. It involves breaking down complex systems into their constituent parts, defining clear objectives, and employing mathematical or computational models to analyze and optimize system behavior. Hence hard systems thinking is very applicable in well-defined technical problems.(Checkland, P., 2000)

Soft systems thinking acknowledges the complexity and diversity of human and social systems, focusing on understanding multiple perspectives and accommodating diverse stakeholders' views. It involves qualitative methods, dialogue and interviews, and participation to address complex, ill-defined problems that are open to interpretation and do not have easily measurable objectives. Hence soft systems thinking is more suitable for ill-defined situations which involve human beings and cultural, political views. (Checkland, P., 2000)

In the below Figure 5, hard systems thinking is built upon defined or accepted systems that need to be engineered, however soft systems thinking is trying to define systems that are tangled, ill-defined and complex and then looks for solution.

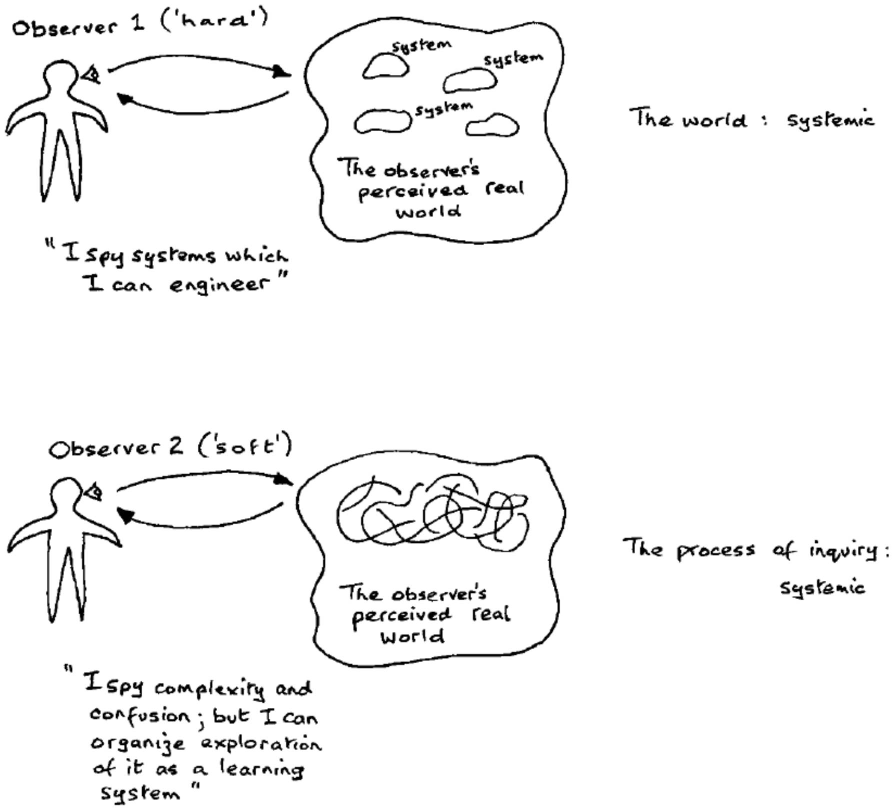

Observer 1 ('hard')
system
system
system
The observer's perceived real world
The world : systemic
" I spy systems which I can engineer "
Observer 2 ('soft')
The observer's perceived real world
The process of inquiry : systemic
" I spy complexity and confusion ; but I can organize exploration of it as a learning system "

Figure 5. Hard and Soft Systems Thinking (adapted from Checkland, P., 2000)

2.3. Soft Systems Methodology (SSM)

There are too many software development methodologies that can be grouped into agile and traditional methodologies which are concentrated on how the software should be developed.(Awad, M.A., 2005) In addition to that, there are lots of process improvement tools which are discussed above that are mostly quality-based such as Kaizen, Lean, Six sigma and etc. which are mainly focuses on improving operational efficiency, reducing waste and eliminating defects that depends heavily on statistical analysis and process engineering principles. (Rosa, A.C.M. & Broday E.E, 2018) On the other hand, SSM is not a system design tool, it questions what operations the system should operate and why it should operate. SSM provides an holistic approach to poorly defined and complex processes in order to improve them. Thus SSM provides "soft" approach (what system should do) that can be a precondition for "hard" approach (how system should do). (Gasson, S.,1994) The purpose of the soft systems methodology is depicted as below Figure 6.

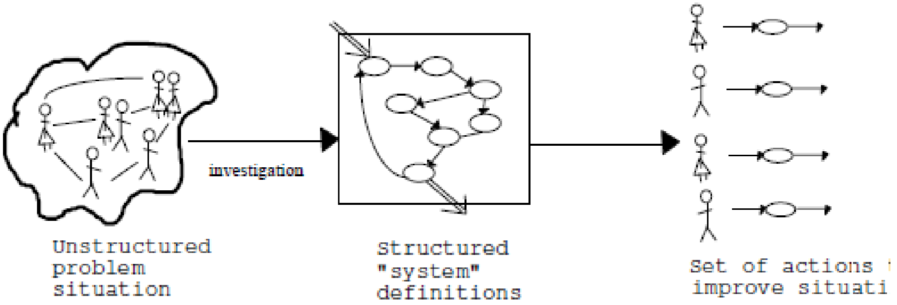

investigation
Unstructured problem situation
Structured "system" definitions
Set of actions
improve situati

Figure 6. The Purpose of Soft Systems Methodology (adapted from Gasson, S., 1994)

This demonstration is a broad explanation of SSM such that these set of actions or solutions can be seen as set of information requirements, however it is more useful to see them as a set of organizational process improvements where process can be defined as organizational tasks which are performed by human actors. (Gasson, S.,1994) Apart from this brief summary of SSM, according to Checkland (2000), there exists four main ideas that shaped the development of SSM.

1. The concept of modeling purposeful human activity systems involves viewing them as interconnected sets of activities that, when combined, have the potential to demonstrate the emergent quality of purposefulness.
2. Prior to initiating the modeling process, it is necessary to make deliberate and explicit choices. Given the multitude of potential human activity system models that can be constructed, the initial decision revolves around determining which ones are expected to be the most pertinent or illuminating in comprehending the current situation.
3. As the initial selection of a few models paved the way for probing the actual situation, generating new knowledge and perspectives on the problem at hand, additional ideas for relevant models emerged. This continuous learning process underscored the fact that the exploration and refinement of models remained an ongoing endeavor.
4. Models which are not ideal in every real world scenarios cannot provide an entry to work on information systems.

SSM models' development evolved over time and as it evolved, it gained a broader meaning. During this process, four models came to light which are Block &Arrows, Seven Stages, Two Streams and Four Main Activities respectively.(Checkland, 2000)

1. Block & Arrows method involves forming a diagram that consists of blocks representing different components of the system and arrows establishing the relationships or interactions between blocks.
2. Seven Stages method's first two stages contain defining the problem situation, finding out about it and expressing its nature. Third stage concerns about root definitions about considered problem in first two stages and modelling of this is done in stage four. Next stages use these models to structure the further questioning of the situation by comparisons and to define the changes which could improve the situation. Last stage takes the action to improve the problem situation. With this implementation, process allows to start over to seek for better improvements.
3. Two Streams method includes the constructing two models to represent the problem: Current Situation Model and Improved Situation Model. These models are used to explore the existing system and propose potential improvements
4. Four Main Activities method composes of four activities which are;
    a. Determining problem situation considering cultural and political aspects of the process
    b. Formulating some relevant purposeful activity models
    c. Analyzing the problem using created activity models, seeking from that debate both;
        i. Improvements about situation which are feasible and desirable
        ii. Adaptation between conflicting interest
    d. Taking action to improve the situation.

Soft system methodology comprises of seven stages. (Checkland, P.&Poulter J.,2006) SSM's conventional seven-stage process model is shown in below Figure 7.

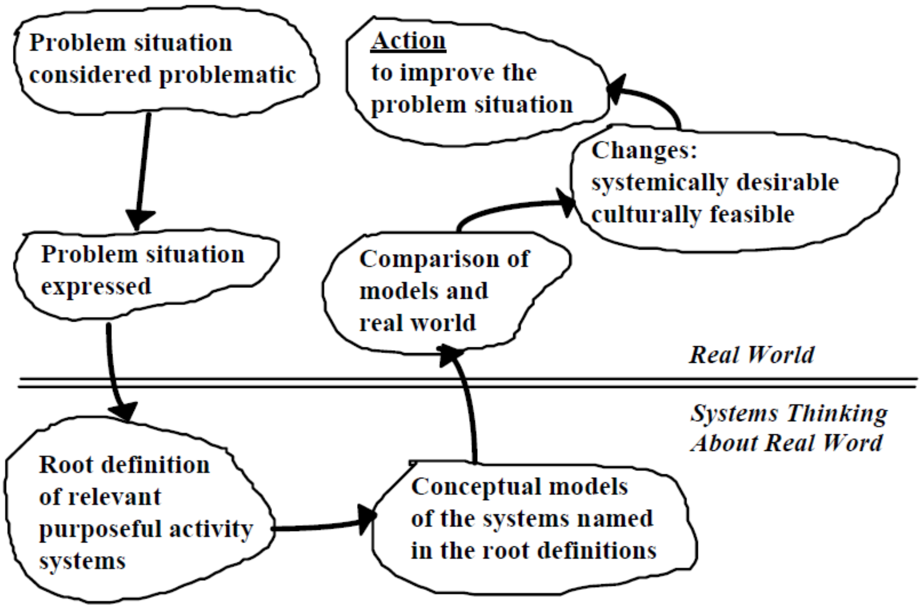

Problem situation considered problematic
Action to improve the problem situation
Changes: systemically desirable culturally feasible
Problem situation expressed
Comparison of models and real world
Real World
Systems Thinking About Real Word
Root definition of relevant purposeful activity systems
Conceptual models of the systems named in the root definitions

Figure 7. The Standard Seven Stage of SSM (adapted from Gasson, S., 1994)

Detailed explanations of each stage are given below.

Stage 1: Finding out about Problem Situation

Checkland and Scholes (1990) define analysts as "would-be improvers of the problem situation" who try to understand the processes as whole as far as possible. To explore context and content of the problem, interviews, observations, group meetings and workshops can be done. With this application, employees – which are organizational actors- can point out the problems they encounter and demonstrate their work to analysts. In this step, analysts main concern is to collect data about the process as much as possible. (Gasson, S., 1994)

Stage 2: Problem Situation Expressed

Checkland (2000) draws attention to rich picture drawing and making analyses one, two and three in this phase. Rich picture drawings are one of the tools that describes current processes. It makes analysts to think about factors that affects the problem situation and analysts should not limit themselves with the number of rich picture drawings, analysts should draw as many as they want to get better understanding of the problem. (Gasson, S.,1994) Checkland (2000) suggests analysts should draw or indicate three perspectives of the problem by doing analyses one,two and three.

Analysis one includes interventions and analysts investigate how the system operates, goals and objectives of the system and why is this operation is perfomed. With these gathered information, analyst can decide “what to leave and what to exclude” (Gasson, S., 1994)

Analysis two represents social context of the investigated problem. It searches relations between involved people and how do they interact with each other. (Gasson, S.,1994) According to Checkland. P. and Scholes, J. (1990), social context consists of three items which are roles, norms and values. Role is the social position of the people in the problem situation, a role has some expected behaviors that are called as norms and actual performance occurred with respect to determined values.

Analysis three is more focused on political context of the problem which deals with who has the power of the process, who has the power over whom or how is the power exerted throughout the process. (Gasson, S. 1994) Checkland, P. and Scholes, J. (1990) give examples about power used in processes as “Examples [of commodities] include: formal (role-based) authority, intellectual authority, personal charisma, external reputation, commanding access (or lack of access) to important information, membership or non-membership of various committees or less formal groups, the authority to write the minutes of meetings, etc.”

There are other tools that help analysts to help define problem situation in SSM. Problem diagram, which is developed by Professor Galliers, can be used for structuring ideas after a reasonable rich picture is obtained. It defines some system perspectives and looks for root definitions by using cause and effect.(Gasson, S., 1994) Figure 8 depicts a simple problem diagram.

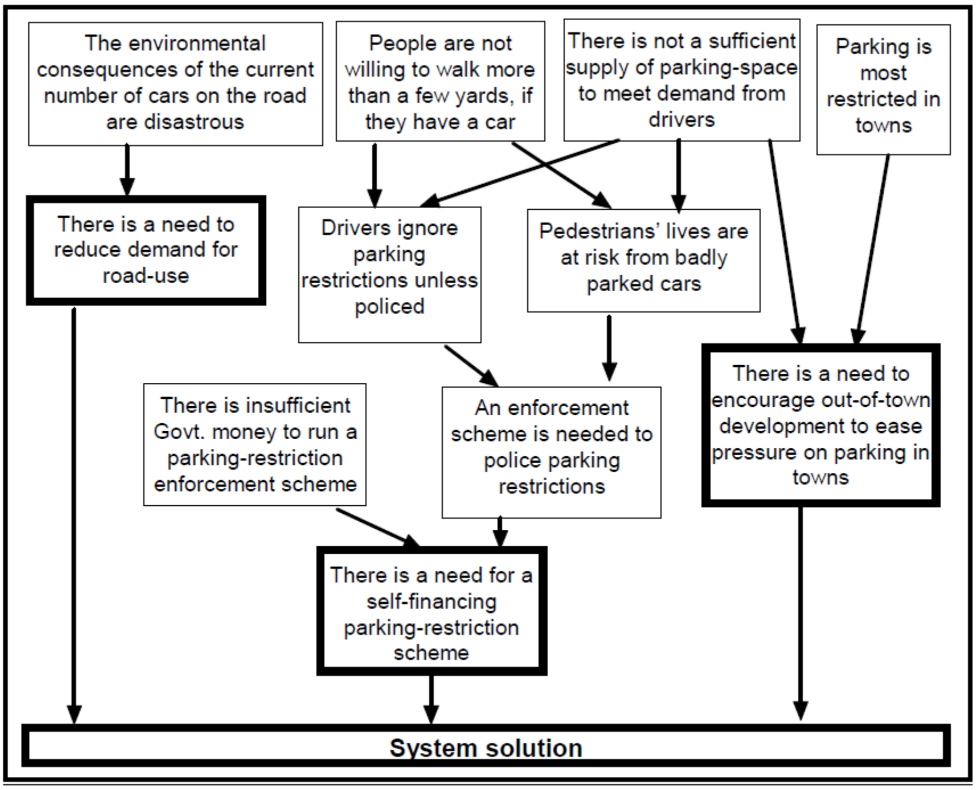

The environmental consequences of the current number of cars on the road are disastrous
People are not willing to walk more than a few yards, if they have a car
There is not a sufficient supply of parking-space to meet demand from drivers
Parking is most restricted in towns
There is a need to reduce demand for road-use
Drivers ignore parking restrictions unless policed
Pedestrians' lives are at risk from badly parked cars
There is a need to encourage out-of-town development to ease pressure on parking in towns
There is insufficient Govt. money to run a parking-restriction enforcement scheme
An enforcement scheme is needed to police parking restrictions
There is a need for a self-financing parking-restriction scheme
System solution

Figure 8. Problem Diagram Example (adapted from Gasson, S., 1994)

Analysis one, two and three requires an extensive process modelling. Process modelling provides instant access to current information, possible future reuse of available information and documentation managing. There are plenty of process models such as workflow models, BPMN, EPC, Petri nets,UML and etc. that can be used with respect to purpose of process modeler.(Cesare, S., 2011) Business Process Modelling Notation (BPMN) has become de-facto standard and defined as ISO standard by object management group. (ISO/IEC 19510:2013) BPMN is a process modelling language that does not require any technical knowledge (such as UML) with easy-to-use graphical representations for analysts. (Zarour, K. et al, 2020)

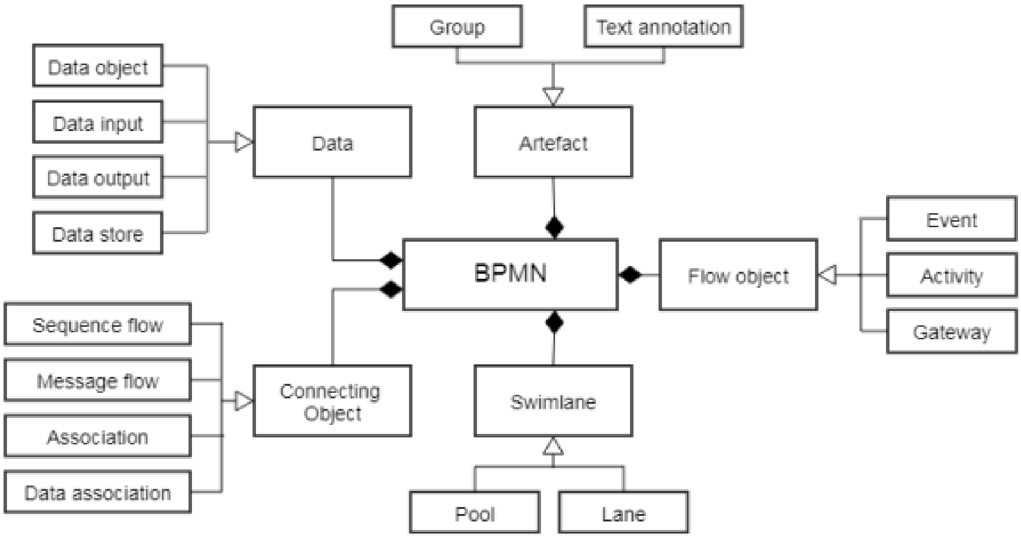


Figure 9. Meta Model of BPMN Elements (adapted from Zarour, K. et al, 2020)

Stage 3: Deriving Root Definitions

Checkland, P. (2000) states that building purposeful model requires clear definition of model. It is mentioned as “These definitional statements, SSM's `root definitions', are constructed around an expression of a purposeful activity as a transformation process T.” The root definition identifies the system such that it becomes obvious how the system is going to operate and what the system going to do. (Gasson, S., 1994) Gasson, S. (1994) proposes two approaches in order to define root definitions.

- Input-Output Diagrams: Deriving inputs and outputs of the system considering different perspectives of involved people, gives analyst a whole clear definition of the system. However, it is analyst’s role that which perspectives should be included or not.
- CATWOE analysis: When deriving root definitions, analyst considers below questions.
    - Customer: who is the system operated for?
    - Actors: which single group of people will perform the activities involved in the transformation process?
    - Transformation: what single process will convert the input into the output? The input and output must be those at the system boundary, which should be the same for all transformations defined.
    - Weltanschhaung: (this means worldview, in German) what is the view which makes the transformation worthwhile? (this has a lot to do with how you define success, but also states why the transformation process is being performed at all)

- Owner: who has the power to say whether the system will be implemented or not?
- Environment: what are the constraints (restrictions) which may prevent the system from operating?

Using these approaches, root definitions for each and every system can be defined with a clear understanding.

Stage 4: Building Conceptual Models

In previous step, it is indicated that "what defines success for the system", in this step, "what should be done by actors to evaluate or achieve that success" question becomes significant.(Gasson, S.,1994) Deriving a conceptual model requires two steps according to Gasson, S., (1994)

- listing all activities required to achieve the root definition of the system (each activity should begin with a verb)
- graphically relating the activities together, with monitor and feedback activities.

Stage 5: Comparison of Conceptual Models and Real World Activities

The aim of this stage is to provide set of prioritized recommendations for what changes can be done in existing system. Gasson, S. (1994) proposes several ways to compare real world activities with constructed conceptual models. It is exemplified in Gasson, S., (1994) as "the activities can be considered individually, with each activity compared to real life for its effectiveness and its links to other activities, or activity diagrams (like conceptual models, but for real-world activities) can be drawn and compared to the conceptual models."

Stage 6: Analyzing Feasible and Desirable Change

In this step, some changes are proposed in order to get more smooth process. These changes are proposed to stakeholders of the process and also they are evaluated by stakeholders. It is important to note that stakeholders are the ones who decides which change is feasible or not. (Gasson, S., 1994)

Stage 7: Taking Action

Gasson, S. (1994) mentions this stage as "[stage] involves the "politics" of the intervention. The actions predicated by the previous stages of analysis must be implemented in a way which avoids upsetting too many people, while still achieving the objectives which you set yourself, for the change." This stage takes action to implement reviewed changes in the system.

CHAPTER 3

RESEARCH METHODOLOGY

Most of the problems that are encountered throughout life involve human beings. Hence problems get complicated, unstructured and sometimes insolvable. Socio-cultural aspects of problems should be taken account of in order to identify problems. There can be many considerations such as perspectives of employees, relationship between them or rules and regulations of firm and government and behavior of employees when working on a problem. Thus for such a complex problems, soft systems methodology is a suitable choice when investigating root causes of problems and developing related improvements. This study uses Soft Systems Methodology in an ERP software development process within two organizations as main stakeholders.

3.1 Research Philosophy

Research philosophy refers to a framework to understand assumptions and beliefs that is caused by researcher's way of doing the study. Research philosophy helps researcher in decision making process throughout the study.

This research uses interpretive research which can be summarized as "interpretive research is an approach to qualitative inquiry that seeks to understand the subjective experiences and interpretations of the social world from the perspective of the individual or group being studied. It involves a deep engagement with the social and cultural context in which people live and work, and a commitment to exploring the multiple meanings and interpretations that shape human behavior." (Charmaz, 2014) Interpretive research is a subjective and contextual. As it is stated earlier, there is a need for an interpretation from the perspectives of the people which can be provided by interviews, meetings or face-to-face dialogues to understand and define problems.

3.2. Research Method

Research methods are the ways that researchers use with respect to research philosophy in order to collect, analyze and interpret data. The choice of research method depends on research question, data that will be collected and the aim of the study.

This study uses qualitative method which can be described as "an inquiry process of understanding based on distinct methodological traditions of inquiry that explore a social or human problem. The researcher builds a complex, holistic picture, analyzes words, reports detailed views of informants, and conducts the study in a natural setting." (Creswell, 2013) As stated, it considers views of participants, analyzes what is said or done for the specific situation. There are lots of qualitative methods techniques but in this research case study technique is used. Robert E.Stake (1995) defines case study as “ the study of the particularity and complexity of a single case, coming to understand its activity within important circumstances.” According to Stake R.E.(1995) in this study, our concerned company has a unique behavior such that it is half private half state-owned company which makes it more complicated than other companies. Moreover, the study cannot be generalized due to its endemic nature that is constrained by being an ERP software development company (as private and public-owned company) which is a subcontractor of a state-owned company. In research design part, used techniques are explained in more detail.

3.3. Research Design

The study searches for improvements or corrections in software development process with using interpretive research. Qualitative method provides related tools and techniques that can be used to collect and analyze data that come from interpretation. To do this case study technique is used in order to understand software development process.

3.3.1. Data Collection

Case studies are used in order to have deeper understanding and analysis of problems. Case studies involve in-depth analysis of a particular individual, group, or phenomenon. Case studies may involve multiple data sources, such as interviews, observations, and documents. (Yin, R. K.,2018). Hence, while investigating software development process, semi-structured interviews, observations, document examination and focus group studies are done.

Interviews involve asking open-ended questions to participants to explore their experiences, beliefs, and attitudes. The researcher may conduct interviews in-person, over the phone, or online.(Kvale, S.,1996) Semi-structured interviews are interviews that interviewer has some predetermined questions to lead the interview but also asks some open-ended and follow-up questions to have interviewee’s responses in detail. The interviewee can also ask questions in order to express thoughts freely and in greater depth.

Focus groups involve bringing together a group of participants to discuss a particular topic or issue. The researcher facilitates the discussion, and participants are encouraged to share their experiences, perspectives, and opinions. (Krueger, R. A., & Casey, M. A.,2014). Focus groups are used in order to discuss process related issues to observe different views, reactions and behaviors.

Observations involve systematically watching and recording the behavior of individuals or groups in a natural setting. The researcher may use structured or unstructured observation techniques. (Angrosino, M. V.,2007) The researcher also one of the stakeholders who works as an employee on software development process.

Hence, observing and using software development process was not difficult for the researcher. Researcher has also accessibility to documents that are used throughout the process. Document reviews and corrections have been made easily.

Interviews, observations and focus group studies are conducted with an ERP project group in subcontractor firm. One acquaintance meeting is held to discuss overall process and its sub-processes in general with technical and project managers and other employees. The topics that will be argued are defined. (see Appendix B) Then four other meetings are held for each sub-process with related employees such as designers, developers and team-leads.(see Appendix B)

According to semi-structured interviews with focus groups participants and observations, four case studies are determined and examined. These four case studies are four sub-processes of software development process which are design management process, technical development process, testing process and training process respectively.

3.3.2. Data Analysis

3.3.2.1. Systems Thinking Behavior of Processes

According to Checkland, P. & Scholes, J. (1990), a system can be defined as "a set of interconnected and interdependent elements or components that are organized in a coherent and purposeful manner to achieve a specific goal or function, and which is bounded by its environment. The system exhibits emergent behavior that arises from the interactions between its components, and is subject to feedback from its environment, which can be positive or negative. The system is more than the sum of its parts, and cannot be fully understood by analyzing its individual components in isolation." This explanation emphasizes importance of system characteristics such as interconnectedness of components, holism, boundary, feedback and emergence behavior. On this basis, software development process constitutes a system. And also each sub-process can be thought as smaller systems that interacts with each other. Each sub-process has their own components to work, has a certain boundary, components produces their outputs, some positive or negative feedback are given in order to correct behavior of the process, and each sub-process is more than the sum of their parts, it can produce completely different output when compared to individual components.

For each sub-process, deciding their components is vital and shows us the relation between them. Subsequent and preceding components give us information about how system works in a timely manner. To give an example, coding part of the technical development sub-process delivers developed code into the system.

Furthermore, determining boundaries for each sub-process is important, if it is not determined, matters, information and energy can cross the boundaries freely, hence researcher should decide the boundaries of the systems. For instance, technical development sub-process includes coding part and it can only be done in specified ERP solution environment.

Having a feedback mechanism provides systems to compare actual performance with desired performance and determines related modifications. With this mechanism, continuous improvement is realized. To exemplify that, testing sub-process has its own feedback mechanism in order to correct errors.

Moreover, emergence behavior of the system ensures the overall process produces desired output without considering what each component produces individually. For instance, each sub-process has their own outputs such as design management process should do design of the software that will be developed, technical development process should develop the code, testing process should test the developed process in order to have flawless process and training process should be ensure all employees have the related skills to do their job.

To sum up, each sub-process shows system characteristics and they form a bigger system which is software development system. In order to have smooth software development process, its parts should work properly.

3.3.2.2 Soft Systems Methodology

There are lots of systems thinking methods such as hard systems thinking, soft systems thinking, system failures thinking, system dynamics (Forrester, J. W.,1961),viable systems model (Beer, S.,1981) and etc. This study applies soft systems methodology based on soft systems thinking.

Soft systems thinking is problem-solving methodology that can be applied to any complex and ill-structured situations such that any linear and structured approaches are not appropriate for dealing with these complicated problems. Soft systems thinking highlights the stakeholder’s views and their experiences on problems and uses some techniques to understand problems.

Soft systems methodology has seven-stage implementation phases. In first step, SSM explores problem situation with all aspects. To do this, SSM collects data that includes multiple perspectives of stakeholders, observes problem situation and makes interviews in order to gain deeper understanding. After having related data, SSM tries to analyze the problem with using rich pictures, current system demonstrations with BPMN and process explanations. In next step, root definition of the process is identified and CATWOE rule is applied. Then, conceptual models are drawn in order to demonstrate what might improve or correct current problematic situation. After that, conceptual models are compared with real world applications in order to detect improvements that are needed. When these are realized, some changes are proposed to eliminate errors in the situation. Lastly, appropriate proposed changes are put into real life practices. (refer section 2.3)

After meetings that are conducted, it is realized that sub-processes mostly related with job titles, only testing and training processes are multidisciplinary, but even so each testing and training processes' steps does not have more than one job title, hence job title is a very distinguishable attribute in sub-processes. When evaluating topics these job titles are considered with this regard.

3.4. Trustworthiness and Triangulation

Trustworthiness refers to findings and conclusions of the study is reliable and accurate. Trustworthiness is a measure of quality of the study and it is vital that study is taken seriously and study has some important impacts. There are some methods that measures quality of the qualitative research such as Patton (2015) recommends using a logic model to guide the research process. A logic model is a visual representation of the relationships between the research question, the research methods, and the expected outcomes. Using a logic model can help to ensure that the research is focused and coherent. Lincoln and Guba (1985) recommend peer debriefing as one of the ways to enhance the credibility, transferability, dependability, and confirmability of qualitative research. According to them, peer debriefing provides an opportunity for researchers to receive feedback on their research design, data collection, and analysis methods. It also helps to ensure that the research findings are grounded in the data and not influenced by the researcher's biases or assumptions.

According to Lincoln and Guba (1985), credibility is established through the use of techniques such as prolonged engagement, triangulation, and member checking. Prolonged engagement involves spending enough time in the field to develop a deep understanding of the research topic and the participants' experiences. Triangulation involves using multiple sources of data and methods to validate the findings, while member checking involves sharing the research findings with participants to ensure their accuracy and validity. In this study, triangulation is provided by semi-structured interviews, observations, face-to-face meetings, document reviews and ERP system review. Prolonged engagement is ensured by research is done in researcher's workplace, observations and face-to-face meetings are held anytime needed. Research findings and proposed changes are shared with participants for member checking purposes. However, cause of the some confidentiality issues in organizations, some details of data cannot be shared in this study.

Transferability can be established through the use of techniques such as thick description and the use of theoretical sampling according to Lincoln and Guba (1985). Thick description involves providing rich, detailed descriptions of the research context and the participants' experiences, which help readers understand the study's findings in different contexts. Theoretical sampling involves selecting participants and data sources based on their relevance to the research question and theoretical framework, which can increase the generalizability of the findings. In this research, drawing current processes with SSM's rich pictures and writing them helps to understand the process. Participants are the stakeholders of software development process and data sources are the ones at triangulation part. Hence, this SSM method can be applied to any software development process and findings of this study can be used in different software development process.

Dependability can be established through the use of techniques such as audit trails and peer debriefing with respect to Lincoln and Guba (1985). Audit trails involve documenting the research process and decisions made at each stage of the study, which can help ensure that the findings are reproducible and consistent over time. Peer debriefing involves seeking feedback from other researchers or experts in the field, which can help identify any biases or assumptions that may have influenced the findings. This study includes some meeting notes in Appendix B. however due to some security issues, some of the information related to organization are omitted from the notes. Moreover, participants gave feedback about processes when arguing in face-to-face meetings.

Confirmability ensures that the research study is based on data collected rather than the assumptions of the researcher. In this research, meetings notes are summarized in order to give important problematic topics and processes. (see Appendix B)

3.5. Ethical Considerations

"Researchers must take into account ethical considerations when conducting research, including protecting the privacy and confidentiality of participants, obtaining informed consent, minimizing harm, and ensuring that the benefits of the research outweigh the risks. Ethical considerations should also be taken into account in data collection and analysis, dissemination of results, and use of research findings. Failure to address ethical considerations can result in harm to participants, damage to the research enterprise, and loss of public trust in research" (American Psychological Association, 2017, p. 2).

In this study, some ethical issues are considered in order to protect confidentiality of participants and organizations. The names of participants and organizations are not shared in this study. Gathered data and meeting notes are shared in this research in order to analyze processes.

CHAPTER 4

EMPIRICAL FINDINGS: CASE STUDY APPLICATION

4.1. Software Development Process

Software development process begins with a need from customer followed by designing the process, developing the code, testing implementations and deploying to live system. This software development process is mainly takes place on an ERP system. The basic rich picture of software development process for organizations is depicted below in Figure 10.

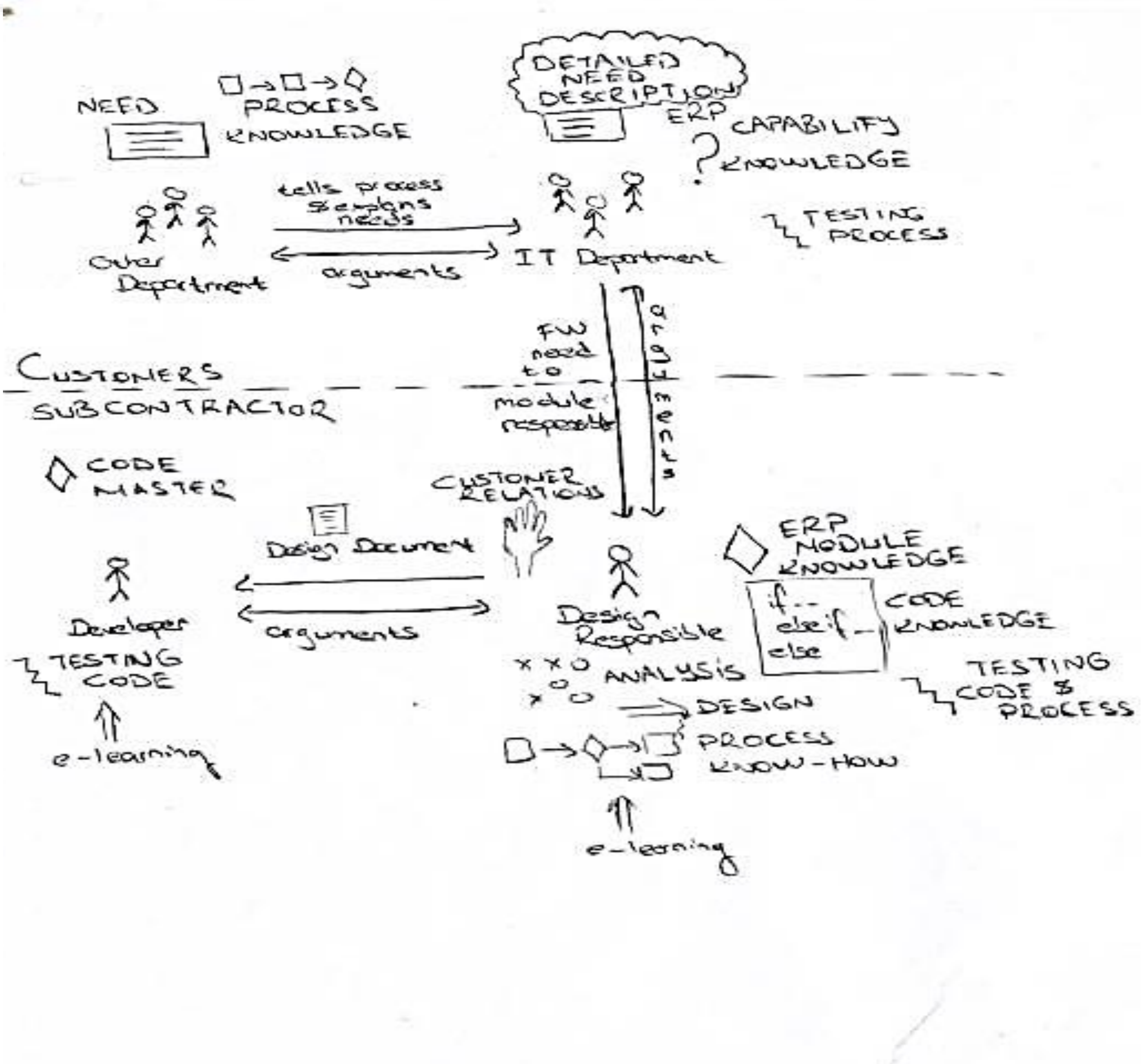


Figure 10. Basic Rich Picture of Software Development Process

In order to ensure Checkland's analysis two and three, the basic organizational roles that are affected by software development process are indicated below in Figure 11. For the sake of confidentialty of these two firms, not all of the departments are shown in the figure.

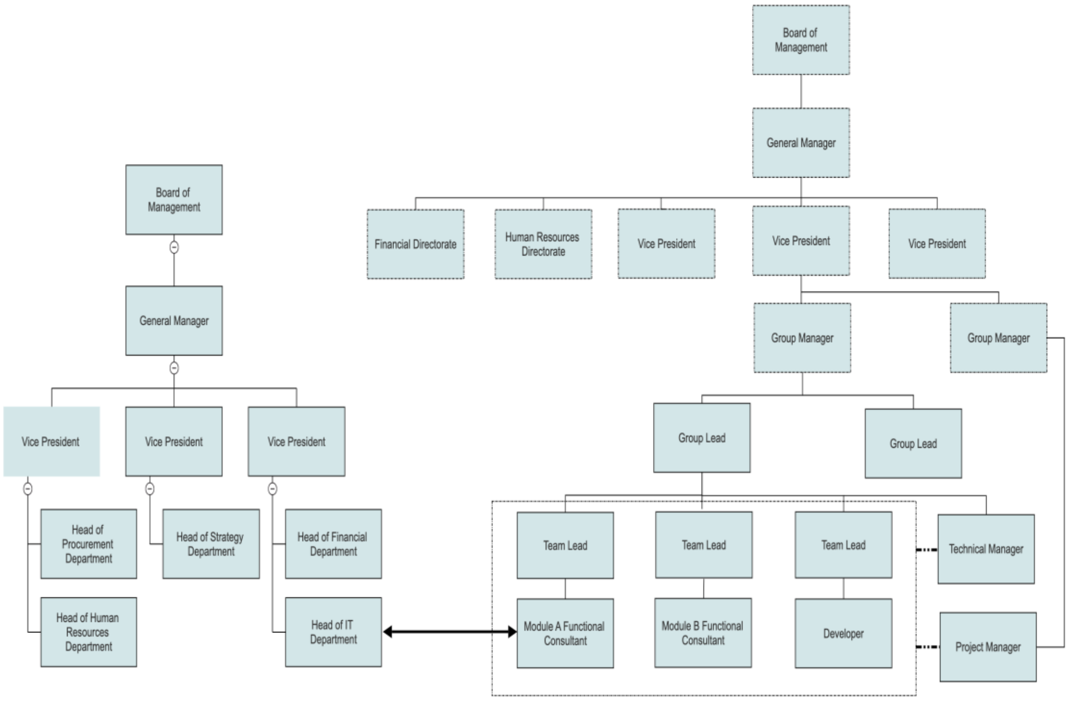


Figure 11. Organizational Charts and Relations of Two Firms

As it's seen from organizational structures of two firms in Figure 11 and from the rich picture that is drawn in Figure 10, development of a software needs two different firms with different level of relationships between or within firms. Right-one of the company works as a subcontractor of other employer company. Subcontractor company who works closely with IT Department, responsible for designing process, making relevant developments, solving problems and implementing them to live system with the information it receives from IT Department. Subcontractor company has a matrix structure whereas other company has a hierarchical structure as it is seen.

In Figure 10, basic rich picture of software development process, client comes with a need and passes over to its IT-department. The IT department pre-analyses the need and hands to other firm's designer. Designer analyses the need in detail, asks questions to relevant persons, forms design document and forwards to developer. Developer goes over to document and asks questions about design if any, then writes the code and makes tests. If everything goes well, developer returns to designer for process testing. Designer tests and if everything goes well, designer returns to IT-Department so that process can be tested again. If customer says it’s ok, developed process can be used in live system.

In order to achieve better process flow, soft software methodology is applied.

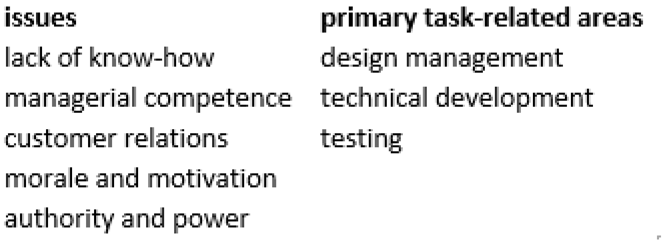
4.2. Stage 1: Finding out about Problem Situation

To define problematic issues in the process, data are collected from meetings that are constituted by stakeholders of the software development process which have roles as developers, designers and client.

In this study, group interviews and face-to-face meetings assist to gather information to enhance software development process considering participants’ roles, duties and relations with each other. Regarding to soft software methodology, this study is important due to participants ’worldviews. Studying relations within involved people in software development process helped to identify problematic areas in a social and economical context more clearly. Hence, it affects to describe next SSM steps in which begins with defining issues and primary task-related areas by using rich pictures after that identifying Root Definition which forms a basis for Conceptual Models. Wrong or lacking parts of software development process come to light with Conceptual Models, moreover, Conceptual Models provide information about how to improve the process. Lastly, proposed improvements can be developed, however, improvements of the software development process is out of this research study.

4.3. Stage 2: Analysis

As a result of these meetings, face-to-face interviews and rich picture of software development process, relations between involved people and departments in software development process are visualized as organizational chart in the above Figure 11 and primary task-related areas have been identified, moreover some problematic issues derived. Issues and primary task-related areas are depicted below Figure 12.



issues	primary task-related areas
lack of know-how	design management
managerial competence	technical development
customer relations	testing
morale and motivation	
authority and power	

Figure 12. Issues and Primary Task Related Areas

According to identified primary task-related areas, the software development process composes of three main processes as indicated in Figure 12 which are ‘Design Management’, ‘Technical Development’ and ‘Testing’ respectively. There is a ‘Training Process’ which can be counted as supporting process.

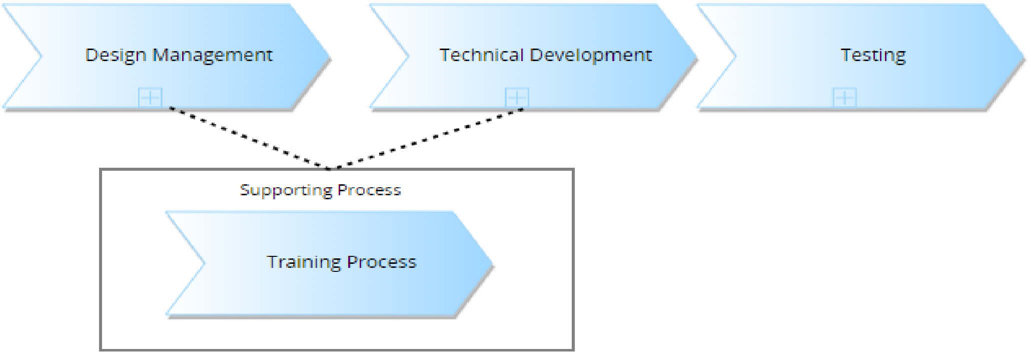


Figure 13. Sub-Processes of Software Development Process

Main processes and supporting process are expressed and examined in detail to identify problematic areas and define root definition.

4.3.1. Design Management

To gain more insights about design process, meetings are arranged with designers and customer separately. Detailed information is gathered about design management process (see Appendix B) Rich picture of requirements process is drawn (see Figure 14) and then value chain and BPMN diagrams of design process are depicted below figures to give more detail about people’s roles and flow of information.

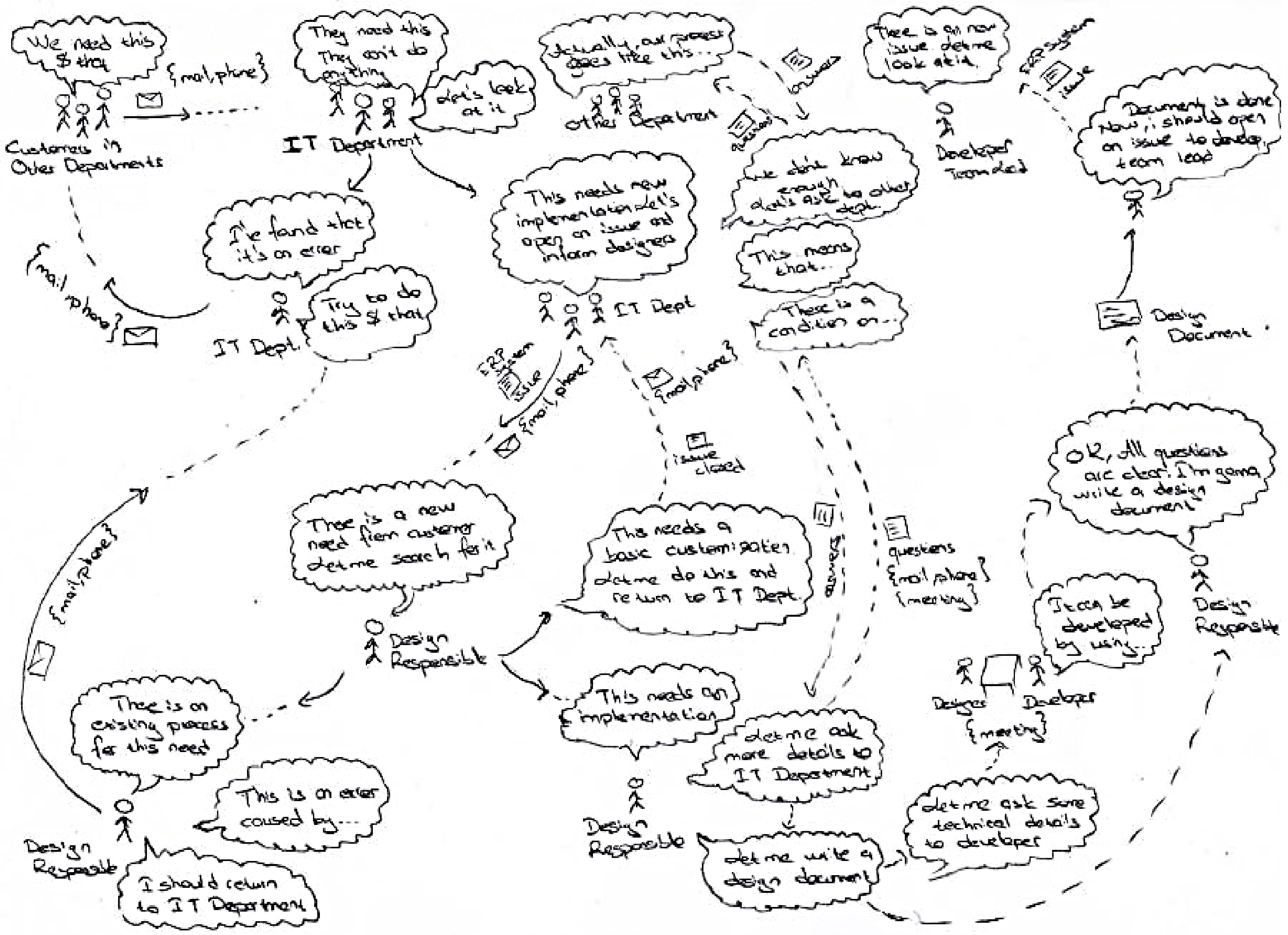

We need this $ that
{mail, phone}
Customers in Other Departments
They need this They can't do anything
Let's look at it
IT Department
Actually, our process goes like this...
Other Department
answers
questions
There is an new issue Let me look at it.
Developer Team Lead
ERP System issue
Document is done Now, i should open an issue to develop team lead
We found that it's an error
Try to do this $ that
IT Dept.
{mail, phone}
This needs new implementation. Let's open an issue and inform designer
IT Dept
We don't know enough details ask to other dept.
This means that...
There is a condition on...
Design Document
ERP System issue
{mail, phone}
{mail, phone}
issue closed
There is a new need from customer Let me search for it
Design Responsible
This needs a basic customization Let me do this and return to IT Dept.
answers
questions {mail, phone} {meeting}
OK, All questions are clear. I'm going write a design document
Design Responsible
{mail, phone}
There is an existing process for this need
This is an error caused by...
Design Responsible
I should return to IT Department
This needs an implementation
Design Responsible
Let me ask more details to IT Department
Let me write a design document
It can be developed by using...
Designer
Developer
{meeting}
Let me ask some technical details to developer

Figure 14. Rich Picture of Design Process

The design process begins with customers' needs. Other departments apart from IT-department state their needs or what they cannot do to IT department of the employer firm. IT department investigates and if they can solve it in a way, they return to customer without bothering designers. However, if they cannot solve it or if the need is something new, they describe what can be done or get in contact with designers to find a solution.

IT-department opens an issue in ERP system to designer with respect to their modules (the field of work), expertise and workload of them. Within issue, the need is described with non-technical words.

When the need arrives to designer with mail, phone and related opened issue, first of all, designer changes issue's statue from open to processing and designer may solve the problem directly, or prepares a design document to developer or if the need is unclear, designer argues with IT-department for what is meant, what is needed exactly. And when the need is understood designer either prepares design specification document or argues with developer to learn how many ways to develop this need and after that design specification document is prepared. If available, designer also includes test data into design specification document.

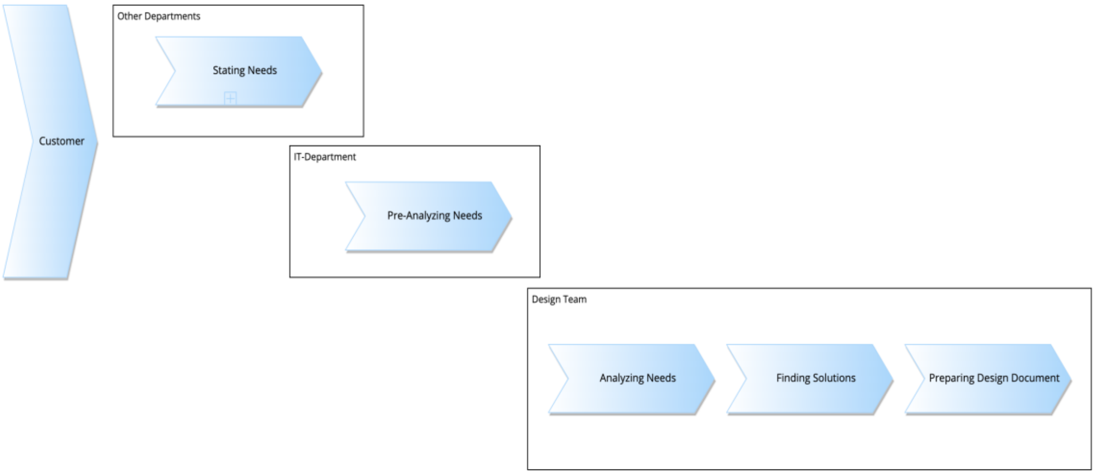


Figure 15. Value Chain Diagram of Design Management Sub-Process

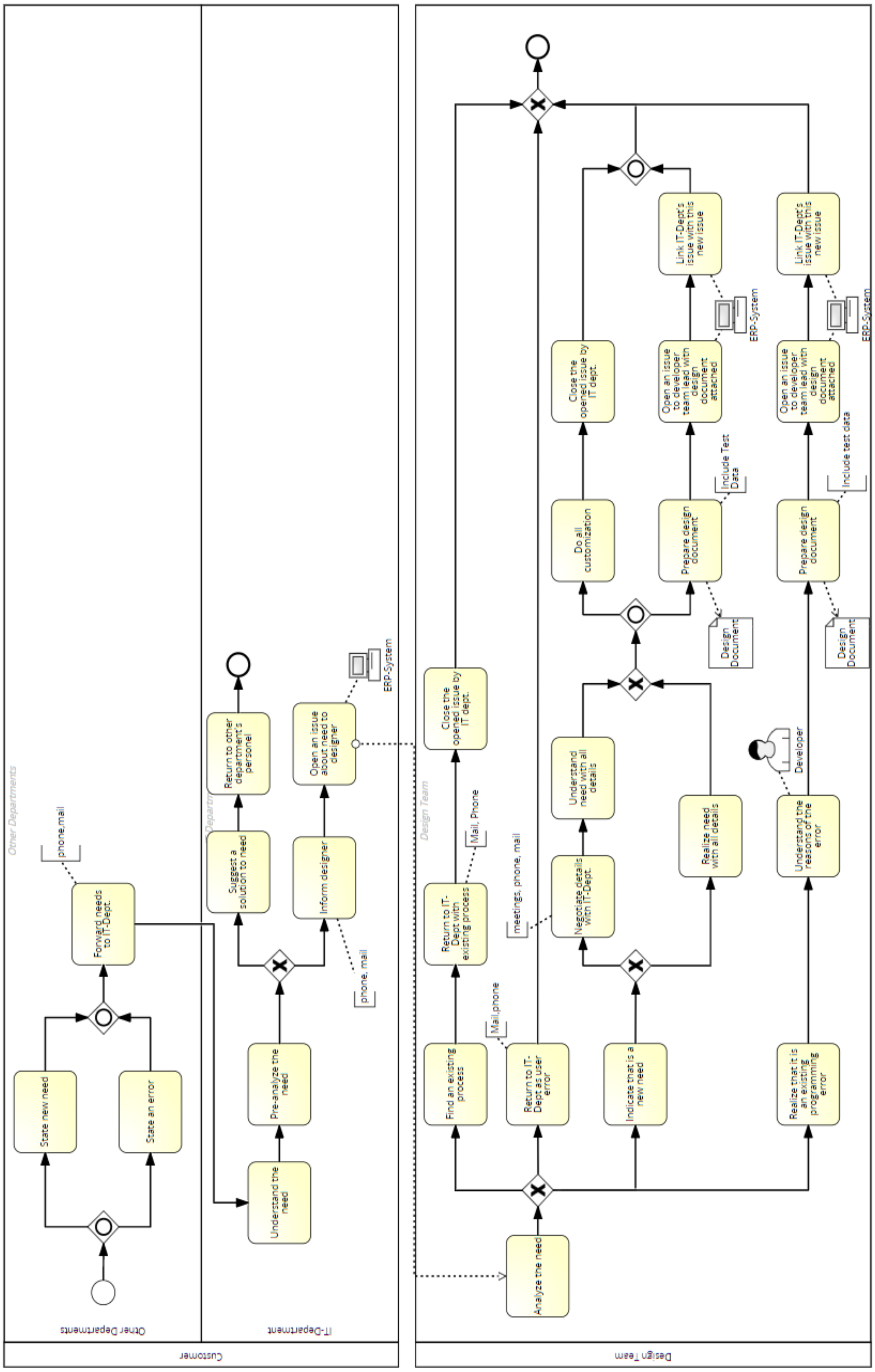

Customer
Other Departments
IT-Department
Design Team
State new need
State an error
Forward needs to IT-Dept.
phone,mail
Understand the need
Pre-analyze the need
Suggest a solution to need
Return to other department's personel
Inform designer
phone, mail
Open an issue about need to designer
ERP-System
Analyze the need
Find an existing process
Return to IT-Dept with existing process
Mail, Phone
Close the opened issue by IT dept.
Return to IT-Dept as user error
Mail,phone
Indicate that is a new need
Negotiate details with IT-Dept.
meetings, phone, mail
Understand need with all details
Realize need with all details
Do all customization
Close the opened issue by IT dept.
Prepare design document
Design Document
Include Test Data
Open an issue to developer team lead with design document attached
ERP-System
Link IT-Dept's issue with this new issue
Realize that it is an existing programming error
Understand the reasons of the error
Developer
Prepare design document
Design Document
Include test data
Open an issue to developer team lead with design document attached
ERP-System
Link IT-Dept's issue with this new issue

Figure 16. BPMN of Design Management Sub-Process

While or after developing the code, there may some addition requests or corrections from IT-department, which may come from other departments or not, that are analyzed again from designer and added to design specification document.

4.3.2 Technical Development Process

According to meetings and group interviews with developers, technical development sub-process is drawn and related information is gathered (see Appendix B).The rich picture of developer group is given in Figure 17.

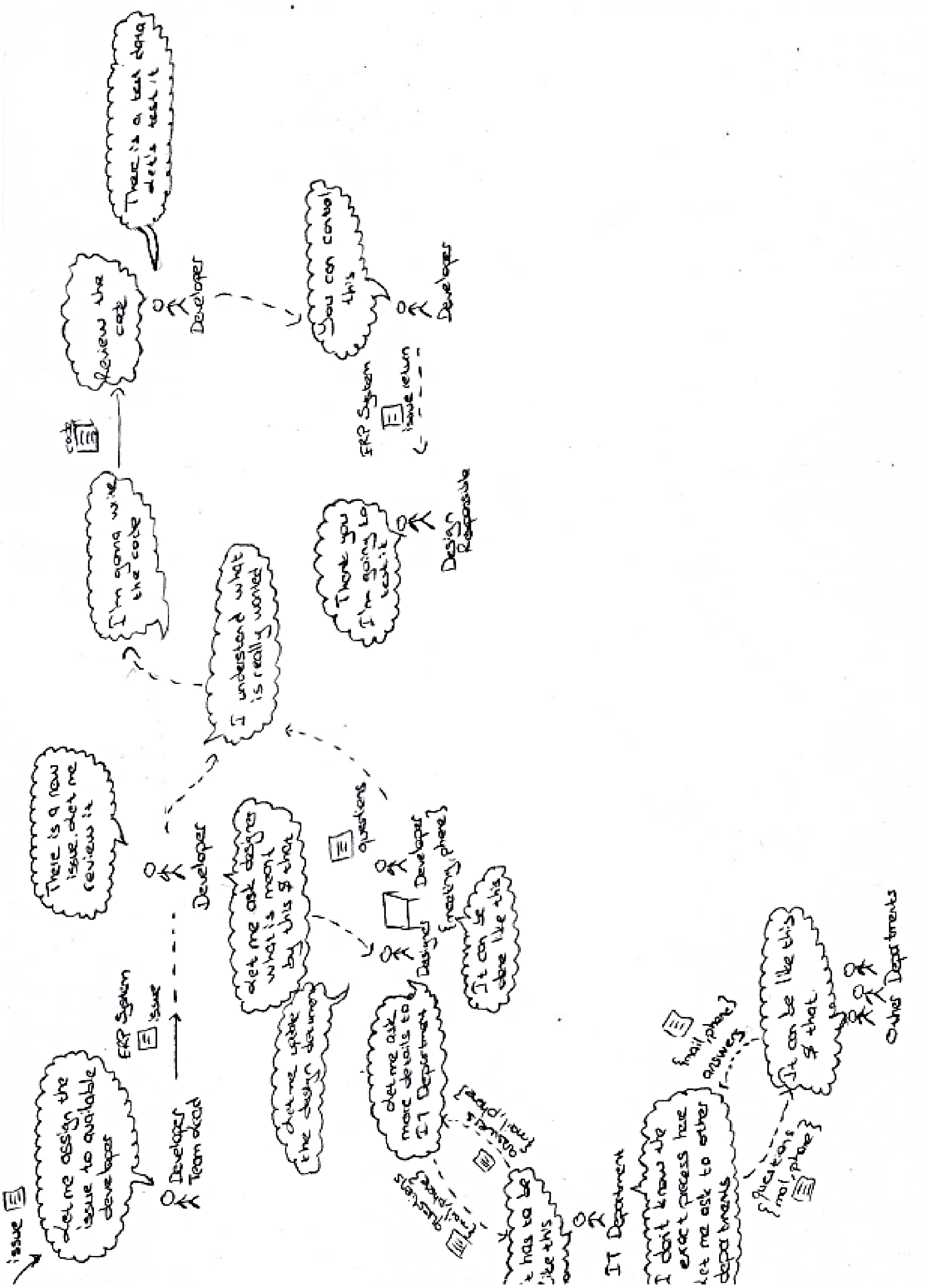

issue
Let me assign the issue to available developer
Developer Team Lead
ERP System
issue
There is a new issue. Let me review it
Developer
Let me ask designer what is meant by this $ that
Let me update the design document
questions
Designer
Developer
{meeting, phone}
It can be done like this
Let me ask more details to IT Department
questions {mail, phone}
answers {mail, phone}
t has to be ike this
IT Department
I don't know the exact process here. Let me ask to other departments
{mail, phone} answers
questions {mail, phone}
It can be like this $ that.
Other Departments
I understand what is really wanted
I'm going write the code
code
Review the code
There is a test data. Let's test it
Developer
You can control this
Developer
ERP System
issue return
Thank you I'm going to test it
Design Responsible

Figure 17. Rich Picture of Technical Development Sub-Process

The activities of sub-process are depicted in value chain and BPMN diagrams in Figures 16 and 17.

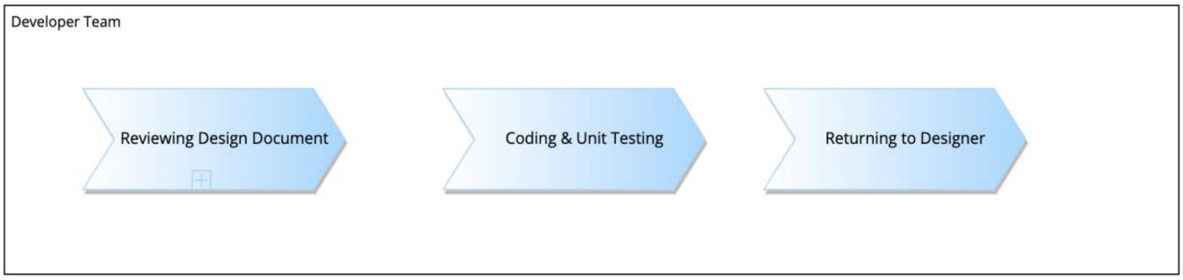


Figure 18. Value Chain Diagram of Technical Development Sub-Process

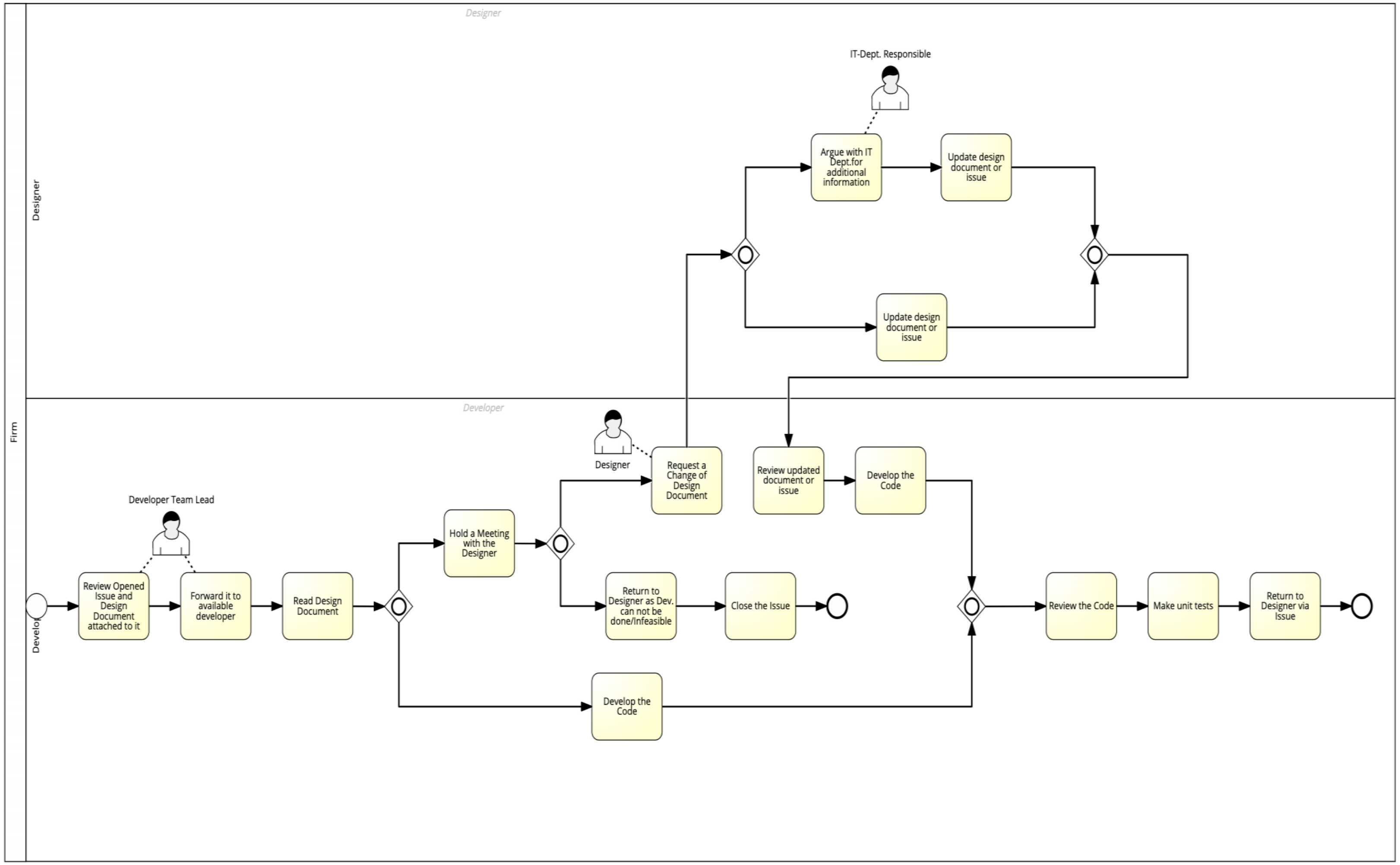

Firm
Designer
Designer
Developer
Develop
IT-Dept. Responsible
Argue with IT Dept.for additional information
Update design document or issue
Update design document or issue
Developer Team Lead
Review Opened Issue and Design Document attached to it
Forward it to available developer
Read Design Document
Hold a Meeting with the Designer
Designer
Request a Change of Design Document
Review updated document or issue
Develop the Code
Return to Designer as Dev. can not be done/Infeasible
Close the Issue
Develop the Code
Review the Code
Make unit tests
Return to Designer via Issue

Figure 19. BPMN of Technical Development Sub-Process

Technical development sub-process begins with an opened issue with software specification document followed by reviewing the document, arguing with designer for misunderstood parts, writing and testing code, returning completed development to designer respectively. In this process, there can be continuous communication between developer and designer for better understanding of the developed process, written code and test data.

Test data and specification document are derived from design management sub-process. While preparing specification document, designer may or may not ask a help from developer. While developing code with respect to specification document, developer can have additional questions. Developer arranges a meeting or makes a phone call with designer. Designer may or may not answer developer’s questions. If designer doesn’t know the answer or there is a gap in the process, designer asks to IT-department. IT-department gives related information with the help of other departments to designer. When all information is gathered, designer re-analysis the process, informs developer and updates the specification document.

After development of code is done, developer reviews and tests the code and corrects mistakes if available Then developer returns issue as functional test to designer. If any mistakes or additions return from designer, developer makes related corrections in the code.

4.3.3. Testing Process

Group interviews are set with all stakeholders of software development process, cause developers, designers and customers are responsible for testing newly implemented software process (see Appendix B). Hence, testing sub-process is constructed and related rich picture is drawn in Figure 4.3.3.1 according participants’ worldviews.

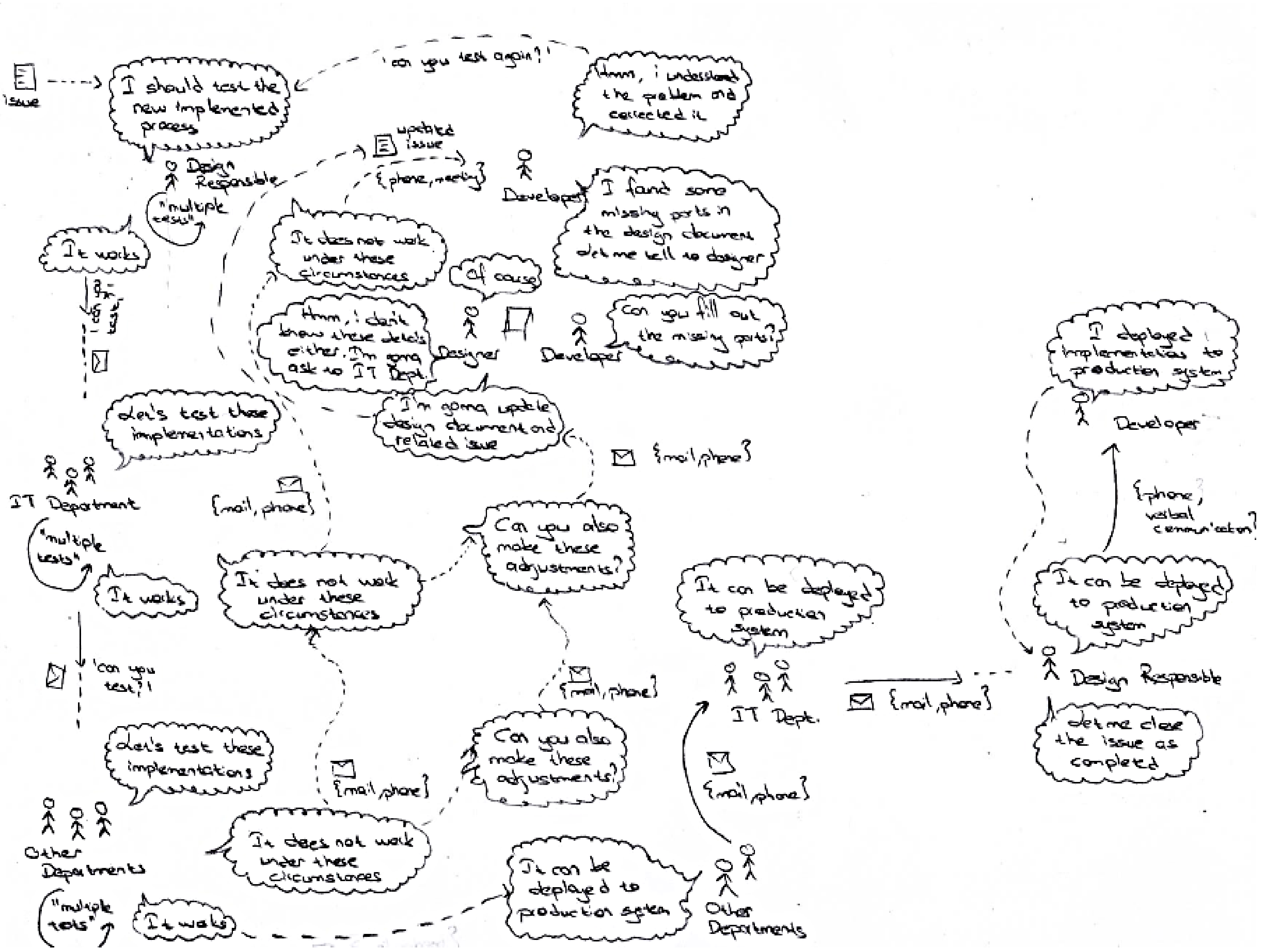

issue
I should test the new implemented process
Design Responsible
"multiple tests"
It works
can you test again?!
Hmm, i understood the problem and corrected it
updated issue
{phone, meeting}
Developer
I found some missing parts in the design document let me tell to designer
It does not work under these circumstances
Of course
Hmm, i don't know these details either. I'm gona ask to IT Dept.
Designer
Developer
Can you fill out the missing parts?
I'm going update design document and related issue
{mail, phone}
let's test these implementations
IT Department
"multiple tests"
It works
{mail, phone}
It does not work under these circumstances
Can you also make these adjustments?
can you test?!
let's test these implementations
{mail, phone}
Can you also make these adjustments?
{mail, phone}
Other Departments
It does not work under these circumstances
"multiple tests"
It works
It can be deployed to production system
Other Departments
{mail, phone}
It can be deployed to production system
IT Dept.
{mail, phone}
I deployed implementation to production system
Developer
{phone, verbal communication}
It can be deployed to production system
Design Responsible
let me close the issue as completed

Figure 20. Rich Picture of Testing Sub-Process

Related activities of testing sub-process visualized in below diagrams.

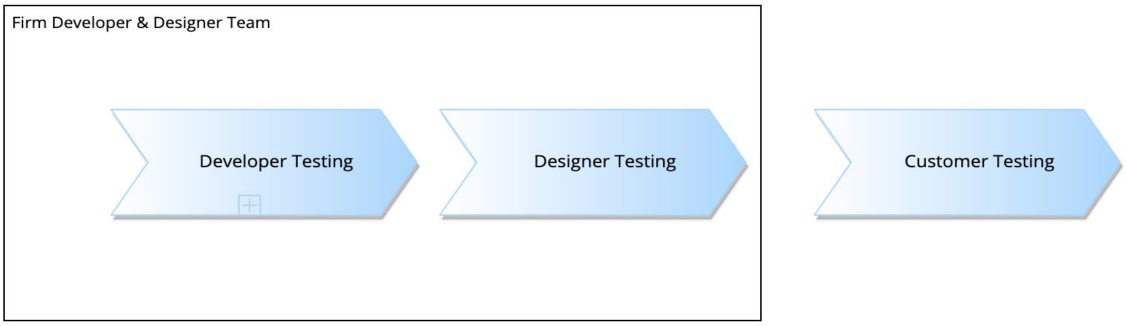


Figure 21. Value Chain Diagram of Testing Sub-Process

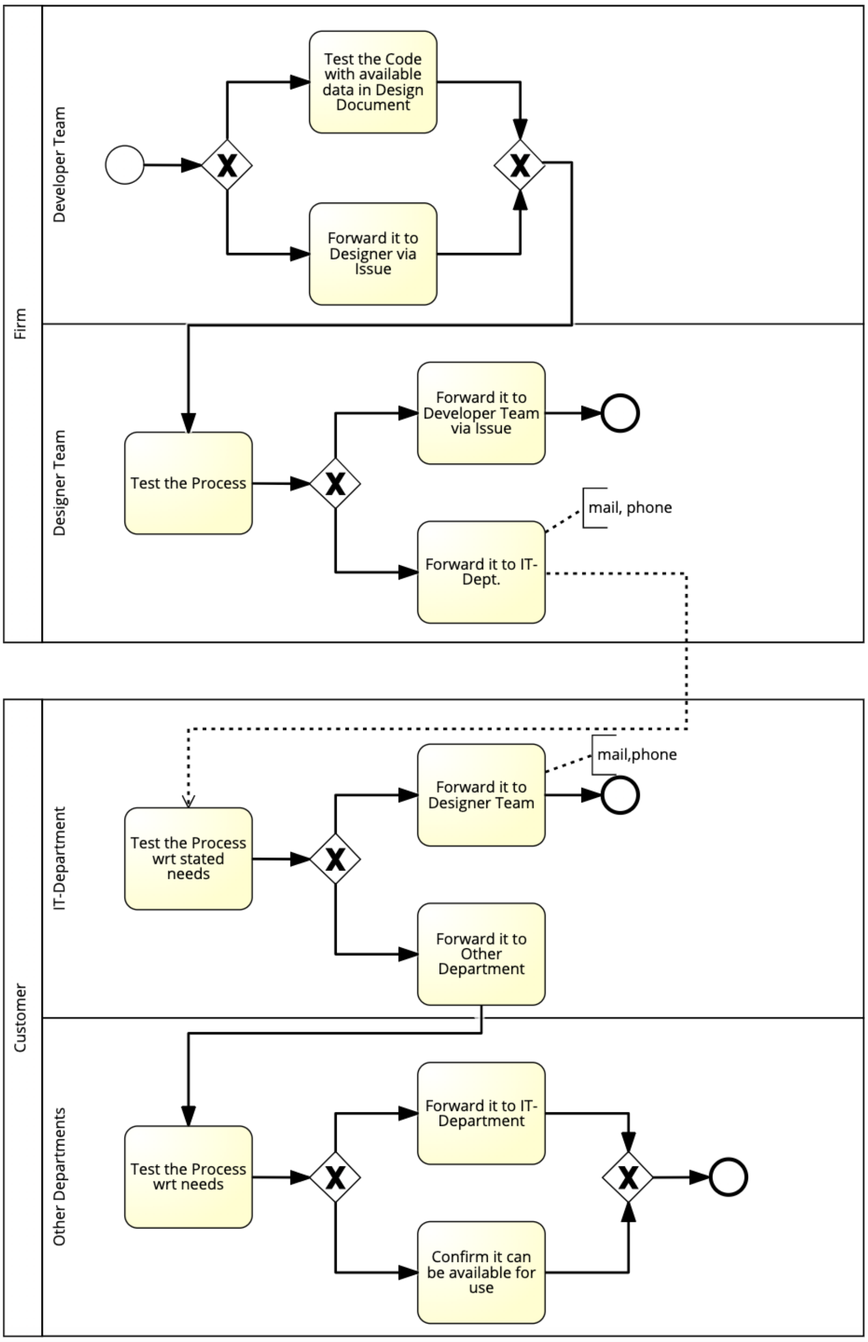

Firm
Developer Team
Test the Code with available data in Design Document
Forward it to Designer via Issue
Designer Team
Test the Process
Forward it to Developer Team via Issue
mail, phone
Forward it to IT-Dept.
Customer
IT-Department
Test the Process wrt stated needs
Forward it to Designer Team
mail,phone
Forward it to Other Department
Other Departments
Test the Process wrt needs
Forward it to IT-Department
Confirm it can be available for use

Figure 22. BPMN of Testing Sub-Process

Testing sub-process' activities begins with developer's unit testing if test data is available in specification document. If testing data is not available in specification document and developer has finished code review, developer returns to designer with completed development of related specification document. The moment that designer sees the returned issue, designer starts to test the process. If designer finds any mistakes, mistakes are returned to developer with proper explanations of test scenarios through related issue. Then developer corrects mistakes and returns to designer again. This process testing between designer and developer may occur many times until designer is sure about there is not any mistakes. After that, designer sends implemented process to responsible people of IT-department via mail or phone. Responsible people of IT-department control the implemented process and returns to designer if any mistakes or any new additions to current developed process. Designer re-examines the mistakes or new needs, and if needed, designer returns to developer with cases that mistakes occur or new additions on the same issue. Testing process between developer and designer occurs again Whenever designer is sure about new additions or corrected mistakes, designer again returns to IT-department. And above testing process between IT-department and designer may occur more than once. After all is done, IT-department returns to responsible people of other departments for testing. If testing of other customers is successful, developed software part goes on live, otherwise IT-Department tests again and if mistakes still occur IT-department returns to designer and above testing processes between IT-Department –designer and designer – developer can repeat many times.

4.3.4 Training Supporting Process

Training process is considered as supporting process with respect to opinions of focus group participants. Training and learning are only be done by subcontractor firm. Employer firm had not given any training for a while, so it only relies on past experience. Training is done via e-learning sources by designers, developers, team leads, group leads and technical managers. Because of it is a supporting process, it does not have any strict tie with other processes. If an employee wants to develop herself, basically go online and search for topics that she feels she lacks of. Below is the rich picture of training process.

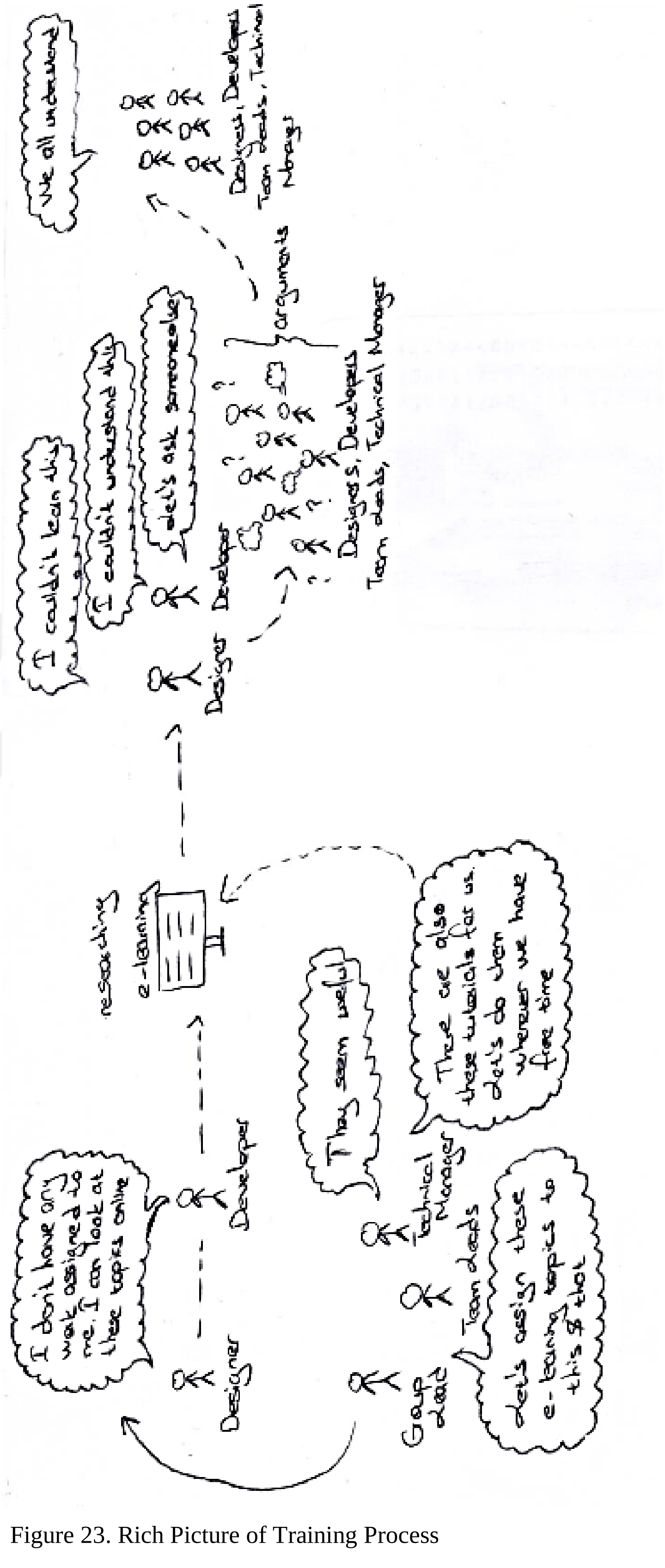

I don't have any work assigned to me. I can look at these topics online
Designer
Developer
They seem useful
There are also these tutorials for us. Let's do them whenever we have free time
Technical Manager
Team Leads
Group Lead
Let's assign these e-learning topics to this & that
researching
e-learning
Designer
Developer
I couldn't learn this
I couldn't understand this
Let's ask someone else
?
?
?
?
Designers, Developers, Team Leads, Technical Managers
arguments
We all understand
Designers, Developers, Team Leads, Technical Managers

Figure 23. Rich Picture of Training Process

Value chain diagram of training process is described in below Figure 22.

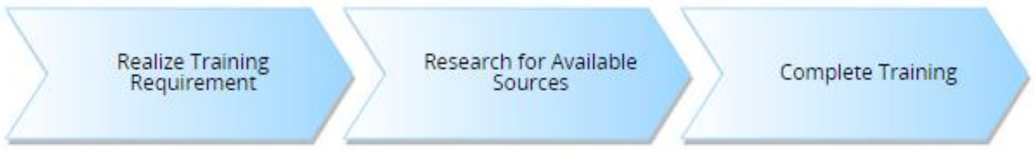


Figure 24. Value Chain Diagram of Training Process

Training process flows very individualistic. Below is the training process' BPMN.

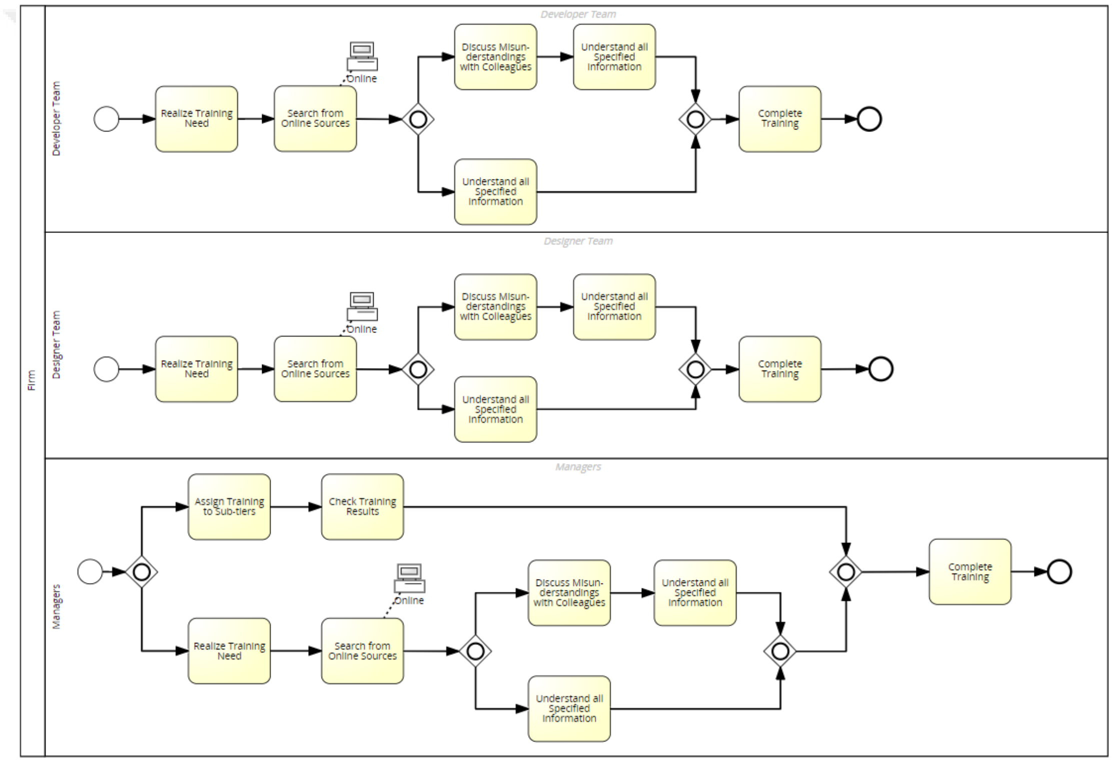

Firm
Developer Team
Realize Training Need
Search from Online Sources
Online
Discuss Misun-derstandings with Colleagues
Understand all Specified Information
Understand all Specified Information
Complete Training
Designer Team
Realize Training Need
Search from Online Sources
Online
Discuss Misun-derstandings with Colleagues
Understand all Specified Information
Understand all Specified Information
Complete Training
Managers
Assign Training to Sub-tiers
Check Training Results
Realize Training Need
Search from Online Sources
Online
Discuss Misun-derstandings with Colleagues
Understand all Specified Information
Understand all Specified Information
Complete Training

Figure 25. BPMN of Training Process

4.4. Stage 3: Root Definition of Related Systems

Root definition is especially useful when there are multiple stakeholders with different perspectives and objectives. In this software development process, defining root definition is vital to clarify essential goals or objectives of the process.

Root definition is applied to all sub-processes in order to improve overall software development process. To achieve this, CATWOE rule is defined and applied.

Customers (C): Customers of software development process are designers, developers, IT-department responsibles and other department's ERP responsibles basically. However, managers or leaders of designers, developers and head of departments can be counted as customers who are directly affected by the usage of the system. End users can be customers of employer which can be affected either directly or indirectly.

Actors (A): Actors of this process are the employees of development process such as designers, developers, IT-department and other departments.

Transformation (T): The required transformation can be defined as reconstructing some sub-process steps in order to achieve more efficient software development process. Because of current activities are inadequate, new process steps can be added or current activities can be changed.

Worldview (W): Worldviews of designers, developers and customers vary with respect to their experiences, job titles and interests.In this software development process, designers and developers have the most influential view in achieving the most effective improvement.

Owners (O): The owners of this process can be ERP system that is currently in use, managers, employees and customers who are benefited from this system's results.

Environment (E): The transformation environment composed of office, work area, infrastructure and an ERP software which are provided by the firm that they are work on.

Root Definition: Improving sub-processes of software development process by changing, redefining and restructuring existing process steps of design management, technical development, testing and training processes which do not have any standard guidelines.

4.5. Stage 4: Conceptual Models of Sub-Processes

Conceptual models are needed after defining root definition of the process. For each sub-process, conceptual models are constructed to improve overall software development process. Conceptual models can be visualized as schematic descriptions to gain more understandability.

4.5.1. Design Management

Defined conceptual model in design management process in Figure 4.5.1.1 gives insights about stakeholders’ way of doing work. Design management process begins with taking needs from customer and ends with designing a relevant ERP software. In conceptual model, requirements management and test management processes are defined in order to provide accurate traceability and stability. This process involves variety of roles such as designer, IT department and other departments. However, main responsible of design management process is designer. Conceptual model and elements of conceptual model are given in Figure 26 and Figure 27 respectively.

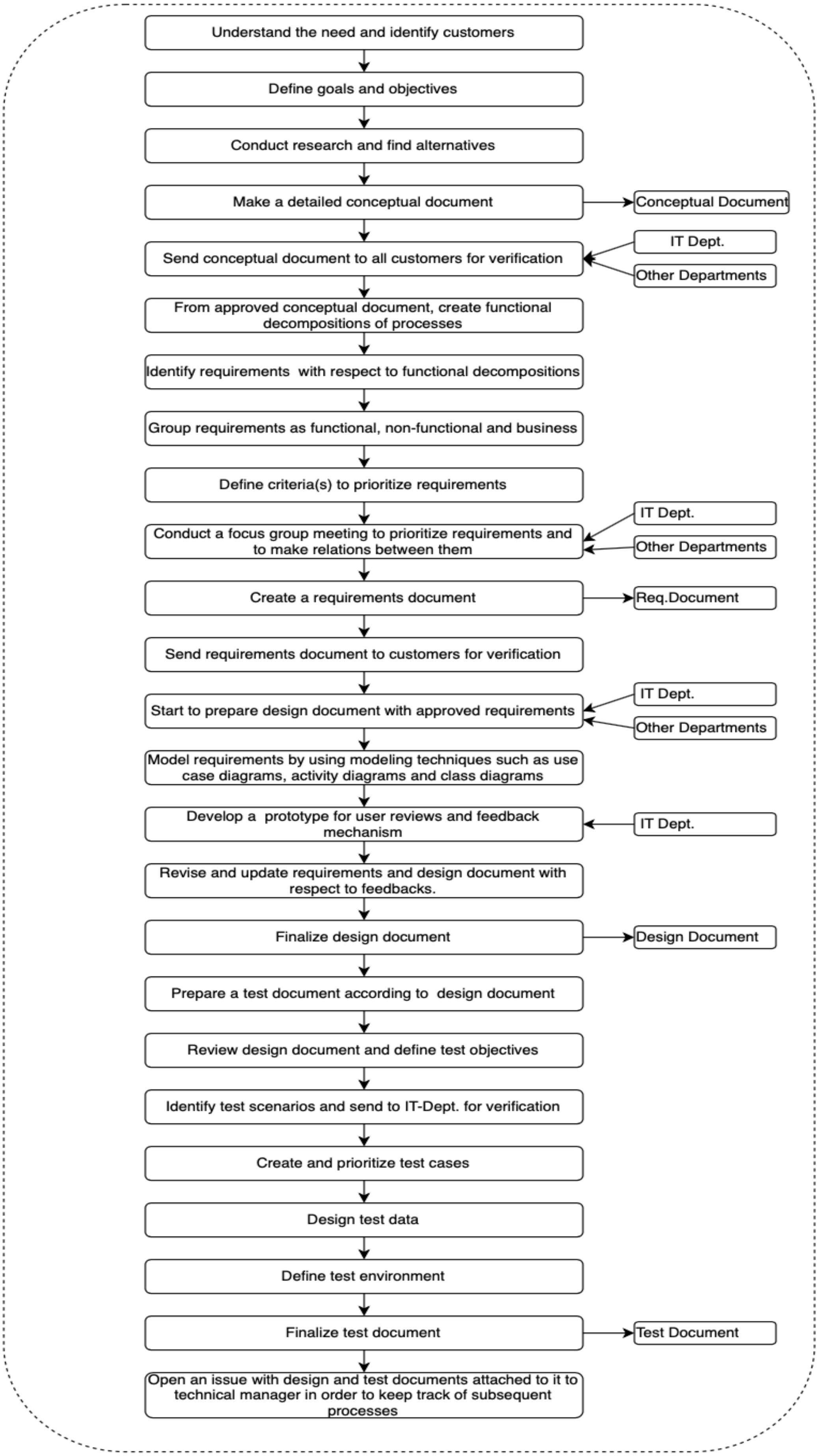

Understand the need and identify customers
Define goals and objectives
Conduct research and find alternatives
Make a detailed conceptual document
Conceptual Document
Send conceptual document to all customers for verification
IT Dept.
Other Departments
From approved conceptual document, create functional decompositions of processes
Identify requirements with respect to functional decompositions
Group requirements as functional, non-functional and business
Define criteria(s) to prioritize requirements
Conduct a focus group meeting to prioritize requirements and to make relations between them
IT Dept.
Other Departments
Create a requirements document
Req.Document
Send requirements document to customers for verification
Start to prepare design document with approved requirements
IT Dept.
Other Departments
Model requirements by using modeling techniques such as use case diagrams, activity diagrams and class diagrams
Develop a prototype for user reviews and feedback mechanism
IT Dept.
Revise and update requirements and design document with respect to feedbacks.
Finalize design document
Design Document
Prepare a test document according to design document
Review design document and define test objectives
Identify test scenarios and send to IT-Dept. for verification
Create and prioritize test cases
Design test data
Define test environment
Finalize test document
Test Document
Open an issue with design and test documents attached to it to technical manager in order to keep track of subsequent processes

Figure 26. Conceptual Model of Design Management Sub-Process

Table 2. Elements of Conceptual Model of Design Management

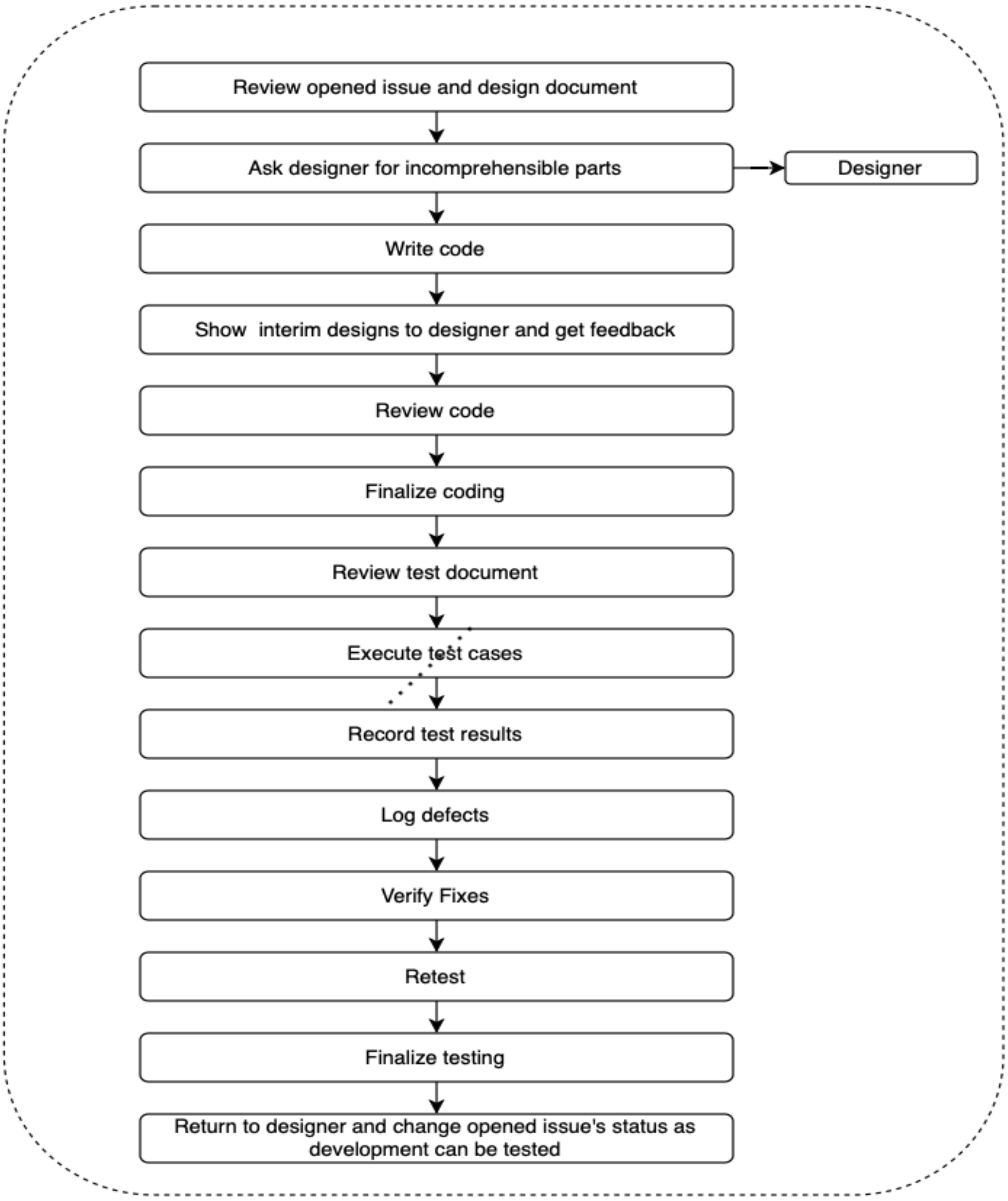
Elements of Design Management Conceptual Model
Understand the need and identify customers
Define goals and objectives
Conduct research and find alternatives
Make a detailed conceptual document
Send conceptual document to all customers for verification (responsible of IT Department and Other Departments)
From approved conceptual document, create functional decompositions of processes
Identify requirements with respect to functional decompositions
Group requirements as functional, non-functional and business
Define criteria(s) to prioritize requirements
Conduct a focus group meeting to prioritize requirements and to make relations between them (responsible of IT Department and Other Departments)
Create a requirements document
Send requirements document to customers for verification
Start to prepare design document with approved requirements
Model requirements by using modeling techniques such as use case diagrams, activity diagrams and class diagrams
Develop a prototype for user reviews and feedback mechanism (by showing IT Department)
Revise and update requirements and design document with respect to feedbacks.
Finalize design document
Prepare a test document according to design document
Review design document and define test objectives
Identify test scenarios and send to IT-Dept. for verification
Create and prioritize test cases

Table 2. Continued

Design test data
Define test environment
Finalize test document
Open an issue in ERP system with design and test documents attached to it to technical manager in order to keep track of subsequent processes

4.5.2. Technical Development Process

Technical development process mainly consists of developing code with respect to identified technical details, program logic that are written into design document. Conceptual model of this process mostly includes recording and reviewing to


Review opened issue and design document
Ask designer for incomprehensible parts
Designer
Write code
Show interim designs to designer and get feedback
Review code
Finalize coding
Review test document
Execute test cases
Record test results
Log defects
Verify Fixes
Retest
Finalize testing
Return to designer and change opened issue's status as development can be tested

minimize errors while coding. It is mostly performed by developer. Below figures are the conceptual model and elements of conceptual model of the technical development.

Figure 27. Conceptual Model of Technical Development Sub-Process

Table 3. Elements of Conceptual Model of Technical Development Sub-Process

Elements of Technical Development Conceptual Model
Review opened issue and design document
Ask designer for incomprehensible parts (review with designer)
Write code
Show interim designs to designer and get feedback
Review code
Finalize coding
Review test document
Execute test cases
Record test results
Log defects
Verify Fixes
Retest
Finalize testing
Return to designer and change opened issue's status as development can be tested

Technical development process mostly done by developer. Process includes unit-testing process to ensure that developed code is error-free. Hence, after finalizing code, developer has to do unit tests and record unit test results for future references.

4.5.3. Testing Process

Testing process ensures the things that are done in previous process steps, which are design management and technical development processes, perform correctly. It mostly controls test scenarios that are formed in conceptual model of design management. Below figures represents conceptual model and elements of conceptual model of testing process in Figure 28 and Table 4 respectively.

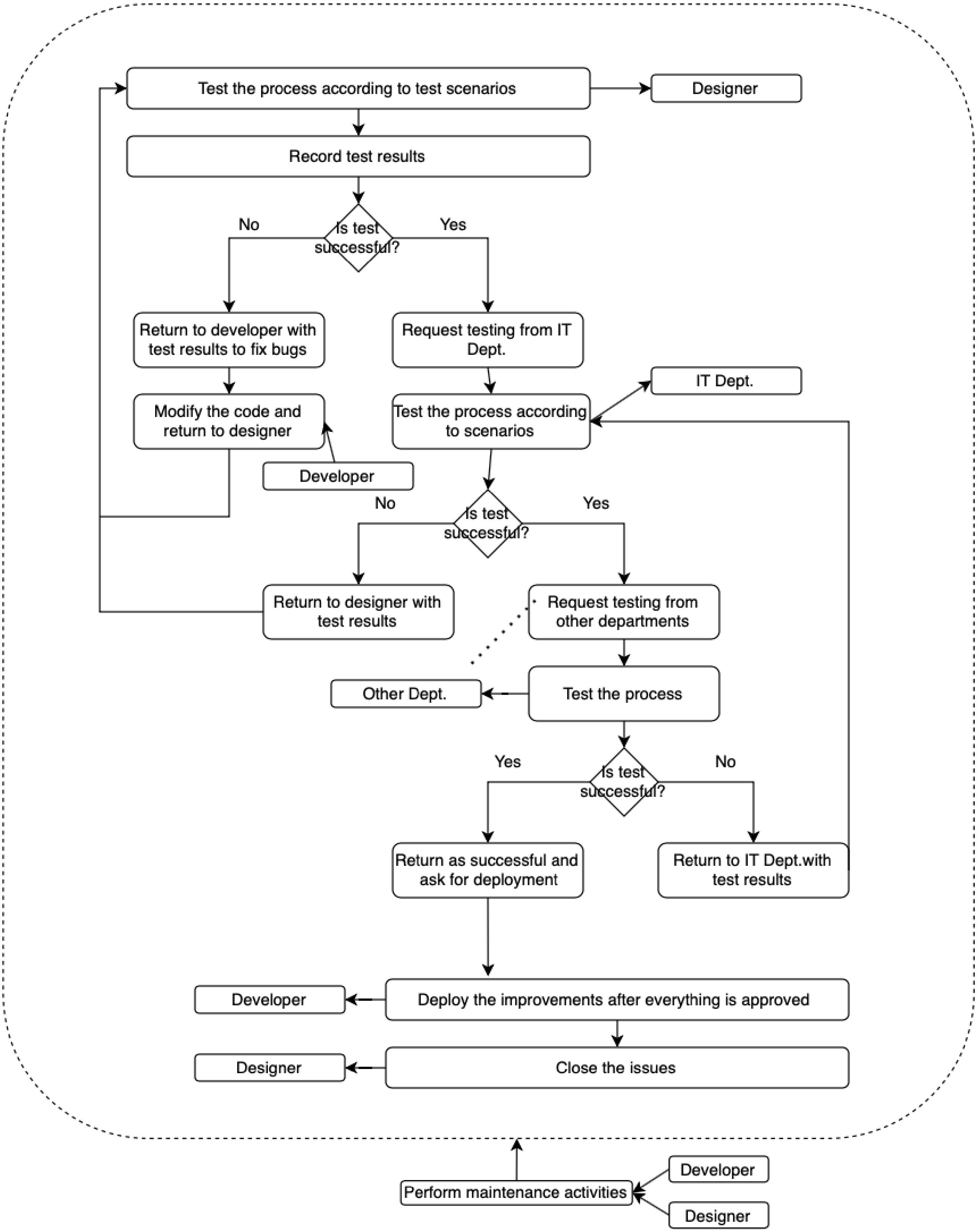

Test the process according to test scenarios
Designer
Record test results
No
Is test successful?
Yes
Return to developer with test results to fix bugs
Request testing from IT Dept.
IT Dept.
Modify the code and return to designer
Test the process according to scenarios
Developer
No
Is test successful?
Yes
Return to designer with test results
Request testing from other departments
Other Dept.
Test the process
Yes
Is test successful?
No
Return as successful and ask for deployment
Return to IT Dept.with test results
Developer
Deploy the improvements after everything is approved
Designer
Close the issues
Perform maintenance activities
Developer
Designer

Figure 28. Conceptual Model of Testing Sub-Process

Table 4. Elements of Conceptual Model of Testing Sub-Process

Elements of Testing Process Conceptual Model
Test the process according to test scenarios (by designer)
Record test results
If test is not successful, return to developer with test results and developer modifies the code
If test is successful, request testing from IT Department
Test the process according to scenarios (by IT Department)
If test is not successful, return to designer with test results
If test is successful, request testing from other departments
Test the process (by other departments)
If test is not successful, return to IT Department with test results
If test is successful, request for deployment
Deploy the improvements after everything is approved (by developer)
Close the issues (by designer)
Perform maintenance activities (by designer and developer)

Conceptual model of testing process ensures double-checks between developer and designer, designer and IT-department, IT-department and other departments with recorded results to ensure that none of the test scenarios are not unpracticed. Moreover, in the long run, if any bugs come from other departments maintenance activities are performed and testing process starts at the beginning.

4.5.4. Training Process

Training process is a supporting process which affects all of the firm’s employees. It helps better understanding of work, enables better performance by providing a steady growth. Below are the conceptual model and elements of conceptual model of training process.

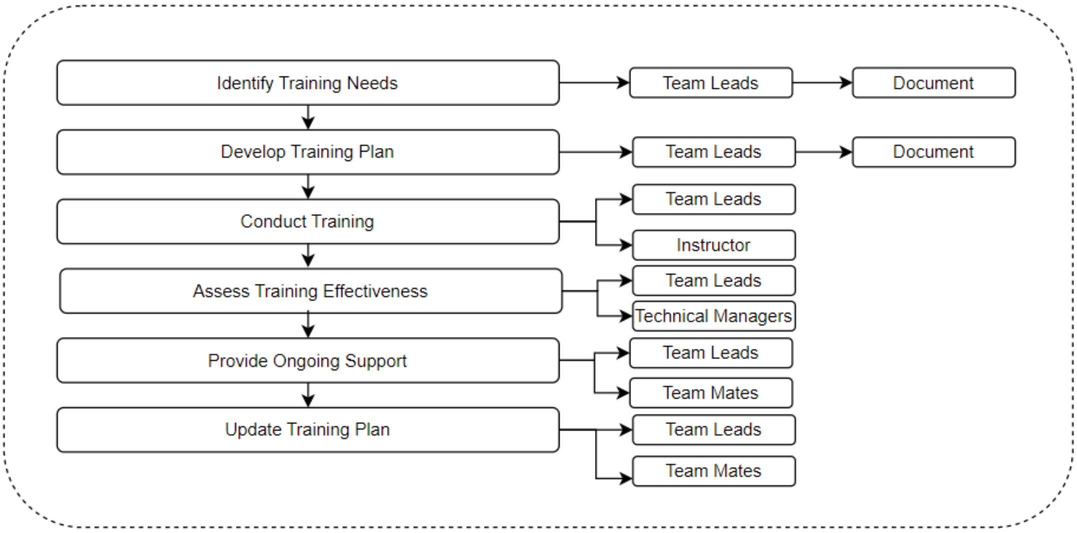


Figure 29. Conceptual Model of Training Process

Table 5. Elements of Conceptual Model of Training Process

Elements of Training Process Conceptual Model
Identify Training Needs
Develop Training Plan
Conduct Training
Assess Training Effectiveness
Provide Ongoing Support
Update Training Plan

To give better understanding of conceptual model, here are the detail explanations of each element:

1) Identifying training needs includes identifying the each team member's (designer, developer) skills and knowledge gaps and determining related training requirements.
2) Developing training plan covers training methods like classroom training, online courses or on-the-job learning and their schedule and duration.

3) Conducting training ensures all team members receive the necessary training that they are lack of and if necessary providing or showing hands-on experiences by team lead or collegues.
4) Assessing training effectiveness calculates effectiveness of the training program by taking into considerations of participants' observations, opinions and long term usage of learnings. Moreover, if needed, additional training areas can be determined.
5) Providing ongoing support means that providing support while team members apply their new learnings on to the job and offering resources whenever needed.

Updating training plan provides continuous growth on each team member by taking their feedbacks and making related changes into training plan.

4.6. Stage 5: Comparison

Conceptual models are compared with real world activities in a tabulated way. These comparisons revealed lots of matches as well as lots of mismatches and omissions. Real world activities of each conceptual model are described in below templates.

4.6.1. Design Management Process

Table 6.Comparison of Design Process Conceptual Model and Real World Activity

Elements of Design Management Conceptual Model	**Real World Activities**
Understand the need and identify customers	Need is understood and noted by designer and external customers or end-users usually can not be identified because designer works only with IT Department
Define goals and objectives	Meetings, phone calls or email exchanges are done in order to define goals and objectives of improvement,
Conduct research and find alternatives	Research is usually conducted, alternatives are usually searched before development. Existing software and processes are reviewed, however sometimes because of designer's lack of knowledge, research cannot be done properly.
Make a detailed conceptual document	A detailed conceptual document is not done for every process implementation. Moreover, there is not any defined conceptual document template.

Table 6.Continued

Send conceptual document to all customers for verification (responsible of IT Department and Other Departments)	Lack of any conceptual document leads to inadequacy of verification part.
From approved conceptual document, create functional decompositions of processes	Processes do not decompose of little sub-processes. And even if so, it is not written into any document.
Identify requirements with respect to functional decompositions	Requirements are not identified w.r.t. prioritization, risks, groups and etc.
Group requirements as functional, non-functional and business	Because of undefined requirements, they do not also grouped.
Define criteria(s) to prioritize requirements	Criteria(s) are not defined for prioritization cause requirements are not defined.
Conduct a focus group meeting to prioritize requirements and to make relations between them (responsible of IT Department and Other Departments)	Focus group meetings are held, however these meetings are mostly held on to understand the process better. They are not for requirements review and also results of meetings are not documented
Create a requirements document	There is not any defined requirements document template that is used.
Send requirements document to customers for verification	Because there is no requirements document, this step also becomes invalid.
Start to prepare design document with approved requirements	Design document is prepared however it is not based on requirements, mostly based on needs that customer mentions.
Model requirements by using modeling techniques such as use case diagrams, activity diagrams and class diagrams	Modeling techniques are barely used, cause designers only know specific ERP module, describes it with technical database terms, not with use cases.
Develop a prototype for user reviews and feedback mechanism (by showing IT Department)	Prototypes are barely used eventhough that specific ERP gives an opportunity to build a prototype.
Revise and update requirements and design document with respect to feedbacks.	Feedbacks are given by customer verbally. Hence, designer describes the new things to do to developer verbally.

Table 6.Continued

Finalize design document	Design document is written by designer.
Prepare a test document according to design document	Test document is not written cause there is not any temlaplate. If available, test data are added to design document
Review design document and define test objectives	This step is not valid cause there is not any test document.
Identify test scenarios and send to IT-Dept. for verification	This step is not valid cause there is not any test document.
Create and prioritize test cases	This step is not valid cause there is not any test document.
Design test data	Test data are barely added into design document if available.
Define test environment	Test environment is mostly defined, cause a specific ERP software is used for development and moreover, affected modules,functions are written into design document
Finalize test document	This step is not valid cause there is not any test document.
Open an issue in ERP system with design and test documents attached to it to technical manager in order to keep track of subsequent processes	An issue is opened by designer with desgin document in it.

4.6.2. Technical Development Process

Table 7.Comparison of Technical Development Process Conceptual Model and Real World Activity

Elements of Technical Development Conceptual Model	**Real World Activities**
Review opened issue and attached documents	Opened issue is reviewed by both technical manager and developer.
Ask designer for incomprehensible parts (review with designer)	Developer mostly asks designer for incomprehensible parts by phone, teleconference or face-to-face meetings.
Write code	When everything is understood, code is written.

Table 7.Continued

Show interim designs to designer and get feedback	Designs are shown at the end of the coding. This causes reworks most of the time.
Review code	Code review is done but sometimes not accurately which is also causes reworks.
Finalize coding	Code is finalized.
Review test document	Absence of test document makes this step invalid. However, if there are test data in design document, they are reviewed.
Execute test cases	Absence of test document makes this step invalid. However, if there are test data in design document, they are executed.
Record test results	Absence of test document makes this step invalid. Moreover, results of test data are not recorded.
Log defects	Absence of test document makes this step invalid.Moreover, there is not any defect logging.
Verify Fixes	Absence of test document makes this step invalid.However, test data can be used to verify fixes.
Retest	Retest is done.
Finalize testing	Testing is finalized by developer.
Return to designer and change opened issue's status as development can be tested	Developer returns to designer to test the developments via issue.

4.6.3. Testing Process

Table 8. Comparison of Testing Process Conceptual Model and Real World Activity

Elements of Testing Process Conceptual Model	**Real World Activities**
Test the process according to test scenarios (by designer)	There is not any formed test scenarios, however, implemented process is tested.
Record test results	Test results are not documented.

Table 8. Continued

If test is not successful, return to developer with test results and developer modifies the code	Test results are not documented, but it is told to developer verbally or short notes are forwarded to developer
If test is successful, request testing from IT Department	Test request is asked via mail or phone
Test the process according to scenarios (by IT Department)	There is not any formed test scenarios, however, implemented process is tested.
If test is not successful, return to designer with test results	Test results and sample data are returned to designer via mail or phone
If test is successful, request testing from other departments	Test request is asked via mail or phone
Test the process (by other departments)	There is not any formed test scenarios, however, implemented process is tested.
If test is not successful, return to IT Department with test results	Test results and sample data are returned to IT Department via mail or phone.
If test is successful, request for deployment	Deployment request is asked via mail or phone.
Deploy the improvements after everything is approved (by developer)	Designer asks for deployment to developer
Close the issues (by designer)	After deployment is done, all issues are closed by designer

4.6.4. Training Process

Table 9. Comparison of Training Process Conceptual Model and Real World Activity

Elements of Training Process Conceptual Model	**Real World Activities**
Identify Training Needs	Training is usually done on-the-job learnings, hence team leads do not identify skills and do not list training requirements for team members.
Develop Training Plan	There is not any training plan document identifed, also there is not any training methodology proposed. Only team lead says that a member can train herself if she wants.

Table 9. Continued

Conduct Training	There does not occur any training session which is conducted by any instructors. Sometimes hands-on experinces are showed.
Assess Training Effectiveness	If a team member receives certificate on her own, that certificate is saved into HR system. Apart from this, there is a performance evaluation, which is prepared by HR, is done once a year. It is a questionnaire and its extent is very basic.
Provide Ongoing Support	There is a one comprehensive resource as an e-source. Besides that, there is not any additional resource.
Update Training Plan	Hence there is not any training plan, there does not occur any updating.

4.7. Stage 6: Debate

In this part, suggested process changes or variations are explained in detail. These suggested changes are discussed with related focus groups.

1. In order to determine goals and objectives of the system which will be developed, there is a need for conceptual document.
    a. Conceptual document should include a detailed description of meetings, phone calls and mail contents about the process that will be implemented.
    b. Conceptual document should be sent to customers for approval, so that new requests can decrease in technical development or testing phases.
    c. Conceptual document should be prepared by designer and designer should work with both developer and customer. This collaboration lessens misunderstandings of the process that will be developed. Also, this working style decreases questions that may arise in the technical development and testing phases.
2. Process that will be developed should decomposed into sub-processes or functions. These sub-processes should be written into conceptual document.
    a. With this functional decomposition, understanding of processes increases.
    b. Functional decomposition is a prerequisite for requirements, so determining functional decompositions can be used for requirements elicitation and relation.
3. There is a need for requirements process.

    a. Requirements document should be created. Its template should be determined by team leads and technical managers. Template should include prioritization, risk degree, dependency and type (like business requirements, functional and non-functional requirements) of that requirement.
    b. Prepared requirements document should be sent to customers for approval. With this, new requests or claims will decrease in technical development or testing phases.
4. Design document should include requirements, modelling techniques and prototypes of the process that will be developed.
    a. Modelling techniques such as use cases, activity diagrams or class diagrams should be used in order to make developers understand more easily. This will help to lessen review duration of design document with developer and designer. Moreover, prototypes will assist developer to visualize the process.
5. There is a need for test document.
    a. Test document template should include test cases, test scenarios, relevant test data and test environment. This template should be determined by team leads and technical managers.
6. Test management activities should be identified.
    a. Test management activities should be done in design management, technical development and testing process according to identified test management steps. These activities are briefly defined in conceptual models of the processes.
7. There is a need for proper training process. Cause training is accomplished via on-the-job practices and optional e-learnings currently.
    a. Skill based training plans, proper training assessments, evaluation processes and support processes should be defined by team leads, group lead and technical managers.

4.8. Stage 7: Take Action

Findings and proposed changes are shared with subcontractor organization. Conceptual document template (see Appendix A), requirements template (see Appendix A), new design document template (see Appendix A) and test document template (see Appendix A) are presented to subcontractor organization.

CHAPTER 5

CONCLUSION

5.1. Discussion

This part aims to argue SSM and its relations with process improvement and software development process. SSM's implications from previous works are also considered.

Checkland identifies SSM as 'SSM is not a recipe, it is a way of thinking about problems'. SSM is an holistic approach that considers problems in its entirety and includes all relationships between stakeholders and covers related factors to help organizations in order to identify problems in their processes. It is not a structured, deductive approach like hard systems thinking where the problem is defined and solutions are explored regarding to some set of rules, it is an inductive approach where problem is investigated and solutions are developed in an iterative way.

As it's seen from the application on previous chapter, SSM is a flexible methodology that can be used any socio-cultural situations. It recognizes the importance of stakeholder involvement in the process and incorporates their perspectives (welteanschaung) while solving problems. On the other hand, hard systems thinking usually relies on mostly technical experts and their experiences.

SSM is a qualitative method that uses conceptual modelling, dialogues, meetings and creative thinking techniques to understand problems. These methods are used to give insights about the process or define alternative solutions and available improvements rather than deriving a single solution.

Continuous improvement is an important aspect in SSM. It is a necessary thing to achieve lasting results. Continuous improvement is achieved through monitoring process and evaluating solutions in a timely manner. Using definite techniques like CATWOE analysis, root definition elicitation in anytime prevent occurrence of wastes like time, defect, resources and etc which also increases quality of end software product.

In addition to above, performance of the SSM application can be easily measured in the ERP system. Key performance indicators can be defined such that completion time of design document, test document or investigation time of conceptual document. These completion times can be measured by using the start and end times of issues from ERP system's issue tracking module. Moreover, decrease of rework times by developers can be seen by analyzing how well design document is constructed with using number of feedbacks given by customers via email, meetings, or phone at the design management phase.

Overall SSM creates a shared knowledge within organizations and provides continuous improvement with using its tools and techniques. It can be easily used any human-involved cases. Even though software development process seems very straightforward process, since it involves human beings, it can be transformed very complex problem.

5.2. Conclusion

Organizational relations become important when they perform software development process together. An Application of SSM into SDP increased degree of collaboration between organizations and related employees.

The study began with interviews, meetings about software development process and its sub-processes followed by visualizing current processes with drawing rich pictures and BPMNs. During this process, the study required a mass collaboration between designers, developers and managers in order to collect data about sub-processes and then to visualize them. When questions about processes, how the processes work asked to participants in meetings, it is revealed that participants are the ones that point out to needed improvements. Analysis of these data showed that there are many areas that can be improved in the process.

According to semi-structured interviews, designers have the most complicated process and they work as medium between customers and developers in the current process. They take the information from customer, design related process and pass it to developers. If new request or any other issue comes from customer, designers should deal with the matter. And also, if any questions comes from developer, again designers should consider. Hence, designers' design management sub-process was a challenging process and most of the mistakes or failures are caused by this initial sub-process. Required changes are proposed in previous section. (Section 4.8)

Apart from this, lack of proper testing process causes some main failures in overall process according to participants' views. Testing process mainly depends on test data if it is indicated on design document by designer. If test data or test cases are not available it is barely tested by developer. However, developed process should be tested by designer and customer currently. In order to have proper testing process, a testing document which will be attached to design document is designed and proposed.

In addition to below parts, some conflicts occurred between participants of the meetings. In that situations, analyst investigated the conflict by making one-to-one dialogues, examining issues in the ERP system and observing the situation on-site to expose the reasons of conflicts.

Last outstanding finding from participants' arguments is that training process does not work properly. Hence training is optional, most of the employees do not attend the e-courses due to time restrictions or unwillingness. Moreover, team leads do not lead or mentor their teams, employees mostly learn during the job. This causes lots of waste of time. So, it is important to train personnel related to their skills and experiment. And some changes regarding to this issue are proposed in this research.

As it can be seen from previous chapter, software development process composes of four sub-process.

SSM is applied to each sub-process. Data collection is done via semi-structured interviews and issues are specified. Interview topics are grouped in Appendix. Analyses are done, after that conceptual models are constructed and conceptual models are compared with real life activities. Suitable changes are proposed to participants.

All in all, SSM is a participatory approach that can be applied to any complex structured problems to understand and define problems. SSM involves stakeholders and embraces ongoing learning with continuous improvement.

5.3. Contribution to Learning

There is a few literature about SSM application on software development process in organizations even though there are some examples of SSM in literature for any other areas. This study contributes improvements that can be done on software development process. Moreover, the study shows that continuous improvement is an important aspect of SSM while investigating software development process.

Even though there are some examples of SSM applications in literature, there’s a need for more studies in order to gain knowledge. Hence, this study highlights some important aspects of SSM application into software development process.

5.4. Limitations of the Study

Data collection for the study is done via semi-structured interviews, hence the study is shaped around participants’ views, comments and knowledge. It is also appropriate for SSM’s qualitative methodology which is also subjective. Sub-processes are defined regarding to results of interviews, so there may be missing processes and real life activities that are not argued in interviews.

Participants mainly consists of developers, designers and some team leads. There is one meeting conducted with technical manager, project manager and other participants. In meetings, some participants may not share all of their opinions on issues in order not to affect analysis or some fear from upper management. Hence, gathering quality data was a challenging process.

Another limitation is that software development process requires 2 different organizations to perform. Meetings mainly done with subcontractor organization due to some permission issues with other organization. However, since analyst works on subcontractor organization for a while, it might not be too difficult to observe what other organization does and communicate with them.

In addition to above limitations, last stage of SSM cannot be applied since it is out of the context of this study. Only changes and some draft documents are proposed to organization.

5.5. Delimitation of the Study

Since subcontractor organization has a matrix structure and works on project based, this SSM application on software development process is limited for specified sub-organization’s culture, its relations between subcontractor firm and other firm, location, ERP sector, business model and employees in the project.

5.6. Future Research

The study was conducted for only one ERP project in organization. For future research purposes, it can be conducted for all ERP projects, hence it can be widely investigated with more participants. Moreover, software development process is affected by organization’s internal processes such as project management, human resources or strategy processes can affect software development process implicitly. Affecting processes’ relations with software development process and how they affect it can be investigated in more detail.

Software development process is investigated and some adjustments are proposed in the study. For future studies, after proper investigations and improvements, key performance indicators can be formulated and with using or comparing these performance metrics in a timely manner, one can obtain continuous improvements for software development process with or without affecting processes.

REFERENCES

Ackoff, R. L. (1979). The future of operational research is past. *Journal of the operational research society*, *30*(2), 93-104.

Aldin, L. & Cesare, S. (2011). A Literature Review on Business Process Modelling: New Frontiers of Reusability, *Enterprise Information Systems,* doi: 10.1080/17517575.2011.557443

Angrosino, M. V. (2007). *Naturalistic observation.* Left Coast Press.

Awad, M. A. (2005). A comparison between agile and traditional software development methodologies. *University of Western Australia*, *30*, 1-69.

Bayraktaroglu, B. (2018). Soft Systems Methodology for Design Process Improvement. *Middle East Technical University*.

Beer, S. (1989). The viable system model. *Viable Systems Model, Wiley, Chicester*.

Bertalanffy, L. V. (1968). *General system theory: Foundations, development, applications*. G. Braziller.

Boaden, R.J. ,(1997). What is total quality management...and does it matter? *Total Quality Management*, 8(4), 153-171.

Canfora, G., Cimitile, A., Garcia, F., Piattini, M., Visaggio, C.A. (2006). Productivity of Test Driven Development: A Controlled Experiment with Professionals. In: Münch, J., Vierimaa, M. (eds) Product-Focused Software Process Improvement. PROFES 2006. Lecture Notes in Computer Science, vol 4034. Springer, Berlin, Heidelberg. doi:10.1007/11767718_32

Capra, F. (2002) *The Hidden Connections: Integrating the Biological, Cognitive, and Social Dimension of Life into a Science of Sustainability*. Used by permission of Doubleday, a division of Random House, Inc., and, in the UK, reprinted by permission of HarperCollins Publishers Ltd.

Charmaz, K. (2014). *Constructing grounded theory*. Sage.

Checkland, P. (2000). Soft Systems Methodology: A Thirty Year Retrospective. *Systems Research and Behavioral Science, Syst. Res.17*, S11-S58.

Checkland, P., & Poulter, J. (2020). Soft systems methodology. *Systems approaches to making change: A practical guide*, 201-253.

Checkland, P., & Scholes, J. (1999). *Soft systems methodology in action*. John Wiley & Sons.

Cheng, C. K., & Permadi, R. B., (2009). Towards an Evaluation Framework for Software Process Improvement .

Creswell, J. W. (2013). Steps in conducting a scholarly mixed methods study.

Curtis, B., Kellner, M.I. and Over, J. (1992) Process Modeling. C*ommunications of the ACM,*35,75-90.doi:10.1145/130994.130998.

Dowson, M., Fernström, C. (1994). Towards requirements for enactment mechanisms. In: Warboys, B.C. (eds) Software Process Technology. EWSPT 1994. Lecture Notes in Computer Science, vol 772. Springer, Berlin, Heidelberg. doi: 10.1007/3-540-57739-4_13.

Forrester, J.W. (1961) *Industrial Dynamics*. MIT Press, Cambridge, Mass.

Fuggetta, A. & Nitto, E. (2014). Software Process, Future of Software Enginering Proceedings, 1-12, doi: 10.1145/2593882.2593883.

Gasson, S. (1994). The Use of Soft Systems Methodology (SSM) As A Tool For Investigation *Warwick Business School*.

Green, F.B., (2007)Six-Sigma and the Revival of TQM. *Total Quality Management & Business Excellence, 17*(10), 1281-1286 doi: 10.1080/14783360600753711.

Jiang, W. (2018). A bibliometric study on the use of Lean thinking in services.

Kılıc, A.B. (2018). Two Strands Model of the Soft Systems in Action: The Case of an SME in Automotive Component and Defense Industry, *Middle East Technical University*.

Korsaa, M., Johansen, J., Schweigert, T.,Vohwinkel, D.,Messnarz, R., Nevalainen, R. & Biro, M. (2013). The people aspects in modern process improvement. Journal of Software Maintenance and Evolution: Research and Practice. 25. 10.1002/smr.570.

Krueger, R. A., & Casey, M. A. (2015). Focus group interviewing, Handbook of practical program evaluation, 506-534.

Kvale, S. (1996). The 1,000-page question. *Qualitative inquiry*, *2*(3), 275-284.

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. sage.

Lucena, P., Braz, A., Chicoria, A. & Tizzei, L. (2016). IBM Design Thinking Software Development Framework. doi:10.1007/978-3-319-55907-0_9

Meadows, D. H. (2008). *Thinking in systems: A primer*. Chelsea Green Publishing.

Omachonu, V.K.& Ross, J. E., (1994) Principles of Total Quality. *Journal For Healthcare Quality* 16(6):p 36,38.

Osterweil, L. J., (1997). Software processes are software too, revisited: an invited talk on the most influential paper of ICSE 9. In *Proceedings of the 19th international conference on Software engineering* (pp. 540-548).

Patton, M. Q. (2015). *Qualitative Evaluation and Research Methods.* Thousand Oaks, CA: Sage.

Perry, D., Staudenmayer N. A. &Lawrence G.V.,(1994) .People, organizations, and process improvement. *IEEE Software* 11(4): 36–45.

Richmond, B. (2000). The Thinking in Systems Thinking: Honing Your Skills. *The Systems Thinker*, 11, 6.

Rosa, A. C. M., & Broday, E. E. (2018). Comparative analysis between the industrial and service sectors: a literature review of the improvements obtained through the application of lean six sigma. *International Journal for Quality Research*, *12*(1), 227.

Schmid, K.,Becker-Kornstaedt, U.,Knauber, P. & Bernauer, F. (2000). Introducing a software modeling concept in a medium-sized company. 558-567. doi:10.1145/337180.337461.

Senge, P. (1990). The Fifth Discipline: The Art and Practice of the Learning Organization. New York: Doubleday/Currency,doi: 10.1002/hrm.3930290308.

Soomerville, I. (2004). *Software Engineering*(9th Ed) Pearson Education p(3-118).

Stake, R.E. (1995). *The Art of Case Study Research*, Sage

Turner, W.C., Mize, J.H., Case, K.E., Nazemetz, J.W., (1993). Introduction to Industrial and Systems Engineering(3rd Ed), Prentice Hall

Waring, A. (1996). *Practical Systems Thinking*, Thomson.

Womack, J. P., Jones, D. T., & Roos, D. (2007). *The machine that changed the world: The story of lean production--Toyota's secret weapon in the global car wars that is now revolutionizing world industry*. Simon and Schuster.

Wheatley, M.J. (2006) Leadership and the New Science: Discovering Order in a Chaotic World. 3rd Edition, Berrett-Koehler Publishers, Inc., San Francisco.

Yin, R. K. (2018). *Case Study Research and Applications: Design and Methods* (6th ed.). Thousand Oaks, CA: Sage.

Zarour, K., Benmerzoug D., Guermouche N.& Drira K. (2019) A systematic literature review on BPMN extensions. *Business Process Management Journal*, 26 (6), pp.1473-1503. ff10.1108/BPMJ-01-2019-0040ff. ffhal-02419886f.

APPENDICES

APPENDIX A

DRAFT PROPOSED DOCUMENTS

DRAFT CONCEPTUAL DOCUMENT CONTEXT

MODULE NAME DATE, VERSION
PROCESS NAME THAT WILL BE DEVELOPED

STAKEHOLDERS THAT ARE BENEFITED

Name, Surname	Responsibility	Firm	Approval

PROCESS NOTATION TABLE

Shape	Explanation
◆	Decision
■	Process Step

Process Overview

1. Process Outline
2. Objective
3. Goals
4. Functional Decomposition
5. Process Modeling
6. Business Rules Analysis
7. Document Analysis
8. Risk Analysis
9. Deadline Estimation

DRAFT REQUIREMENTS DOCUMENT CONTEXT

MODULE NAME DATE, VERSION
PROCESS NAME THAT WILL BE DEVELOPED

STAKEHOLDERS THAT ARE BENEFITED

Name, Surname	**Responsibility**	**Firm**	**Approval**

MAIN FUNCTIONS

Main Functionalities	**Must**	**Should**	**Could**	**Won’t**
Stock Control				
Production				
Packaging				
…				

REQUIREMENTS

Requirement no: DFD_01			
Main Functionality:			
Priority:	Explanation		
Criticality:			
Risk:			
Type:	**Feasibility:**	**Role:**	**ERP Solution:**

COVERAGE MATRIX FOR REQUIREMENTS

Req/Req	**Requirement 1**	**Requirement 2**	**Requirement 3**
Requirement 1	X		
Requirement 2		X	✓
Requirement 3		✓	X

DRAFT NEW DESIGN DOCUMENT CONTEXT

MODULE NAME DATE, VERSION
PROCESS NAME THAT WILL BE DEVELOPED

Prepared by	
Preparation Date	
Due Date	
Priority	
Related Requirement No	
Description	
Development type	
Roles	

EXPLANATIONS

Definition/Goals of Development
Business Rules / Process Flows
Technical Explanations

(class diagrams, ER diagrams, data flow diagrams, sequence diagrams, authorizations, data modeling and etc.)

Interface Design

Prototype Design

DRAFT TEST DOCUMENT CONTEXT

MODULE NAME DATE, VERSION
PROCESS NAME THAT WILL BE DEVELOPED

Test Plan No	
Requirements No	
Test Description	
Test Goal	
Prepared By	
Due Date	

USE CASE NO X

Scope
Level
Primary Actor
Preconditions
Postconditions
Main Success Scenario
Alternatives
Assumptions

RELATED TEST CASES

Test Case 1

Test Case 2

Test Case N

Test Results

Test Summary

APPENDIX B

MEETING NOTES

Meeting Note #1 (General Meeting)

We are a subcontractor company specializing in the development of ERP solutions for our clients... Unlike being based at our company's headquarters, our operations are carried out on-site at the customer's location... Our primary responsibility is to align the ERP processes with the customer's existing workflows, ensuring compatibility and integration... We create designs and developments accordingly, striving to adhere to the rules and guidelines set by the ERP system… The project comprises a technical manager and a project manager overseeing the overall operations… The project is tracked by project engineers in headquarters, budget and time of the project is known by them…Within the project, we have dedicated teams for various modules, including procurement, sales, finance, human resources, maintenance, repair, BW (Business Warehouse), and development, each led by a team leader… It is worth noting that team leaders may also be involved in other projects and are not exclusively tied to this project… The coordination and direction of the designer and developer colleagues are generally done by the technical manager… Here, we usually do not follow some of our company's documents, it's seen as very time consuming and not appropriate…We have designers and developers basically, need is coming to designers and according to communications with customers either designer solves the problem directly, or developer makes a development…We see designers as main responsibles for everything… Including developers' developments…It is believed that customer is always right, hence when we interrogate needs, they are usually done the way what customer says…As a project team we are a very small group about 30 people… Sometimes headquarter's absence is felt by project team… It is thought that HR department is inefficient in terms of moral and motivation… Of course, we are currently work on hybrid bases, so socialization may be affecting this situation…Designers and developers sit on one office…This arrangement may disturbs to developers cause there can be lots of noises due to designers and customers conversations…We have open office…Technical manager and project manager don't have to come to office everyday…There is not much quality or verification activities, since we work on an ERP solution, it gives some standard solutions…Employee qualification is important but not a necessity, they usually learn on the job… There is not any training given by team leads, it would be nice, but there is no time for it… Documents… We have one design document and everything is connected to this document… Sometimes designers do not use the document, they just write on the issue on ERP… Testing can be annoying, cause we do not have a defined one…Responsibilites are complicated…

Meeting Note #2 (Design Process)

The design process begins with a customer's request, which is conveyed to the IT department for an initial assessment... The IT department then communicates the request to us, either via email or through their interpretation... Based on these information, meetings we create a design…However, there are instances where the IT department misinterprets requests from other departments, resulting in faulty designs... Alternatively, they may forward requests without review, leading to user errors or unimplementable situations… These incidents cause wasted time and effort...

In addition, instead of providing general training for new members of our design team, we prefer on-the-job learning… This sometimes results in new employees spending time evaluating the suitability of requested features...Or searching for lot of time… Moving forward, the IT department examines requested tasks and assigns them to the appropriate module consultant... The consultant reviews the request, documents the design in a "design" document, and opens an issue with the developers' team lead… Insufficient discussion with the IT department prior to the design phase can make it challenging for developers… Incomplete designs may lead to incorrect assumptions by designers, making it difficult to fulfill customer requests… Lack of proper communication with the customer also hampers the feedback mechanism…

Sometimes, test data are not always included in the design document, causing developers to request additional test data or test what has already been added without specifying which conditions to test… Designers may encounter difficulties with testing and preparing test data, resulting in them relying on the IT department for testing and feedback…Having unequal live an development systems also a cause…These delays significantly impact the overall process…We expect that all module related processes should be known by designer or module team…Customers and IT department responsibles freely visit desginers in their office…Causing lots of noise…

Meeting Note #3 (Development Process)

Developers thoroughly examine the documents they receive, but there are instances where designers bypass creating a document and directly input the customer's request within the issue opened in the ERP system… This practice leaves developers uncertain about the appropriate course of action, necessitating frequent clarifications from designers… It is crucial to prevent this situation… Furthermore, there are occasional deficiencies in the design documents received… Incomplete design documents require developers to frequently consult designers, who themselves may not possess a comprehensive understanding of the process, leading to additional inquiries directed to the IT department or even the customer… A significant amount of time is lost… Apart from these challenges, assuming the design is accurate, developers proceed to code implementation and subsequently enter the testing phase… They commonly acquire knowledge about unfamiliar topics by seeking guidance from their teammates, engaging in on-the-job learning... During testing, developers execute the tests if test data are available… However, in cases where there is no test data, they typically review the design without conducting tests… Sometimes test data is requested from the designer…It is thought that testing job mainly belongs to designer…No need to concern about infrastructure, ERP is a package, it is provided… Only software coding is done…We usually work with earphones, cause open office has lots of noises…

Meeting Note #4 (Testing Process)

In this process, if test data are available in the design document, developers proceed with testing… Otherwise, they rely on the designer for testing... Once the designer performs the test, it is further evaluated by the IT department... Subsequently, the IT department returns the results to the responsible department for testing… Throughout this process, there are feedback loops established among the designer and developer, designer and IT department, and IT department and customer…Sometimes the absence of a standardized test document often leads to overlooked scenarios and associated test data… As a consequence, process deficiencies arise, sometimes necessitating a complete rework of both the design and code… The responsibilities and roles in testing are not clearly defined, lacking a clear delineation of tasks and responsibilities… Due to this testing gap, we frequently receive negative feedback from the customer…

Meeting Note #5 (Training Process)

Newcomers in our organization receive limited training opportunities… Usually, learning occurs through on-the-job experiences and seeking guidance from colleagues… While we do have an online learning platform called the Learning Hub, it is not actively utilized… It is very optional…We can have certification from this platform, it is acceptable for worldwide…The reason behind that we don't actively use Learning Hub is the challenges we face are often unique to our organization's processes, having standard solutions ineffective… In light of this situation, what steps can be taken to address the issue?... One potential solution could be for designers to document all processes related to the modules they are working on whenever feasible… Maybe design document's context can be perfected…A comprehensive plan can be developed to facilitate this documentation effort…Since each team has a designated team leader, they could potentially provide in-team training to their respective team members… It is crucial to engage in discussions and collaborate with the team leaders to outline and implement these training initiatives…