AN AGILE BUSINESS PROCESS SOFTWARE DEVELOPMENT METHODOLOGY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DAVUT ÇULHA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

SEPTEMBER 2014

Approval of the thesis:

AN AGILE BUSINESS PROCESS SOFTWARE DEVELOPMENT METHODOLOGY

submitted by **DAVUT ÇULHA** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı ____________________
Head of Department, **Computer Engineering**

Prof. Dr. Ali Doğru ____________________
Supervisor, **Computer Engineering, METU**

Examining Committee Members:

Prof. Dr. Onur Demirörs ____________________
Information Systems, METU

Prof. Dr. Ali Doğru ____________________
Computer Engineering, METU

Assoc. Prof. Dr. Halit Oğuztüzün ____________________
Computer Engineering, METU

Assoc. Prof. Dr. Pınar Karagöz ____________________
Computer Engineering, METU

Asst. Prof. Dr. Bedir Tekinerdoğan ____________________
Computer Engineering, Bilkent University

Date: 02.09.2014

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 : Davut Çulha

Signature :

ABSTRACT

AN AGILE BUSINESS PROCESS SOFTWARE DEVELOPMENT METHODOLOGY

Çulha, Davut

Ph.D., Department of Computer Engineering

Supervisor: Prof. Dr. Ali Doğru

September, 2014, 161 pages

An agile business process software development methodology is proposed, developed and tested in this research. To speed up the business process software development practices in the organization and to address the requirements more efficiently, an agile approach was adapted. Two new processes were developed using the new methodology. The improvement was assessed by utilizing nine older developments: A formula was developed in this research that estimates the development efforts for old business process software development projects. The motivation mainly was to efficiently gather desired requirements and decrease the development time. There are difficulties in applying agile practices to the domain: stakeholders of the business process software development deal with more than one project at the same time. Moreover the proposed methodology suggests a critical utilization of training that improves the gathering of quality requirements. Agile requirements gathering, periodic meetings, and incremental and iterative development are observed to be the building blocks of the proposed methodology during the studies for applying the methodology to two processes in an organization. A survey on business process software development methodologies is included. There are currently process development methodologies and limited adaptation work on agile approaches to process redesign. Such existing work does not define a specialized agile methodology for business process software development. In addition, the proposed methodology is examined based on the effort spent during the development. The examination is realized with the effort estimation formula. According to the formula, a 21% effort saving is realized with the proposed methodology compared with the traditional methodology.

Keywords: agile, business process, development methodology

ÖZ

ÇEVİK İŞ SÜRECİ YAZILIMI GELİŞTİRME YÖNTEMİ

Çulha, Davut

Doktora, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Ali Doğru

Eylül, 2014, 161 sayfa

Bu çalışmada, çevik bir iş süreci yazılımı geliştirme yöntemi önerildi, geliştirildi ve test edildi. Organizasyondaki iş süreci yazılımı geliştirme pratiklerini hızlandırmak ve gereksinimleri etkin şekilde belirleyebilmek için çevik bir yaklaşım uyarlandı. Yeni yöntemle iki yeni süreç geliştirildi. Dokuz eski geliştirmeyi kullanarak iyileştirme değerlendirildi: Bu çalışmada eski iş süreci yazılımı geliştirme projelerinin geliştirme iş gücünü kestirmek için bir formül geliştirildi. Temel güdülenme istenen gereksinimleri etkin şekilde belirlemek ve geliştirme zamanını azaltmaktı. Sorun kümesine çevik pratikleri uygulamada zorluklar vardı: iş süreci yazılımı geliştirme paydaşları, aynı anda birden çok projeyle uğraşıyorlar. Bunun yanında önerilen yöntem, kaliteli gereksinimlerin toplanmasını iyileştiren eğitimin önemli şekilde kullanımını önermektedir. Organizasyondaki iki sürece yöntemin uygulanması çalışmaları sırasında, çevik gereksinim toplanmasının, düzenli toplantıların, artan ve döngüsel geliştirmenin, önerilen bu yöntemin önemli parçaları olduğu gözlemlenmiştir. Ayrıca iş süreci yazılımı geliştirme yöntemleri hakkında bir tarama da eklenmiştir. Şu anda süreç geliştirme yöntemlerine ve çevik yaklaşımların süreç yeniden-tasarlamasına yönelik kısıtlı çalışmalar bulunmaktadır. Var olan bu çalışmalar iş süreci yazılımı geliştirmesi için özelleştirilmiş çevik bir yöntem tanımlamamaktadır. Ek olarak, önerilen yöntem geliştirme esnasındaki iş gücüne göre incelenmiştir. İnceleme bir iş gücü tahmin etme formülü ile gerçekleştirilmiştir. Bu formüle göre, geleneksel yönteme göre %21'lik bir iş gücü kazancı sağlanmıştır.

Anahtar Kelimeler: çevik, iş süreci, geliştirme yöntemi

To my family

ACKNOWLEDGMENTS

I would like to thank to my supervisor Prof.Dr. Ali Doğru for his encouragement, advice and guidance throughout this study.

I would like to thank to Prof. Dr. Onur Demirörs and Assoc. Prof. Dr. Pınar Karagöz for their valuable comments and guidance during the thesis monitoring meetings.

I would like to thank to the members of the thesis committee Assoc. Prof. Dr. Halit Oğuztüzün and Asst. Prof. Dr. Bedir Tekinerdoğan for their valuable and constructive comments.

I would like to thank to my dear parents Rahime Çulha and Cemalettin Çulha for their infinite support and understanding in my life.

I would like to thank to my paternal uncle Remzi Çulha and my maternal uncle Hacı Mustafa Özkan for their guidance beginning from my childhood in my life.

I would like to thank to my parents-in-law Saime Şenol Karcı and Mehmet Ragıp Karcı for their endless support.

I would like to thank to my handsome son Ahmet Alp Çulha and my sweet daughter Ayşe Ece Çulha for wasting some of their playtime and for their patience during this study.

At last, but not the least, I would like to thank to my beloved wife Nisa, who provided everything I needed. This study would not be possible without her love, support, encouragement, and patience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

A business process is a collection of structured and related tasks or activities. Tasks or activities are executed by actors. An actor is a human, an application or a combination of a human and one or more applications.

There is high number of studies on business processes. Business processes are studied from many different points of view. Business processes are defined, developed, implemented, enacted, configured, and optimized. In other words, management of business processes is required, coining the phrase “Business Process Management” (BPM) [1]. The aim in this research is to propose an agile methodology for developing business processes that is superior to the classical Waterfall methodology.

Manual hand-filled forms are gradually replaced by electronic forms. These forms are embedded in business processes.

The methodology has been built completely based on the experiences which are obtained during the automation of business processes using the classical waterfall model. During that work, the problems of the classical methodology are examined and in order to solve those problems an agile approach is being proposed. During the examination of business processes in the organization, the following results were extracted:

- A task which is automated does not completely fulfill the actual task. Therefore, some parts of the task should be completed manually.
- Some tasks are not automated because they are not specified enough.
- The interactions between tasks are not automated because each task may have a different data exchange format.

These results show the problems standing in front of the business processes. If these problems are solved, business process software may be developed easily.

Some business processes have been implemented in an organization. They have been implemented through the phases which are analysis, design, and coding. This phased development is actually based on the Waterfall model. Each of them took approximately 1-year development time. During the development some problems were observed. The main problem was the missing requirements. Actually this problem was noticed at the beginning of going live. At that time, some ignored actors in the process or unknown actors of the process appeared. This caused the review of the requirements specification, redesign and reimplementation. Another problem was that some additional tasks were required to be added to the processes. Also, roles of the tasks changed. In other words, missing actors appear and want to undertake related roles.

These problems address the need to use agile methodologies with the main expectation of effort saving. Also, requirements would be gathered better. Some business processes were tried to be implemented using agile methodologies in an organization. However, it was realized that some problems occur because of the characteristics of business processes and the organization. The basic problems of using agile methodologies are:

- Business process automations are usually more complex than standard software automations. Therefore, a 2 or 3-week iteration periods do not fit to business processes. A bit longer period seems to be suitable for them.
- In the organization a person deals with more than one project at a time. For example s/he can be involved in three projects. Also daily meetings are time consuming and confusing. Its period and structure need some arrangements.

In addition to the problems, there are also some common customer requirements in the organization. These are:

- A person who accepts a task can send it to other people serially or in parallel for review.
- Some steps in the process may require anterior approval. i.e., the same task is approved by someone else before the actual task is approved.
- Some tasks are approved by a chain of approvers according to the organizational hierarchy.
- Before a step in the process, some actions may be needed.
- Moreover, some actions may be needed after a task.
- Roles in the tasks can be delegated to some other people.
- Everyone should be able to see the approvers' list of a process.
- Some parts of the process are dynamic. i.e., those tasks which are assigned to some coordinators are sent to some other people dynamically according to the coordinator's decisions.

The encountered problems and the customer requirements in the organization address that a development methodology is needed. Since missing requirements can cause reimplementation, gathering of the requirements is very important. Therefore, the methodology may include some prototyping. Furthermore, agile approaches may be included to determine the actual requirements of the process. Also, agile approaches may decrease the development time. In conclusion, it was decided to develop an agile methodology for business process software development. In this context, the term "development" covers the phases that occur during business process management projects.

The rest of this thesis is organized as follows. The background information is given in the next chapter. Then, the business processes domain is analyzed. After the introduction of the new methodology, the estimation formula is derived to compare the methodologies based on effort values. Related work is presented in the following chapter. Before conclusion the proposed methodology is applied as a case study.

1.1. A Summary of the Conducted Work

Motivation: Effort savings in developing Business Processes. Handling requirements more efficiently for Business Processes. Adaptation to change requests.

Approach: Developing an agile methodology and enacting it for two new business processes.

Evaluation of Existing Capabilities: 4 approaches were investigated (Waterfall, RUP, XP, Scrum) using 51 criteria.

Contribution: An Effort estimation formula for Waterfall based business process software development. A new business process software development methodology. A case study for trying the proposed methodology. A methodology comparison approach through compiling 51 criteria.

Evaluation: Effort gain report, validity threat discussion, discussion on software quality factors

CHAPTER 2

BACKGROUND

In this chapter, the background information is given. In the following sections, business processes and their related technologies are explained.

2.1. Business Processes

A business process is a collection of structured and related tasks or activities. In general the terms task and activity are used interchangeably. Generally, the term task is used to describe a step in a business process in which there is a human interaction. If there is no human interaction, the step is called an activity. Moreover, the terms such as process activity, logical step, work element, and work item are also used for steps in a business process [5].

Tasks or activities are executed by resources. A resource is a human, an application or a combination of a human and one or more applications. The capabilities of a resource are provided by a set of roles. Each task requires a specific role. Roles are used to map the task instances to resources. The terms actor and participant are also used for resource.

2.2. Business Process Management

There is considerable amount of work on business processes. Business processes are studied from many different points of view: They are defined, developed, implemented, enacted, configured, and optimized. In other words, management of business processes is required, coining the phrase "Business Process Management" (BPM). Surveys on BPM can be found in [1] [26].

Over the past two decades, previously manual hand-filled forms were increasingly replaced by their "paperless" electronic counterparts [12]. These electronic forms are found places in business processes. Business processes are executed by a process engine contained in BPM. A process engine can be seen as a software server that keeps track of each individual work item and routes the item to the next task. BPM tries to optimize the processes to get competitive advantages.

Historically BPM has been affected by the information systems trends [22]. In 1970s and 1980s, data-driven approaches were effective. The focus of information technology was on data-modeling because storing and retrieving information was important. Modeling of business processes was neglected during that period. Later the trend shifted to process-oriented approaches because business process re-engineering increased emphasis on processes. As a result, more process driven approaches are being used for building information systems.

Service-oriented Architecture (SOA) combines loosely coupled, standards-based, and protocol-independent distributed computing concepts [73]. SOA, XML and web services have contributed to the advance of BPM technology [2]. In addition, the component orientation technology and good business process modeling standards contributes to the advance of BPM technology [1].

BPM application areas are enriched and large number of BPM application cases exposed in the last decade. As a result of this, business process modeling has gained more importance. There are two reasons for them. One is the organizational needs to survive in the turbulent environment. The other one is the increased demand on the quality of the commercial tools for BPM. Availability of many commercial BPM tools requires methods and methodologies for tool acquisition. In [13], some issues are considered and some guidelines are suggested for tool acquisition.

2.3. Workflow Management

A workflow consists of a sequence of connected steps. A workflow may be seen as an abstraction of real work. In other words, it is a virtual representation of real work. In a real work, usually a document is transferred from one step to another. This control flow can be described using workflows and can be modeled to assess the actual work.

Information systems try to extract functionalities as separate systems. In the 60s, an information system was composed of a number of standalone applications. In the 70s, data was pushed out of the applications, and Database Management Systems (DBMSs) were developed. In the 80s, user interface was pushed out of the applications and User Interface Management Systems (UIMSs) were developed [5]. In the 90s, workflows were tried to be pushed out of the applications as WFMSs (Workflow Management Systems). However, this has not been realized fully. Consequently, WFM systems have not been used as a separate tool in building systems. Successful workflow applications are limited to specific industries such as banking and insurance. However, in reality, most of the requirements in organizations seem to be suitable for workflows [1].

There are two reasons why WFMS were not successful. One is technical, the other is conceptual. The technical reason is that the WFMS required some component orientation. However, there were no easy orchestration technologies available such as web services. The conceptual reason is the lack of good standards for workflow modeling [1].

2.4. WFM versus BPM

The terms WFM and BPM are being confused. BPM can be considered as an extension of traditional WFM systems. According to Gartner, BPM is a process-oriented management discipline. It is not a technology. Workflow is a flow management technology found in business process management suites (BPMSs) and other product categories [9]. The main difference between them is the diagnosis phase in the lifecycle of the BPM. In short, BPM extends the traditional WFM systems by adding the diagnosis phase.

2.5. The Diagnosis Phase and Related Technologies

The diagnosis phase extends the WFM systems as BPM systems. The diagnosis phase is also called as business process analysis (BPA) [1]. Processes should adapt to the ever-changing environment. Therefore, they should be changed dynamically. This can be done using BPA. In BPA, as well as diagnosis, simulations are done. BPA is used to optimize the business processes.

Many BPM suites offer inbuilt diagnosis tools that do not have many adequate reporting features. Consequently, companies have to rely on external reporting tools like Crystal Reports and Microsoft Reporting Services. A complete industry standard would be very beneficial for BPM. Moreover, data mining contributes to diagnosis. Especially, process mining researches can be applied to address the needs of diagnosis standards [12].

Business processes are defined, deployed, executed and monitored in BPM systems. In the diagnosis phase, special query languages are used to query processes and monitoring systems take role to monitor processes. Monitoring of business processes is a critical activity in modern enterprises because of their central role in carrying out business activities.

Business Process Query Language (BPQL) [23] is a query language for querying business process specifications. The BPQL language is based on an intuitive model of business processes, an abstraction of the emerging BPEL standard. It allows users to query business processes visually, in a manner very analogous to how such processes are typically specified, and can be employed in a distributed setting, where process components may be provided by distinct providers.

BP-Mon (Business Processes Monitoring) is another query language for monitoring business processes, which allows users to visually define monitoring tasks and associated reports, using a simple intuitive interface, similar to those used for designing BPEL processes. In [15], the BP-Mon language and its underlying formal model are described. Also, the language implementation is presented and novel optimization techniques are described. An important feature of the implementation is that BP-Mon queries are translated to BPEL processes that run on the same execution engine as the monitored processes.

An instance of a business process specification is an actual running process which includes specific decisions, real actions, and actual data. BPM systems allow to trace process instances – the activities they perform, messages sent or received by each activity, variable values, performance metrics – and send this information as events (in XML format) to monitoring systems (often called BAM – Business Activity Monitoring – systems) [15].

There is a similar concept with BPA, which is Business Process Improvement (BPI). BPI is a systematic approach to optimize an organization's processes to achieve more efficient results. In the paper [20], BPI methodologies are discussed. It shows the existing BPI methodologies, analyzes their effectiveness and their use of benchmarking, and develops a new BPI methodology. BPI is closely related to the diagnosis phase. They both try to optimize processes. There are very fast changes in the technology and standards. Optimization is realized with change. So, existing applications should be changed using new technologies to adapt to the changing environment.

Another concept close to BPA is Business Process Reengineering (BPR). Radical change of business processes is called BPR. BPR is a management practice that aims to improve the efficiency of the business process. The key to BPR is for organizations to look at their business processes from a "clean slate" perspective and determine how they can best construct these processes to improve how they conduct business. Reengineering is a fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in cost, quality, speed, and service. BPR combines a strategy of promoting business innovation with a strategy of making major improvements to business processes so that a company can become a much stronger and more successful competitor in the marketplace. Although BPM does not include BPR, in business process software development methodology the techniques used in BPR can be used [8].

2.6. The BPM Lifecycle

BPM can be considered as an extension of traditional WFM systems. The main difference between them is hidden in the lifecycle of the BPM. Figure 1 shows the lifecycle of a BPM [1]. The BPM lifecycle starts with the design phase, continues with the configuration and enactment phases, and ends with the diagnosis phase. After the diagnosis phase, the lifecycle returns to the design phase and these phases are iterated in cycles.

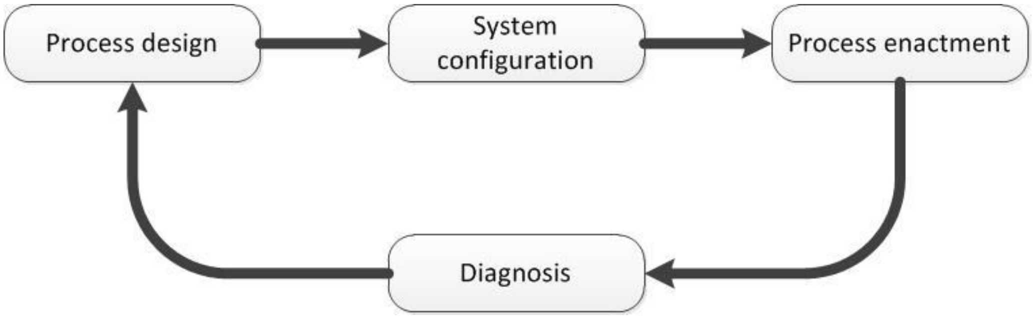


Figure 1: BPM lifecycle

The upper part of the figure shows the topics of a traditional WFM system. The diagnosis part belongs to the BPM and is used to optimize the business processes.

Now, the BPM lifecycle will be examined again. In the design phase, the business process is designed. In the configuration phase, the business process is implemented and configured for execution. In the enactment phase, business process is executed according to the configuration. In the diagnosis phase, the process is analyzed to identify the problems and to improve the process.

The traditional WFMS focuses on the upper part of the figure. However, to adapt to the ever-changing environment, the processes should also be changed dynamically. In other words, the business processes should be optimized. As a result, these requirements can only be fulfilled by BPM.

In the design phase, business processes can be modeled with formal languages and graphical notations. In the configuration phase, the models are executed.

2.7. Formal Languages

Business process models should be understood by all the stakeholders in a correct manner. The meaning of the model should be same for all of them. Therefore, formal representations may be used to model business processes.

The vast majority of business process modeling efforts lack formal methods for verifying properties of processes. In [17], a formalism is presented that can be used to represent knowledge about organizations and their business processes. Also, a methodology is discussed that enables business analysts to go from high-level enterprise objectives, to detailed and formal specifications of business processes for realizing these objectives. The methodology can be used by an enterprise that wishes to develop a new business process, or alternatively model, document and analyze formally an existing process.

The following formal languages may be used to model business processes: Petri nets, pi-calculus, situation calculus, and concurrent programming.

Petri net [5] is a state-based description of workflow procedure. In other words, it is a process modeling diagram. The standard workflow diagrams are event-based diagrams and they suppress the states. In Petri net, states are explicitly showed. Circles are used to represent states and squares are used to represent tasks. In Petri net, states are called places and tasks are called transitions.

States are important because state-based description allows for a clear distinction between the enabling of a task and the execution of a task. Since enabling of a task does not imply that the task will be executed immediately, it is important to have this distinction. A task is enabled only when the immediate input places have tokens. Only an enabled task can be executed.

Pi-calculus is a language that defines concurrent processes that interact with one another dynamically [10]. Pi-calculus provides a way of modeling for a wide variety of process management systems, in the same way that differential calculus provides a general modeling of electrical systems and the relational calculus underpins database systems.

The generality of the Pi-calculus gives the third wave of process management its inherent ability to capture, describe and manage whole processes—not just integration between existing algorithmic procedures written in conventional software languages and embodied in today's packaged software. This approach to process representation can be applied to a wide range of problems [16].

The Situation Calculus is a logic formalism designed for representing and reasoning about dynamical domains. The Situation Calculus represents changing scenarios as a set of first-order logic formulae. The basic elements of the calculus are actions that can be performed in the world, fluents that describe the state of the world, and situations.

Situation Calculus is a formal language use in the field of Artificial Intelligence. In [17], enterprise knowledge is represented using the formalism of Situation Calculus.

ConGolog is another formal language. ConGolog is a concurrent programming language based on logic developed by the Cognitive Robotics group of the University of Toronto. ConGolog was originally developed as a high-level language for programming robots and software agents. Recently, it has also been used for business process modeling and analysis [17].

2.8. Graphical Notations

Graphical standards often trace their theoretical roots to Petri Nets but the actual underlying formalism is often not clear. If there are to be new graphical formalisms, these should be more theoretical [12]. Graphical standards are currently the most human-readable and easiest to comprehend without prior technical training. Business Process Modeling Notation (BPMN), UML Activity Diagrams (UML AD), Event-driven Process Chains (EPC), Role-Activity Diagrams (RADs) and flow charts are common techniques used to model business processes graphically.

Although BPMN, UML AD, and EPC are easy to use and compactly depict business processes, their notation are not the most intuitive to humans. This causes end-users to fall back on the less expressive but easy-to-use RADs and flow charts. Greater effort is needed to foster more widespread learning and adoption of BPMN, UML AD, and EPC notations and symbols [12].

BPMN consists of one diagram called the Business Process Diagram (BPD). BPMN's BPD is used to model complex business processes [4].

UML activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Activity diagrams are constructed from a limited repertoire of shapes, connected with arrows. The most important shape types are: rounded rectangles representing activities, diamonds representing decisions, bars representing the split or join of concurrent activities, a black circle representing the start of the workflow, and an encircled black circle representing the end. Arrows run from the start towards the end and represent the order in which activities happen.

An EPC is a type of flowchart used for business process modeling. EPCs can be used for configuring an Enterprise Resource Planning (ERP) implementation and for BPI. It is used by many companies for modeling, analyzing, and redesigning business processes.

An EPC is an ordered graph of events and functions. It provides various connectors that allow alternative and parallel execution of processes. Furthermore it is specified by the usages of logical operators, such as OR, AND, and XOR. A major strength of EPC is claimed to be its simplicity and easy-to-understand notation. This makes EPC a widely acceptable technique to denote business processes.

RADs are a notation originally developed for software process modeling. RADs can be considered to be a state of the art single paradigm process modeling approach, and are well known among the process modeling community. Any business process consists of a number of distinct, concurrent activities corresponding to the many contributing ‘roles’. It is this notion of roles and the interactions between them that form the basis for RAD descriptions. Such a description of a world of roles is intended to exploit the notion of concurrently executing agents all coordinating to achieve a common goal.

2.9. Execution Languages

There are also business process execution languages. The prominent ones are Business Process Modeling Language (BPML), and Business Process Execution Language (BPEL). BPML is actually a superset of BPEL. In other words, BPML is also a business process modeling language. These two languages can execute business process models designed with BPMN [4]. Of the two, BPEL is more widely adopted in several prominent software suites (e.g. IBM Websphere, BEA AquaLogic BPM Suite, SAP Netweaver, etc.) even though BPML can better address business process semantics [12].

Technically, BPEL can be seen as an XML-programming language for Web Service compositions. The recent BPEL standard provides an XML-based language to describe both the interface exposed by a process, and its full operational logic and execution flow. Since the BPEL syntax is quite complex, commercial vendors offer systems that allow designing BPEL specifications via a visual interface, using an intuitive view of the process, as a graph of activity nodes connected by control flow edges. Designs are automatically converted to BPEL specifications.

BPML is an XML process definition language that describes the structural representation of a process and its execution semantics [10]. The block-oriented constructs enable a BPML business process to be programmed, making BPML the leading light of the Process-Oriented Programming paradigm. It is important for BPM practitioners to note that, in BPML, recursive block structures play a significant role in scoping issues that are relevant for declarations, definitions and process execution. Flow control is also handled entirely by block structure concepts (e.g. executing all the activities in the block sequentially).

Typical business applications today are not adaptable. New versions have to be developed whenever the processes being supported change. Applications written in BPML will be direct representations of business processes. By producing applications in BPML, within the framework of a suitable business process management system, organizations should be able to produce applications which adapt to organizational change. Process and application effectively will converge, the mutability of each reflected in the other [16].

2.10. Business Process Modeling

One should clearly differentiate between process definition and process execution. Process definition is the design part whereas process execution is the enactment of tasks using a process engine.

Business process models should be simple. All the stakeholders should understand them in a straightforward manner. In addition, all the stakeholders should capture the same meaning from the models. Business process modeling can be divided into four modeling categories. These are process modeling, organizational modeling, data modeling, and activity modeling.

Process modeling describes the control flow of the business process modeling. Roles and actors in a business process are represented by organizational modeling. Data modeling is a method used to define and analyze data requirements needed to support the business processes of an organization.

2.11. Orchestration and Choreography

Nowadays, there are many applications developed for distinct purposes. For example, there are many operating systems, office tools, and custom applications. However, those applications are not fully suitable for business processes. Because, business processes requires the cooperation of many applications [1]. Therefore, the challenge is to orchestrate and to glue existing pieces of software to achieve an organizational goal. Orchestration can be realized by BPEL. Technically, BPEL can be seen as an XML-programming language for Web Service compositions [12]. Orchestration part of BPM resembles component orientation. Therefore, SOA, XML and web services have contributed to the advance of BPM technology [2].

BPM and SOA are related. They link the business with the IT department. BPM is a top-down approach whereas SOA is a bottom-up approach [2]. According to Gartner, BPM organizes people for greater agility, while SOA organizes technology for greater agility [9].

The development in the web services technology makes orchestration easier. Actually, web services composition languages such as BPEL and BPML can be used to glue services defined using the Web Services Description Language (WSDL).

WSDL is an XML-based language that provides a model for describing web services. WSDL defines services as collections of network endpoints, or ports. The WSDL specification provides an XML format for documents for this purpose. Abstract definitions of ports and messages are separated from their concrete use or instance, allowing the reuse of these definitions. A port is defined by associating a network address with a reusable binding, and a collection of ports defines a service. Messages are abstract descriptions of the data being exchanged, and port types are abstract collections of supported operations. The concrete protocol and data format specifications for a particular port type constitutes a reusable binding, where the operations and messages are then bound to a concrete network protocol and message format. In this way, WSDL describes the public interface to the web service.

WSDL is often used in combination with SOAP and an XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what operations are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the operations listed in the WSDL.

Service choreography is a form of service composition in which the interaction protocol between several partner services is defined from a global perspective. That is, at run-time each participant in service choreography executes its part of it according to the behavior of the other participants. Choreography's role specifies the expected messaging behavior of the participants that will play it in terms of the sequencing and timing of the messages that they can consume and produce.

Service choreography is better understood through the comparison with service orchestration. On one hand, in service choreographies the logic of the message-based interactions among the participants is specified from a global perspective. In service orchestration, on the other hand, the logic is specified from the local point of view of one single participant, called the orchestrator. In the service orchestration language BPEL, for example, the specification of the service orchestration (e.g. the BPEL process file) can be deployed over the service infrastructure (for example a BPEL execution engine like Apache ODE). The deployment of the service orchestration specification creates the composed service.

2.12. The Concept of Business Process Software Development Methodology

In business process software development the specific properties of workflow applications should be considered although the general structure of the development methodology is based on software engineering process models [8].

Typically, different perspectives are covered in workflow schemas: The functional perspective describes what has to be done within a workflow. The operational perspective determines how it is done, i.e., which methods, techniques and tools are used to perform a given workflow. The behavioral perspective defines the behavior of the workflow, i.e., it specifies when and under which conditions a workflow is executed. The informational perspective specifies the data objects which are being manipulated during workflow executions and the flow of data between workflow activities. The organizational perspective describes how the workflow is embedded in the context of the organization, both in terms of personnel and information infrastructure. This perspective is often covered by roles, which specify properties of personnel and software systems. When a workflow is performed, a technique called role resolution is used to assign persons or software systems to workflow activities.

2.13. Workflow and Communication Patterns

Business processes have some patterns. In the book [10], P4 (Aalst, Hofstede, Kiepuszewski, and Barrios) documented 20 process patterns. These patterns may be used to evaluate the modeling tools. In the paper [6], these patterns are used to evaluate the BPML. The capabilities and limitations of BPML are showed. In addition to this, BPEL, XLANG, WSFL, WSCI, and BPML are compared using the same set of patterns. Despite BPML being a formally complete business process standard, it is no longer supported by its founding organization BPMI after its merger with OMG in 2005 [12].

CHAPTER 3

DATA ANALYSIS OF THE WATERFALL METHODOLOGY

In software engineering, the domain of the software is important. The research domain is basically business processes. Moreover, the development team deals with more than one project concurrently. In [37], the importance of the domain for software estimations is emphasized. Therefore, to determine the boundaries of the domain is crucial. In this chapter, the characteristics of the domain are analyzed using the collected information in the organization.

3.1. Analysis of Help Desks Data

In the organization, user requests come through a help desk. This help desk keeps data about the requests and the actions related with them. For example, it keeps what the request is, when it is requested, and how much time spent for the request. This help desk keeps business process related requests also. This current help desk has been being used in the organization for approximately one year. Before this, there was the old help desk and it had been used for approximately 5 years. The two help desks have different databases which have different data models. In order to obtain all the data, these two databases should be combined. Then the aggregated data can be used to analyze the business processes which are developed. Firstly, this data will be used for comparing the waterfall model and the proposed methodology. Secondly, this data will be used to compare the methodologies in different aspects according to multi-dimensional analysis [25].

3.1.1. Combining Help Desks

Combining these databases was difficult because the data models are different. Actually, the old database has missing information according to the new data model. For example, for a request in the former help desk only one value is kept for effort whereas in the latter many values are kept for each person. In addition, some status information was changed in the new help desk.

Another big difficulty was to extract the requests which are related to business processes from the help desks. In other words, there is no exact field showing business process requests. Therefore, text search is performed to find business process requests. Another text search is also conducted to categorize the located requests.

Both help desks have the same following fields: request id, request date, free text, and responsible person.

There are similar fields in both of the help desks. The first one is request type which shows whether the request is an issue or a problem. In the old help desk, this field has values as issue, problem, and suggestion, or it has no value for many requests. In the new help desk, there are only two values as issue and problem. I combine these fields as taking only the two values issue and problem. In other words, fields having no value or value “suggestion” are considered as an issue.

Topic fields of help desks are similar. I found similar three topic fields in the help desks. In the old help desk topics are broken down in three levels with the three topic fields. In the new help desk, topics are broken down in many levels hierarchically in two viewpoints as service-oriented and ordinary. The top level topic and the last levels of the two viewpoints are taken as three topic fields.

In both help desks, there are status fields. However, these fields keep different status values. In the old help desk, there are 9 status values whereas in the new help desk, there are 19 status fields. I combine these status fields as taking only four status values which are completed, canceled, rejected and on-going.

When I combine fields, I use general values for combined fields. Using general values removes specialized needs of the company from the data.

The priority fields of both help desks have different levels of priority. The old help desk has 3 priority levels as low, medium, and high. However, the new help desk has 12 priority levels. I combine these levels into three priority levels as low, medium, and high. If there is no assigned level for priority, I accept it as medium.

Table 1 summarizes the consolidation work described above. It shows help desk fields and the combined fields.

Table 1: Common fields of the help desks

Old Help Desk	New Help Desk	Combination
TALEP_ID	TALEP_ID	REQUEST_ID
TALEP_TARIHI	TALEP_TARIHI	REQUEST_DATE
ACIKLAMA	ACIKLAMA	FREE_TEXT
SORUMLU	SORUMLU	RESPONSIBLE_PERSON
TIP	TALEP_TURU	REQUEST_TYPE
TALEP_UST_TURU	TALEP_SINIFI	MAIN_TOPIC
TALEP_TURU	SERVIS_TURU	TOPIC
KONU	KONU_TURU	SUBTOPIC
DURUM	DURUM	STATUS
ONCELIK	ONCELIK_BYD	PRIORITY

Both help desks keep efforts in minutes. The old help desk keeps only one effort value for a request. However, the new help desk can keep many effort values for each person and for each week in the calendar. Consequently, the duration of processing a request can be obtained from the new help desk. Since there may be many effort values for a request, I keep them in a separate table. Table 2 shows the structure of the efforts table.

Table 2: Efforts table

EFFORTS
REQUEST_ID
EFFORT_DATE
EFFORT

In the new help desk, there is an importance field entered by customers. This field shows the urgency of the request according to the customer. Therefore, this field may show the attitude of the customer to the request. The new help desk also has an urgency field. This represents the real urgency of the request. Moreover, there is also impact field in the new help desk. Table 3 shows these three fields.

Table 3: The fields only existing in the new help desk

New Help Desk	**Combination**
ONEM_KUL	CUSTOMER_URGENCY
ONEM_BYD	URGENCY
ETKI_BYD	IMPACT

In Table 4, the numbers of the records in the help desks are shown. There are 78297 requests in the old help desk, whereas in the new help desk, there are 21620 requests.

Table 4: Numbers of records in the help desks

	Old Help Desk	New Help Desk
# of Records	78297	21620

Extracting requests about business processes has been difficult because there is no exact field showing the business process requests. Therefore, for the old help desk, I took the requests having the values:

- “Yeni İş Akışı Talebi”, (New workflow request)
- “İş Akışı Sorunları”, and (Workflow problems)
- “İş Akış Talebi” (Workflow request)

in the field topic, and has the values:

- “İş Akışı Talepleri” (New workflow requests)

in the main topic field. Moreover, the requests whose free text fields contain the values:

- “iş akış”, (work flow)
- “işakış”, (workflow)
- “uçak talep”, (plane ticket)
- “mhf”, (material request)
- “malzeme hareket form”, (material request form) and
- “sigorta talep” (insurance request)

are taken also as business process requests. These keywords are related terms for business processes in Turkish. For the new help desk, again the same keywords are used for the free text field to extract the business process requests. And the requests whose main topic field has the entries “Mevcut İş Akışında Değişiklik” or “Yeni İş Akışı” are taken as business process requests from the new help desk. Table 5 shows all these things in a tabular format.

Table 5: The criteria for the extraction of business process requests

Combined Field Name	Operation	Old Help Desk	New Help Desk
MAIN_TOPIC	**equal**	"İş Akışı Talepleri"	"Mevcut İş Akışında Değişiklik"
			or "Yeni İş Akışı"
TOPIC	**equal**	"Yeni İş Akışı Talebi"	
		or "İş Akışı Sorunları"	
		or "İş Akışı Talebi"	
FREE_TEXT	**contains**	"iş akış"	"iş akış"
		or "işakış"	or "işakış"
		or "uçak talep"	or "uçak talep"
		or "mhf"	or "mhf"
		or "malzeme hareket form"	or "malzeme hareket form"
		or "sigorta talep"	or "sigorta talep"

From the old help desk, 536 requests are extracted using the above criteria. On the other hand, 208 requests are extracted from the new help desk. Table 6 shows the number of the extracted business process requests and the total number of requests.

Table 6: Total numbers of business process requests

	Old Help Desk	New Help Desk
Total # of Records	78297	21620
Total # of Business Process Requests	536	208

For the business processes, the efforts spent have been keeping also. These records can be used in the analysis of the help desk data. In other words, the table structure of Table 7 can be added to the combined help desk tables. Business process requests are categorized according to their types. This process is nearly manual except scanning the complete names of the business process types. This is done manually because nearly all the analysis depends on these categories. Business process related requests are marked using BUSINESS_PROCESS_RELATED field in the combined requests table. For each related request, the type of the business process is kept in the BUSINESS_PROCESS_TYPE field in the combined requests table. This field has the following values “MHF”, “SAT”, “SIGORTA”, “SAS”, and “DIGER”. The value “DIGER” is for uncategorized request whereas the others are the related business process types.

Table 7: Business process types

BUSINESS_PROCESS_TYPES
BUSINESS_PROCESS_TYPE
TOTAL_EFFORT_KEPT

3.1.2. Building an OLAP Cube for Detailed Analysis of Help Desks

For detailed analysis of the help desks, I will use multidimensional data cube [24] [25]. Table 8 shows the basic table for the cube. There are dimension fields and facts. HELP_DESK_TYPE field has values as “OLD” or “NEW” according to the related help desk. RESPONSIBLE_PERSON field is taken as a dimension because people can be categorized as members of analysis and development groups. The fields beginning with “ORIGINAL_” hold the original value of the related help desk. These values can be used to rearrange the categories of the dimensions.

Table 8: Basic cube table

REQUESTS	
HELP_DESK_TYPE	DIMENSION
REQUEST_DATE	DIMENSION
RESPONSIBLE_PERSON	DIMENSION
REQUEST_TYPE	DIMENSION
STATUS	DIMENSION
PRIORITY	DIMENSION
CUSTOMER_URGENCY	DIMENSION
URGENCY	DIMENSION
IMPACT	DIMENSION
BUSINESS_PROCESS_RELATED	DIMENSION
BUSINESS_PROCESS_TYPE	DIMENSION
EFFORT_ID	DIMENSION
REQUEST_ID	FACT
FREE_TEXT	FACT
MAIN_TOPIC	FACT
TOPIC	FACT
SUBTOPIC	FACT
ORIGINAL_RESPONSIBLE_PERSON	FACT
ORIGINAL_REQUEST_TYPE	FACT
ORIGINAL_STATUS	FACT
ORIGINAL_PRIORITY	FACT
ORIGINAL_CUSTOMER_URGENCY	FACT
ORIGINAL_URGENCY	FACT
ORIGINAL_IMPACT	FACT

Table 9: Dimension tables

EFFORTS	
EFFORT_ID	PRIMARY KEY
REQUEST_ID	
EFFORT_DATE	
EFFORT	
BUSINESS_PROCESS_TYPES	
BUSINESS_PROCESS_TYPE	PRIMARY KEY
TOTAL_EFFORT_KEPT	
START_DATE	
END_DATE	
DESIGN_START_DATE	
CONSTRUCTION_START_DATE	
PILOT_START_DATE	
LIVE_START_DATE	
HELP_DESK_TYPES	
HELP_DESK_TYPE	PRIMARY KEY
REQUEST_DATES	
REQUEST_DATE	PRIMARY KEY
RESPONSIBLE_PERSON_TYPES	
RESPONSIBLE_PERSON	PRIMARY KEY
RESPONSIBLE_PERSON_TYPE	
REQUEST_TYPES	
REQUEST_TYPE	PRIMARY KEY
STATUS_TYPES	
STATUS	PRIMARY KEY
PRIORITY_TYPES	
PRIORITY	PRIMARY KEY
CUSTOMER_URGENCY_TYPES	
CUSTOMER_URGENCY	PRIMARY KEY
URGENCY_TYPES	
URGENCY	PRIMARY KEY
IMPACT_TYPES	
IMPACT	PRIMARY KEY
BUSINESS_PROCESS_RELATED_TYPES	
BUSINESS_PROCESS_RELATED	PRIMARY KEY

Table 9 shows the dimension tables.

3.1.3. The Built Multi-dimensional Cube

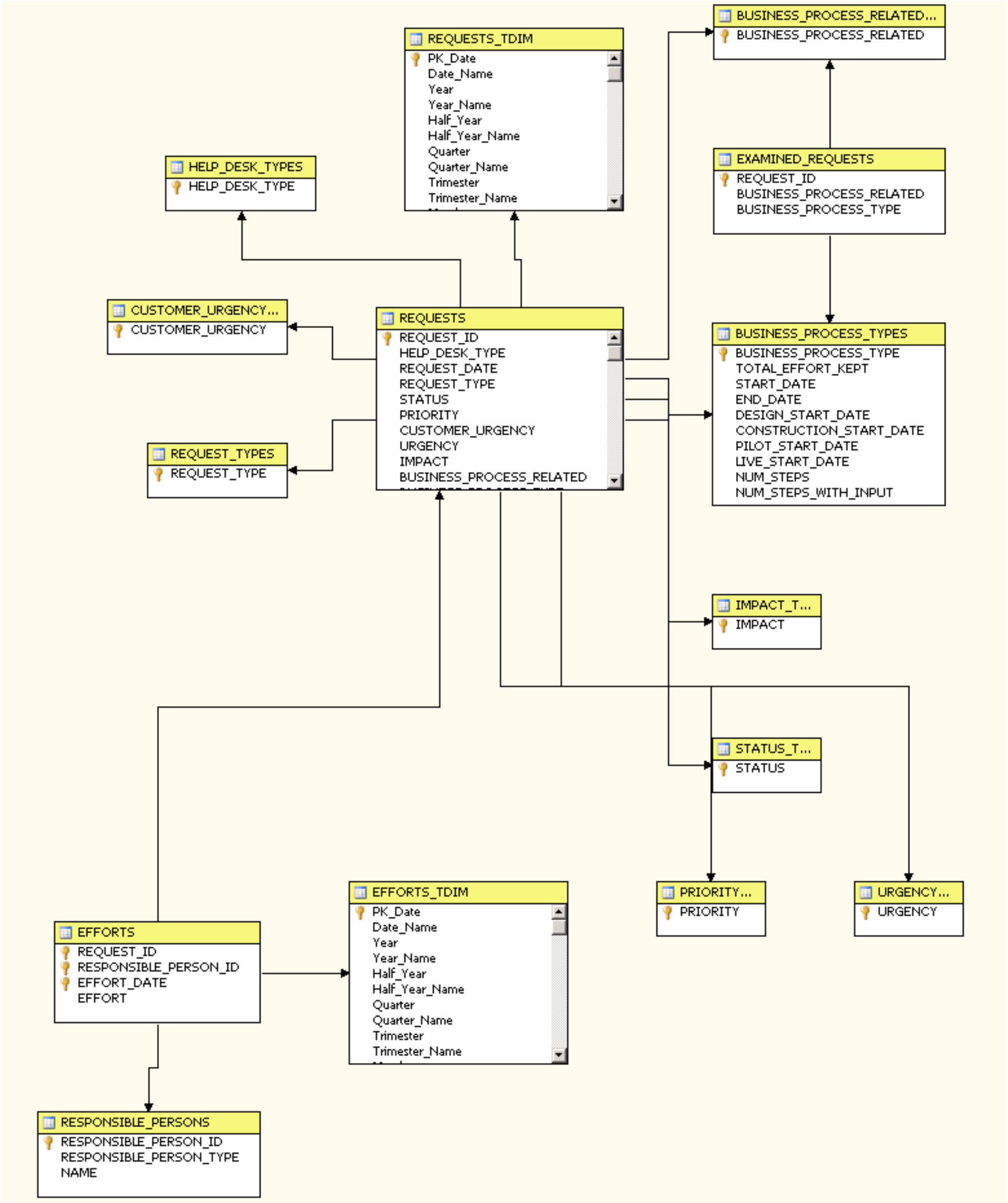


Figure 2: The structure of the multi-dimensional cube

I have built a multi-dimensional cube from the organization’s help desk databases. There are two help desk systems in the company. One is the old one and had been used until the new help desk is implemented. These two help desk systems are keeping similar data with some differences. Actually the new one keeps more structured and detailed data. The structures of the systems were explained in the previous section. In that section, also how the business process related data are extracted is described.

Figure 2 shows the combined data model of the two help desk databases. At the upmost level, there is a table called EFFORTS. Table 10 shows the structure of the table.

Table 10: Structure of the table EFFORTS

COLUMN	COMMENT	BASE TABLE
REQUEST_ID	DIMENSION	REQUESTS
RESPONSIBLE_PERSON_ID	DIMENSION	RESPONSIBLE_PERSONS
EFFORT_DATE	DIMENSION	
EFFORT	FACT	

The table EFFORTS keeps the efforts spent by the personnel. It keeps the effort based on the related request and the personnel dealing the request with a date. Therefore, for a request there can be more than one personnel who spend effort. This table is the main fact table and according to the dimension tables the efforts spent are available to be analyzed.

Table 11: Structure of the table RESPONSIBLE_PERSONS

COLUMN	COMMENT	BASE TABLE
RESPONSIBLE_PERSON_ID	PRIMARY KEY	
RESPONSIBLE_PERSON_TYPE		
NAME		

Table 11 shows the structure of the table RESPONSIBLE_PERSONS. This table keeps the personnel and his/her job type in the development of the applications. Basically, the personnel are divided into two categories which are ANALYSIS and DEVELOPMENT. The developers are categorized as DEVELOPMENT and the other personnel are categorized as ANALYSIS.

Table 12: Structure of the table REQUESTS

COLUMN	**COMMENT**	**BASE TABLE**
REQUEST_ID	PRIMARY KEY	
HELP_DESK_TYPE	DIMENSION	HELP_DESK_TYPES
REQUEST_DATE	DIMENSION	
REQUEST_TYPE	DIMENSION	REQUEST_TYPES
STATUS	DIMENSION	STATUS_TYPES
PRIORITY	DIMENSION	PRIORITY_TYPES
CUSTOMER_URGENCY	DIMENSION	CUSTOMER_URGENCY_TYPES
URGENCY	DIMENSION	URGENCY_TYPES
IMPACT	DIMENSION	IMPACT_TYPES
BUSINESS_PROCESS_RELATED	DIMENSION	BUSINESS_PROCESS_RELATED_TYPES
BUSINESS_PROCESS_TYPE	DIMENSION	BUSINESS_PROCESS_TYPES
FREE_TEXT		
FREE_TEXT_UPPER		
MAIN_TOPIC		
TOPIC		
SUBTOPIC		
ORIGINAL_REQUEST_TYPE		
ORIGINAL_STATUS		
ORIGINAL_PRIORITY		
ORIGINAL_CUSTOMER_URGENCY		
ORIGINAL_URGENCY		
ORIGINAL_IMPACT		

The requests in the help desk systems are kept in the table REQUESTS. Table 12 shows the structure of the table REQUESTS. For each request, a record is kept in this table. This table is the second most important fact table of the cube. According to the dimensions the request counts can be analyzed. The column HELP_DESK_TYPE tells whether the request comes from the old or the new help desk system. The column REQUEST_DATE keeps the creation date of the request. The column REQUEST_TYPE keeps two different values which are PROBLEM and ISSUE. All the requests which are about problems are categorized as PROBLEM and all the other ones are categorized as ISSUE. The column STATUS keeps four different values which are ON-GOING, COMPLETED, CANCELED, and REJECTED. The meanings of the status values are clear and they represent the related state for the request. The column PRIORITY keeps three different priority values which are LOW, MEDIUM, and HIGH. This three level category is also used for the columns CUSTOMER_URGENCY, URGENCY, and IMPACT. Actually, the field PRIORITY is intended to keep the related priority level based on the urgency and impact values of the request. The column CUSTORMER_URGENCY is for the urgency level according to the customer who has created the request. The column URGENCY is the interpreted state of the column CUSTOMER_URGENCY by the personnel. The column IMPACT keeps the impact level of the request. The column BUSINESS_PROCESS_RELATED keeps whether the request is related to business processes or not. The column BUSINESS_PROCESS_TYPE keeps the related business process if the request is categorized as business process related.

Table 13: The structure of the table BUSINESS_PROCESS_TYPES

COLUMN	COMMENT	BASE TABLE
BUSINESS_PROCESS_TYPE	PRIMARY KEY	
TOTAL_EFFORT_KEPT		
NUM_STEPS		
NUM_STEPS_WITH_INPUT		

All the business processes which have been developed and have been used or not in the company are kept in this table BUSINESS_PROCESS_TYPES. Table 13 shows the structure of the table. The values of the table are shown in the Table 14. There are information about the business processes named as MHF, SAS, SAT, and SIGORTA. The type DIGER is used for uncategorized requests.

Table 14: Values of the table BUSINESS_PROCESS_TYPES

BUSINESS_PROCESS_TYPE	TOTAL_EFFORT_KEPT	NUM_STEPS	NUM_STEPS_WITH_INPUT
DIGER	NULL	NULL	NULL
MHF	102	5	4
SAS	101	12	0
SAT	130	14	0
SIGORTA	51	4	0

3.2. The Results of the Analysis

Table 15: Business process related requests

BUSINESS PROCESS RELATED	REQUESTS Count	EFFORT_DAYS
NOT_RELATED	114228	11657.5
RELATED	812	732.82
Grand Total	115040	12390.32

In the Table 15, general information about the request is given. 115040 requests have been recorded. 812 of them are business process related requests. For these 812 requests 732.82 person days have been spent.

From this point forward, only these 812 business process related requests will be analyzed. For each request, 0.9 person day in effort has been spent. In other words, approximately one person day has been spent for a request.

Table 16: Efforts according to request types

REQUEST TYPE	REQUESTS Count	EFFORT_DAYS
ISSUE	651	668.83
PROBLEM	161	63.98
Grand Total	812	732.82

In the Table 16, efforts are shown according to their request types. Of 812 requests, 161 are of type problem. 19.82% of the requests are of type problem. In other words, approximately 20% of the requests are because of problems. 8.73% of the total effort has been spent for problems. Approximately 9% of the effort is spent for problems. 1.02 person-days for an issue and 0.39 person-days for a problem have been spent. In other words, for each issue approximately 1 person day has been spent. The effort for an issue has been spent approximately 2.5 times as much as for a problem.

Table 17: Analysis and development times

RESPONSIBLE PERSON TYPE	EFFORT_DAYS
ANALYSIS	110.94
DEVELOPMENT	621.88
Grand Total	732.82

In the Table 17, analysis and development efforts are shown. 15.13% of the effort has been spent for analysis. In other words, approximately 15% of the effort is spent for analysis whereas 85% of the effort is spent for development of business processes.

Table 18: Priority of requests

PRIORITY	REQUESTS Count	EFFORT_DAYS
HIGH	8	19.6
LOW	94	15.83
MEDIUM	710	697.38
Grand Total	812	732.82

In the Table 18, the count of requests and efforts spent are shown for each priority level. 0.98% of the requests are of high priority and 87.43% are of medium priority. In other words, approximately 1% is high in priority and 90% is medium in priority.

Table 19: Status of requests

STATUS	REQUESTS Count	EFFORT_DAYS
CANCELED	102	8.55
COMPLETED	651	706.8
ON-GOING	33	15.98
REJECTED	26	1.48
Grand Total	812	732.82

In the Table 19, information about the requests is shown according to status. 12.56% of the requests have been canceled, and 3.20% of them have been rejected, and 80.17% of them have been completed. In other words, approximately 13% are canceled, 3% are rejected, and 80% are completed.

Table 20: Impact of requests

IMPACT	REQUESTS Count	EFFORT_DAYS
LOW	72	14.03
MEDIUM	740	718.78
Grand Total	812	732.82

In the Table 20, impact of requests is shown. 91.13% of the requests have come as medium in impact. In other words, approximately 90% of the requests are medium in impact.

Table 21: Urgency of requests

URGENCY	REQUESTS Count	EFFORT_DAYS
HIGH	17	12.66
LOW	67	9.46
MEDIUM	728	710.69
Grand Total	812	732.82

In the Table 21, urgency of requests is shown. 2.09% of the requests have been come as very urgent (high in urgency). 89.65% of the requests have been medium in urgency. In other words, approximately 2% of the requests are very urgent, and 90% of the requests are medium.

Urgency, impact, and priority values are similar for the requests with medium state. In other words, approximately 90% of the requests are categorized as medium in urgency, impact, and priority.

Table 22: Customer Urgency of Requests

CUSTOMER URGENCY	REQUESTS Count	EFFORT_DAYS
HIGH	34	4.42
LOW	5	0.76
MEDIUM	773	727.63
Grand Total	812	732.82

In the Table 22, customer urgency state of the requests is shown. 4.18% of the requests have been described as very urgent according to their owners. In other words, approximately 4% of the requests have been depicted as very urgent by customers but only half of them are accepted as very urgent.

Table 23: Urgency, impact and priority relation of requests

URGENCY	IMPACT	PRIORITY	REQUESTS Count	EFFORT_DAYS
HIGH	MEDIUM	HIGH	1	8.1
		MEDIUM	16	4.56
		Total	17	12.66
	Total		17	12.66
LOW	LOW	LOW	66	8.59
		Total	66	8.59
	MEDIUM	LOW	1	0.87
		Total	1	0.87
	Total		67	9.46
MEDIUM	LOW	MEDIUM	6	5.44
		Total	6	5.44
	MEDIUM	HIGH	7	11.5
		LOW	27	6.36
		MEDIUM	688	687.37
		Total	722	705.24
	Total		728	710.69
Grand Total			812	732.82

In the Table 24, the relation among priority, impact, and urgency is shown. In the help desk systems, priority of a request is determined according to its urgency and impact state. In other words, priority is a function of impact and urgency. This can be shown as the following:

$$\text{Priority} = \text{Urgency} * \text{Impact} \tag{1}$$

The above formula (1) describes the priority matrix. A priority matrix can be built from the values in Table 23. High in urgency and medium in impact gives high in priority with 5.88% or medium in priority for rest. Low in urgency and low or medium in impact gives low in priority. Medium in urgency and low in impact gives medium in priority. Medium in urgency and medium in impact gives high in priority with 0.96%, low in priority with 3.73%, or medium in priority for rest. In summary, the priority matrix with percentages is shown in the Table 24.

Table 24: Priority matrix with percentages

PRIORITY %		IMPACT		
		HIGH	MEDIUM	LOW
URGENCY	HIGH	NA	6% HIGH + 94% MEDIUM	NA
	MEDIUM	NA	1% HIGH + 4% LOW + 95% MEDIUM	100% MEDIUM
	LOW	NA	100% LOW	100% LOW

3.3. Discussion of the Analysis Results

The results of the analysis of the application domain roughly show the present situation of the classical methodology. The most important result is the proportion of the analysis.

It says that nearly 15% of the total effort is spent for analysis. Actually this is not a good percentage for analysis because generally and naturally business process software development is complex. It means that analysis has not been done enough. It is evaluated that the proportion should be increased to determine the requirements better. If a quarter of the total effort was spent for analysis, the change requests would be less, and the determined requirements would have a good quality. Moreover, the proportion for problems also supports this finding. It says that approximately 20% of the requests are because of problems. In other words, 80% of the requests are new issues. The latter proportion is very dominant and means that the business processes are changed very much. These results indicate that the quality requirements have not been determined enough and the implemented business processes did not fit to the customer's optimal requirements. These findings motivate to apply agile approaches in the new implementations.

CHAPTER 4

PROPOSED METHODOLOGY

In this chapter, the proposed methodology is introduced. First of all, the proposed methodology is defined. Its process model and main elements are given in the following sections.

4.1. Definition of the Proposed Methodology

This methodology is an iterative and agile methodology. It depends on the process framework where there are some framework activities and some umbrella activities. Process framework is explained in [21]. The framework activities form the stages of the methodology. These stages are analysis, design, construction, going to pilot, going to live, and diagnosis. These stages are shown in Figure 3. In Waterfall methodology, a phase ends and only after that the next phase begins. In other words, at the same time only one phase is active in a given time. Unlike Waterfall methodology, the stages in this methodology do not require ending of the previous stage. Namely, transition between stages is not strict and all the stages can exist at the same time with changing densities according to the progress of the development. This is also shown in Figure 3. From another perspective a stage actually means a set of tasks in this methodology. For example, in the figure “construction” is shown as the third stage and can exist during the entire automation. Therefore, a stage can be considered as a set of related tasks. In addition to the framework activities, there are umbrella activities which are spread over the entire methodology. In other words, umbrella activities are related to nearly all the framework activities. The umbrella activities are project management and training. Also, keeping history of the automation is important. Initial work on the proposed methodology can be found in [68].

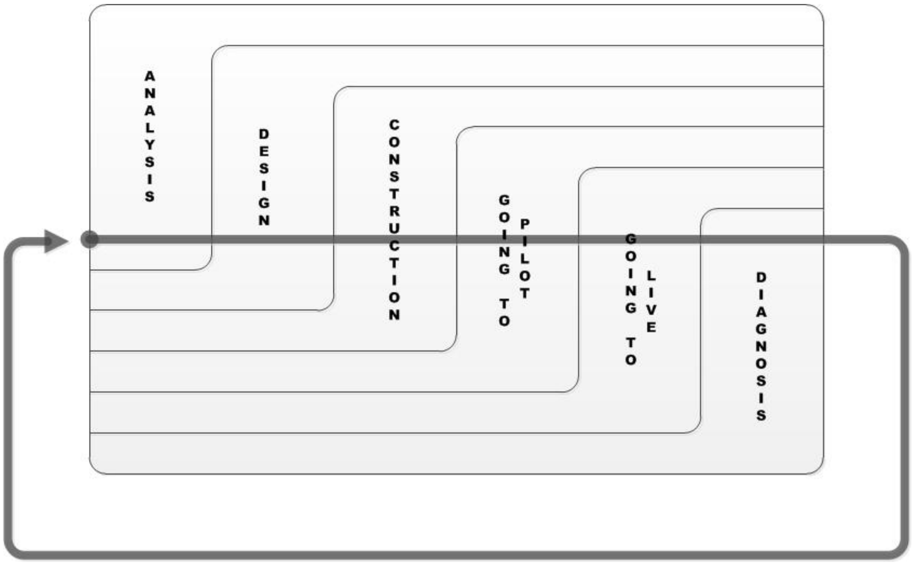


Figure 3: Stages of the methodology

4.2. The Process Model of the Methodology

In this methodology, going to pilot is taken as a stage because it has been seen that pilot application is very important for developing processes. Basically, it provides a preparation for a live and increases the success of the implementation. Moreover, it provides real problems for the development and allocates time to solve these problems in a relatively low stressed condition.

In the going to live stage, production system is configured. All users are informed about the process and user support system is defined. Then the process is deployed, and started based on the gained pilot know-how.

The diagnosis stage is also included in this methodology because the diagnosis stage is a part of the BPM lifecycle. The diagnosis stage is used to improve processes.

There are two kinds of activities. These are work tasks and work products, whose definitions can be found in [21]. Work tasks are the tasks in the related activity, whereas work products are the tasks which deliver output to the next stages. This methodology places great emphasis on modeling of the roles. Actually, determining all the actors properly accelerates the development process. Otherwise, missing actors would cause redesign of the process.

This methodology requires sponsorship. Actually, sponsorship is very significant in all projects. Especially, for business processes sponsorship is a must because usually many organizational units are included in business process software development and the number of the stakeholders is plenty. This increases the complexity of the project and the conflicts should be solved by the sponsor when needed.

4.3. Iteration Base

In this methodology, as an incremental part an iteration base is defined. All the things are done in iteration bases. The structure of an iteration base is shown in Figure 4. An iteration base is usually a 5-week period and yields an increment. i.e., the process is evolved and something is added to it. In agile methodologies, the similar task sets are implemented in approximately in a month. For example, in extreme programming a story is implemented in up to a 3-week period [32]. In scrum model, a sprint is implemented in 30-days [14]. However, according to the experiences it has been seen that a bit longer period is more suitable because there are more actors in business processes than in similar software projects. In short, a period of 5 weeks is chosen for a typical iteration base.

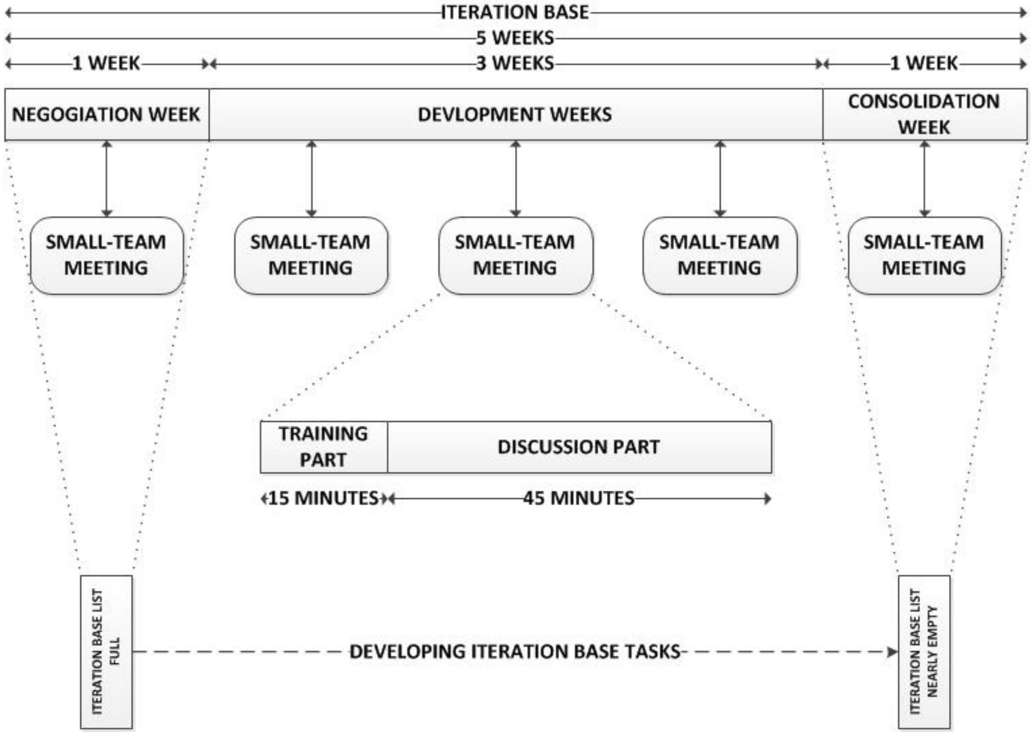


Figure 4: Structure of an iteration base

There can be more than one iteration bases at the same time. For example, there can be 3 iteration bases for the analysis stage, 2 iteration bases for the design stage, and one iteration base for construction.

In Scrum [14], the list of requirements is maintained. For each iteration base, a subset of the list is implemented, which is called an iteration base list. In other words, an iteration base list is the set of tasks in the iteration base. An iteration base list is determined before an iteration base or in the first week of the iteration base. All the tasks of a project form the project task list. The remaining tasks of the project form the backlog list. In the beginning of the project, the project task list and the backlog list are the same. New tasks can be added to the backlog list and some tasks are dropped from the backlog list if they have not been needed so far. An iteration base can take some tasks from the backlog list and uncompleted tasks remain it puts those tasks to the backlog list again. In time the backlog list diminishes. If the backlog list is finished, the project has also finished.

4.3.1. Structure of an Iteration Base

First week of an iteration base is called the negotiation week. This week of iteration base is used to negotiate with the customer, and prioritize the backlog list and determine the tasks of the iteration base, the iteration base list. The last week of iteration base is called the consolidation week. In the consolidation week, the iteration base is reviewed and the remaining tasks of the iteration base list are determined and they are added to the backlog list. The middle weeks of an iteration base are called development weeks and the tasks of the iteration base list are conducted in these weeks.

An iteration base is usually completed by a small team. Small teams are encouraged in this methodology because they are more agile and conflicts are solved easier in small teams. The same teams can be used for each iteration base or new teams can be constructed. A team can conduct more than one iteration base at the same time. The team of an iteration base is called the iteration base team.

The very first iteration base starts with a small-team meeting with the customer. General requirements are gathered in this meeting and the actors are determined. Then an iteration base is activated for each actor if needed. In addition to this, iteration bases are activated when needed such as for special organization units, or other dimensions of the process.

Small-team meetings are kept short and realized by a small number of related people. All the stakeholders can take place in these meetings at different times. These meetings are frequently organized and it is recommended a weekly period for small-team meetings.

More than one iteration base can be activated at the same time. Therefore, a person would attend more than one small-team meeting in a week. According to the experiences, more than three meetings in a week are excessive and decrease the development efforts. For this reason, the project leader should not activate more than three iteration bases related with the same person.

Small-team meetings should be short, i.e., they may take one hour. Small-team meetings are conducted in two parts. The first part is for training and lasts approximately 15 minutes. The remaining is the discussion part.

In training parts of the meetings, people are informed about the business processes. Especially, the process being developed is shown to them. If the process has not been developed enough, then prototypes of the process can be shown. If both of them are not available, then other processes can be demonstrated. In the training part, the decisions taken before are repeated also. The aim of this training part is to reduce the total development time. Even when everything is implemented, there is an adoption period of the processes in order to take them to live. Training would decrease the adoption period and also helps to gather requirements better.

The adoption period is defined as the total time to adopt the new process excluding the total time of the process development life-cycle. In other words, the life-cycle efforts plus the adoption period equals the total automation time of the process.

Usually, there should be at least one small-team meeting in every week of an iteration base. The aim of this is to train all the stakeholders and to adapt to the changing situations.

In discussion parts of the small-team meetings, requirements are gathered, use-cases and actors are determined.

There are also whole-team meetings. Nearly all the stakeholders attend these meetings. These meetings are also structured with training and discussion parts. The whole-team meetings are conducted for consolidation purposes. Different dimensions of a subject are consolidated in whole-team meetings. Actually, each different dimension is conducted in an iteration base and an iteration base is activated for consolidation of the different dimensions. In these consolidation iteration bases, whole-team meetings are organized and take common decisions about the process. The whole-team meetings are usually two times longer than small-team meetings. In other words, 30 minutes are used for training and 90 minutes are used for discussion. Totally, a whole-team meeting takes a two-hour period.

Number of the whole-team meetings should be less. In other words, these meetings should be organized when consolidation is needed or when the project needs a refreshment, i.e. drawing attention to the project again. A whole-team meeting is recommended in every 5 weeks. This is because an iteration base lasts nearly 5 weeks so at the end of the parallel iteration bases a consolidation would be required.

4.3.2. Documentation of an Iteration Base

Each iteration base is named with the keyword “Iteration Base” and a number. The first iteration base is called as “Iteration Base 1”. After this first iteration base, other iteration bases are started at some times and are numbered sequentially such as 2, 3, and 4. These numbers can be thought of as the increments of the automation. If at some time more than one iteration bases are started concurrently, then they are numbered following a decimal point. For example, after the first iteration base, if three iteration bases have started concurrently, these are named as “Iteration Base 2.1”, “Iteration Base 2.2”, and “Iteration Base 2.3”. To refer to any iteration base in the second increment, Iteration Base 2.X is used.

Each iteration base has at least one document with the same name with the iteration base. This document is called iteration base document. Being the first iteration, Iteration Base 1 document keeps the project plan of the automation and very first requirements of the automation. In this iteration base, nearly all the work is done within the department of the process owner. Although a business process crosses over many organization units, the process owner department is very important for the automation. The motivation of the process owner to execute the business process plays a key role in the success of the automation.

Usually a detailed analysis of the requirements is conducted in the iteration bases related with the second increment. For each organization unit or a group of organization units, an iteration base can be started in the second increment. Use cases are detailed in these iteration base 2.X’s. Especially, the input/output requirements of the use cases are determined.

Generally, the consolidation of the requirements is done in Iteration Base 3. In this iteration base, the work products of the Iteration Base 2.X’s are used, all the use cases are consolidated, and the process model is drawn. Moreover, the basic input output requirements of a step in the process model are determined.

An iteration base document describes the related iteration base. In an iteration base document, everything important in that iteration base is expressed. If that iteration base uses other documents, they are also referenced in the iteration base document. There is also a general document called “Iteration Base 0”. This document keeps the history of the automation. In other words, every item is added to this document with a date. This document also glues all the iteration bases in the automation. In other words, the relations and interactions between the iteration bases are explained in this document.

General information about the automation is also kept in the Iteration Base 0 document. For example, the storage locations of the documents for the projects are kept in that document. In addition to this, which tools are used and for what purposes they are used are kept in the document. Also name of the documents incorporating use case information are held in the document. The location of the process model diagram is another item for the document. The development tool and the place of the source codes of the automation are held further in the iteration base 0 document. The Iteration Base 0 document can be a means of communication if there is more than one team in the automation. All the teams can track the automation by following the Iteration Base 0 document.

All iteration base documents except the iteration base 0 document have a scope part. This part explains what the scope of the related iteration base is. Other parts of an iteration base document can be built freely by the related iteration base team.

This methodology is a light weight methodology [45]. Therefore, the documentation requirements should be kept at minimum. However, for each iteration base a simple documentation is good to track the task list of the related iteration base. All the related and required information about an iteration base should be written to this simple document. For the general and required information, again simple but a general document is kept. This document is used to record the interactions and the relations among iteration bases. Moreover, it keeps the history of the development. In other words, every item is added to this document with a date. Therefore, this document glues all the iteration bases in the development.

4.4. Stages of the Methodology

The lifecycle has 6 stages. These are analysis, design, construction, going to pilot, going to live, and diagnosis. The transition from one stage to another is not strict. In other words, the boundaries between the stages are not solid. For example, when the analysis stage has not been totally finished, the design stage can begin, and also construction stage can be started. Namely, concurrency is allowed.

The stages and their interactions with iteration bases are shown in Figure 5. Tasks of the stages are handled in iteration bases. Each iteration base is conducted with a small team that includes the customer. Training the customer is important as well as sponsorship, keeping history and project management. At the end, the product is developed that results in customer satisfaction.

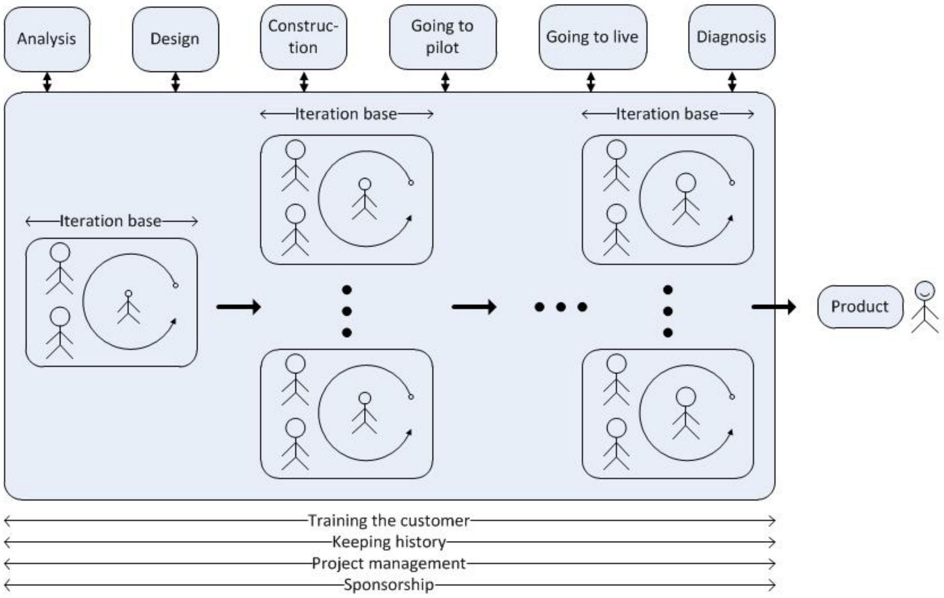


Figure 5: Process model of the proposed method based on iteration bases

Each stage has its own work tasks and work products. The work products go to the next stage as input.

4.4.1. Analysis

4.4.1.1 Gathering Requirements

Requirements are gathered through use case diagrams. Use cases are employed because:

- They are simple.
- All the stakeholders can understand them easily.
- They show actors and use cases. The optimal roles of the process can be derived from the actors of the use cases. Moreover, the optimal activities of the process can be derived from the use cases.

To show the application of the methodology a sample process is used. The sample process is shown in Figure 6.

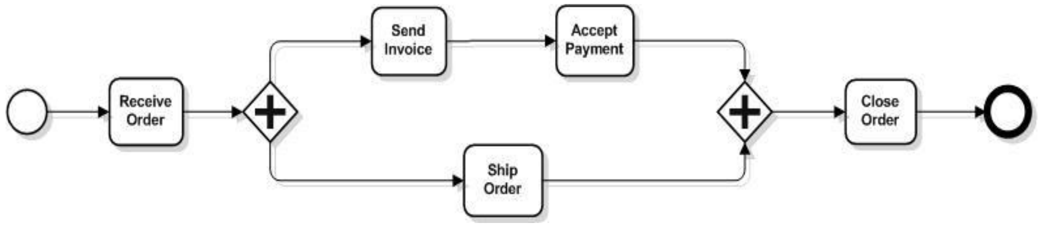


Figure 6: An example process

The example process describes a seller who receives an order and processes it. First of all, an order is received by an employee. Then, the employee sends the order to the accounting department for invoice and to the warehouse department for shipment. The accountant sends the invoice to the customer and waits for payment. At the same time, the warehouse staff starts the shipment of the order. Lastly, the archiver closes the order.

Figure 7 shows the use cases of the sample process.

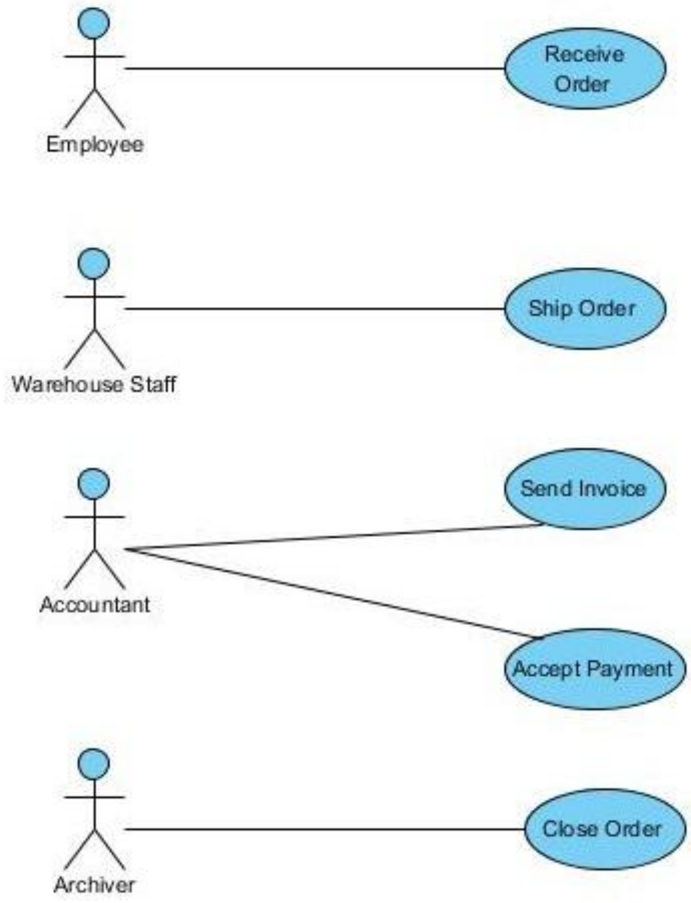


Figure 7: Use cases of the sample process

4.4.1.2 Site Inspection

The requirements and the problems should be inspected in their own places. Site inspection assists to build quality requirements. In business process software development, determining of all the roles is important. Site inspection may also help to determine missing roles.

4.4.1.3 Sequencing of Use Cases

If drawing of the initial process model seems cumbersome, sequencing of use cases may be applied to draw the model automatically. Sequencing of use cases is introduced here as an assistant tool to model a business process. First of all, the use cases which are related to business process steps are selected. Then the use cases are sorted if there are before and after relations. The resultant sorted diagram is called Sequencing Use Case Diagram (SUD). Building of SUD is explained in the following. The built SUD is reviewed and the analysis is ended with the BPMN BPD.

Every pair of an actor to a use case is called as actor-use-case-pair. In SUD, each actor-use-case-pair is taken and for each of the remaining actor-use-case-pairs the precedence relation is determined. In other words, which actor-use-case-pair comes after which actor-use-case-pair? If there is a precedence relation between the actor-use-case-pairs, it is indicated with an arrow which starts from the preceding actor-use-case-pair and ends with the succeeding actor-use-case-pair. If there is no relation between the actor-use-case-pairs, then arrow is not drawn. Topological sort of SUD gives a draft version of the process model.

If there is an actor-use-case-pair which will not be implemented, it is not shown in the SUD. If there are many actor-user-case-pairs, then the complexity of the SUD increases. Therefore, they may be grouped according to their use case diagrams and for each group a sub-SUD is built. Probably each sub-SUD would be a sub flow of the process model. In short, the complexity of SUD of many actor-use-case-pairs should be decreased by better analyzing the precedence relation.

SUD shows the main flow between the use cases. Therefore, a draft version of the process model can be built easily from SUD.

After building use cases, SUD is prepared. Figure 8 shows the SUD of the example process.

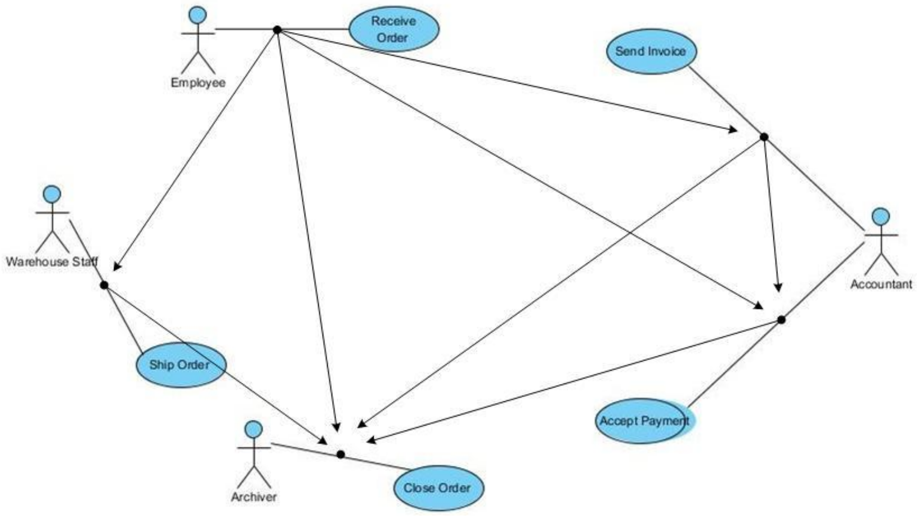


Figure 8: SUD of the sample process

4.4.1.4 Determining All the Roles

In the analysis stage, the most important work task for the business process automation is the determination of all the roles of the business process. All roles should be determined as much as possible. Otherwise missing roles can cause redesign of the process and increase the total development time. On the other hand, site inspection may help to determine missing roles.

4.4.1.5 Work Products

Use case diagrams and sequencing use case diagrams go to the design stage.

4.4.2. Design

4.4.2.1. Process Modeling

There are some options to model a process. For example,

- Petri Net may be used in case the states of the process should be expressed clearly.
- Moreover, the other formal methods like pi calculus, or
- situation calculus may be applied.
- Unified modeling can be used. Therefore, the UML activity diagrams may be employed.
- The Aris modeling method EPC can be applied.
- Flowcharts, or
- RAD models may be applied.

However, Business Process Diagram (BPD) of Business Process Modeling Notation (BPMN) is used to model the processes. Because, according to the analysis the model should be simple and should be understood easily by all the stakeholders of the process. BPD provides this. Also, the model should simply represent the parallel and optional branching of processes. This also is held by the BPD using the parallel and xor branch symbols.

BPD has the following advantages over the other modeling methods:

- It is designed for business process modeling.
- It is simple.
- It shows parallel executions simply.
- It shows serial executions simply.
- It does not contain formal steps so it is easy to understand. For example, in Petri nets there are states and in EPC diagrams there are events.
- All the stakeholders can understand it easily.

Figure 9 shows UML Activity Diagram (UML AD) of the sample process.

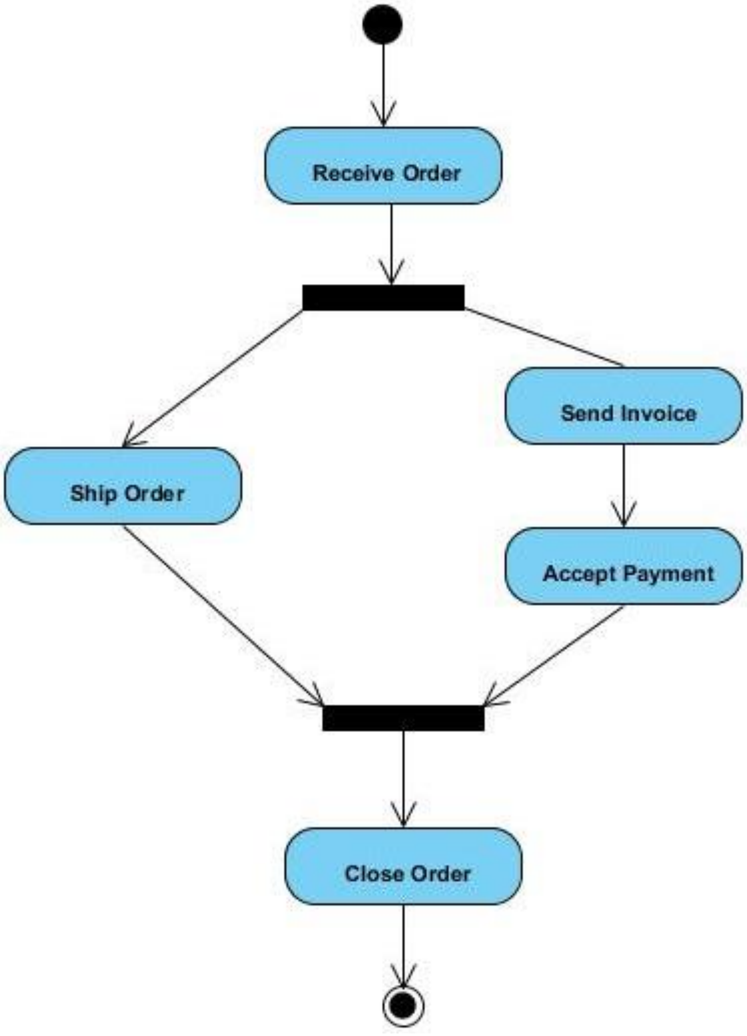


Figure 9: UML AD of the sample process

Figure 10 shows EPC diagram of the sample process.

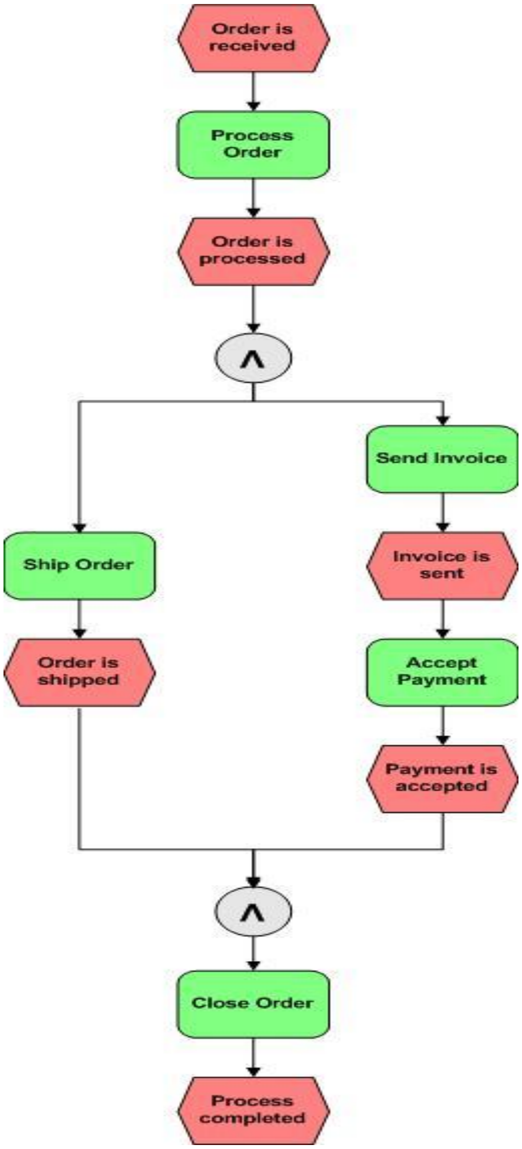


Figure 10: EPC diagram of the sample process

Figure 11 shows flowchart of the sample process.

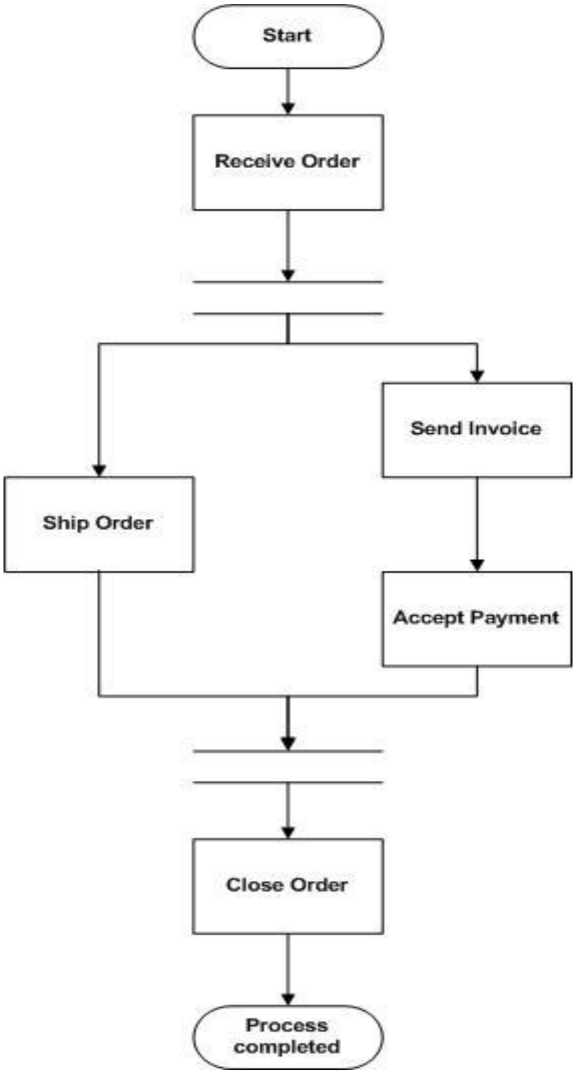


Figure 11: Flowchart of the sample process

Figure 12 shows the Petri net of the sample process.

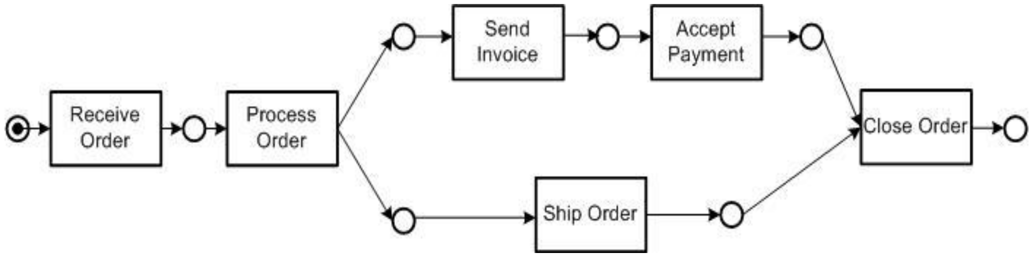


Figure 12: Petri net of the sample process

Figure 13 shows an example of Role Activity Diagram (RAD).

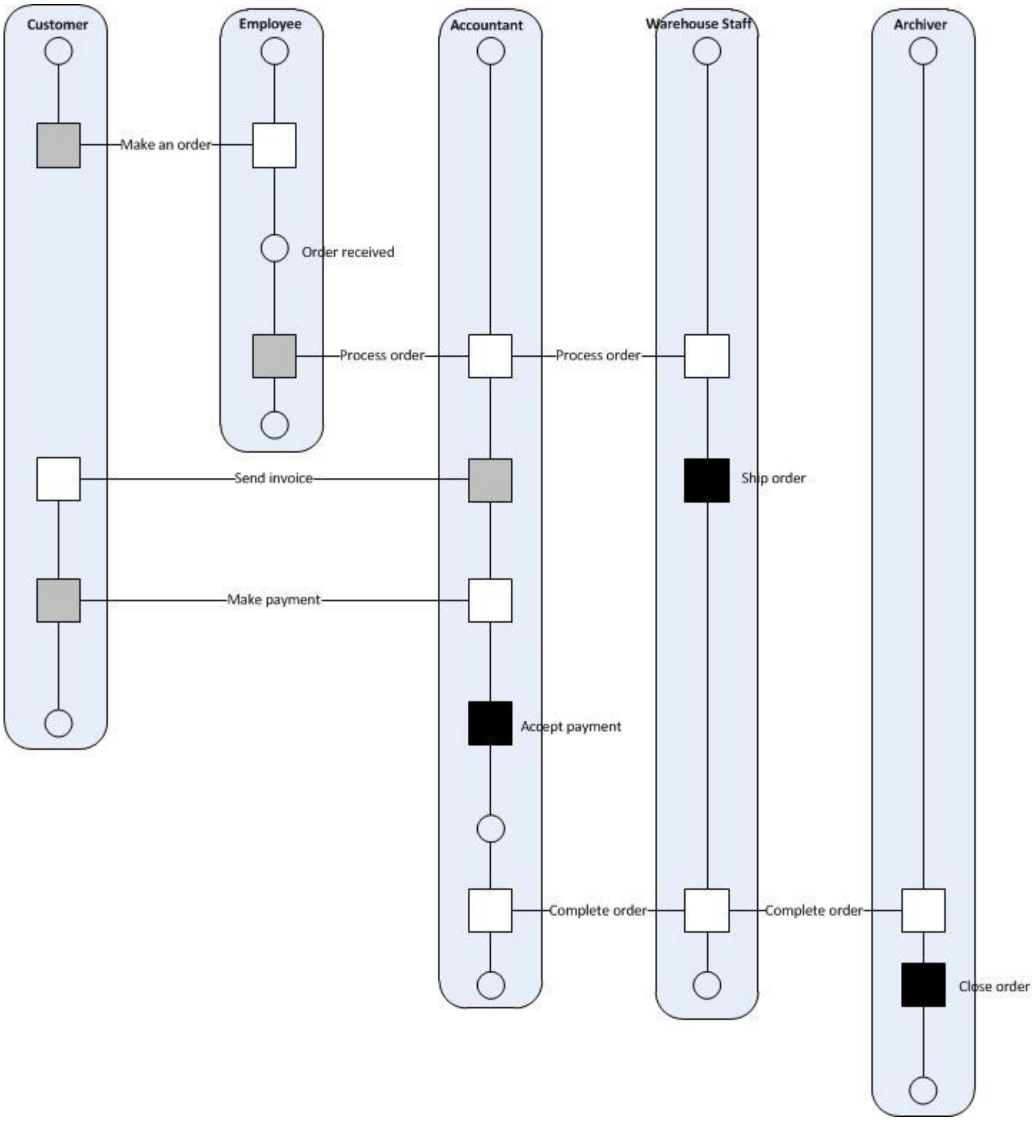


Figure 13: A RAD example

In Figure 13, circles represent events. Empty squares and grey squares are used for synchronization of the actions. Grey squares initiate synchronization and empty ones are initiated. The black squares are used to show simple actions.

4.4.2.2. From SUD to BPD

From SUDs, the initial BPD is obtained. Then, the BPD is reviewed and revised. In other words, some steps may be combined, dropped, or added to the initial BPD.

The initial BPD is firstly reviewed with the process owner department in the small-team meetings. Then BPD is revised with the other roles of the process model. After obtaining a mature process model, it is reviewed by all the stakeholders. Namely, it is reviewed in a whole-team meeting.

Figure 14 shows the BPD of the sample process obtained from the SUD.

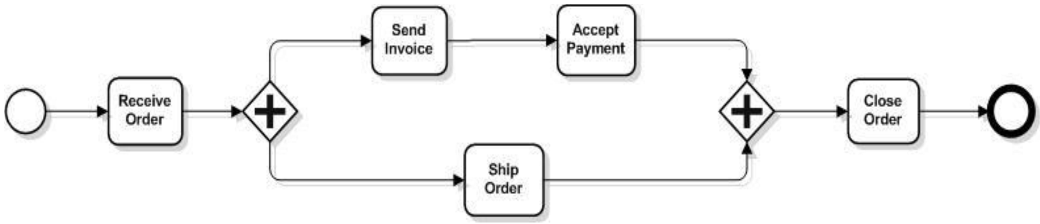


Figure 14: BPD of the sample process

4.4.2.3. Main Input Output Screen

For the fast implementation of the process model, it is assumed that a common screen is used in all the steps of the process model. In small-team meetings, basic input output requirements are determined with the process owner department. This screen is called as main input output screen. At that time a spike solution may be implemented through the main input output screen.

The main input output screen is detailed with the inclusion of the other roles of the process model. The spike solution can contribute to the determination of the input output requirements in detail.

4.4.2.4. Work Products

The work product of the design stage is the BPD.

4.4.3. Construction

4.4.3.1. Process Implementation

The BPD diagram is implemented in this stage. If there are suitable templates, they are used in the implementation.

4.4.3.2. Verification

In the verification part, the correctness of the implemented items in the iteration base lists is checked. This operation is done in small-team meetings. In other words, developers of the iteration base teams show the implemented parts to the team. Then, implemented parts are reviewed and tested for verification. This provides involvement of the customer with the development. Also, the related parts of the automation are executed in the meetings. Therefore, verification is realized by the team related with the specific iteration base. Each verified item of the iteration base lists are checked as verified.

4.4.3.3. Validation

The validation process is realized by the customers who take part in the iteration base teams. The items which have been validated during the validation process are checked also as validated.

4.4.3.4. Templates

4.4.3.4.1. Sub-process Templates

Sub-process templates are like general purpose functions. Common parts of processes are combined and developed as a template. Then the template is used for the related part of a process.

4.4.3.4.2. Process Templates

Process templates are used to develop a family of processes. These are general-purpose business processes designed for similar cases. For example, a template can be prepared in which a read-only form can be approved by a few actors.

4.4.3.5. Work Products

The main work product of the construction stage is the application itself. User guides are also prepared in this stage. In addition, configuration guides for the administrators of the business process are also prepared.

4.4.4. Going to pilot

Pilot application is crucial in business process automation. It simplifies “going to live” stage because:

- It provides a preparation for live process and increases the success of the implementation.
- Moreover, it provides real problems to development and allocates time to solve these problems in a relatively low stressed condition

This stage should be conducted if possible in this methodology.

4.4.5. Going to live

In this stage live process is deployed and started:

- User training is completed.
- Production systems are configured.
- System administration procedures are defined.
- Help desk personnel are informed.
- User support services are defined.
- Users are informed.
- Process is deployed.

4.4.6. Diagnosis

Training is important for the diagnosis phase. The more the customer is trained the more the business process is understood. Therefore, training shortens the duration of the diagnosis stage. Moreover, the general agile approach to the whole development process makes the diagnosis stage easier.

In diagnosis stage, the following work tasks are conducted:

- The process is monitored.
- The process is measured.
- The lifecycle restarts according to the diagnosis results.

4.5. Umbrella Activities

4.5.1. Light Project Management

This methodology encourages a light version of project management. Actually, the project is managed by a project leader. The main job of the project leader is to track the backlog list and activate the iteration bases with teams.

Project leader also deals with sponsorship. He tries to keep alive the sponsorship for the whole project.

Project leader also organizes the whole-team meetings.

4.5.2. Training

Training is important so that first parts of the meetings are used for training. The followings are the important reasons:

- Training improves the gathering of quality requirements.
- Training informs people about the process and decrease the development time.
- Training remembers the decisions taken before, and people do not need to solve again the solved problems.

4.5.3. Keeping History

History of the business process automation is important. This history is kept in the Iteration Base 0 document. This document is also called history document. Main activities of the automation, and the relations and the interactions between the iteration bases are recorded. Each team can add items to this document. Keeping history has the following advantages:

- It shows the state of the automation. Therefore all stakeholders can track the automation by following the history document and can adapt themselves to the new conditions.
- Different team members can use it as a means of communications. One team member provides information about their iteration base and members of other teams can access this information.
- The history document shows the relations and the interactions between all the iteration bases.

4.6. Spike Solutions

It has been seen that spike solutions speeds up development. Actually they train the stakeholders and provide to gather requirements better.

4.7. Templates

Any kind of template usage is encouraged because templates can accelerate the development process. In construction stage, sub-process templates and process templates can be used if suitable. In other stages of the life-cycle other templates may be employed. For example, if a template for gathering requirements was constructed, it would be used in analysis stages of different projects.

4.8. Why an agile method is required?

An agile method is used

- To adapt to the fast changing environment.
- To gather the requirements better. Because the requirements usually cannot be gathered easily. Always there are missing requirements. With agile methods,

requirements are gathered periodically. Therefore, the optimal requirements are gathered easier.

- To develop fast.
- To keep requirements at minimum. The requirements should be kept at minimum. Especially, all the requirements should be prioritized and unimportant ones should be handled later. If possible, they should be postponed after going to live. After going to live, some changes occur in the process and some requirements lose importance and some requirements come into prominence. And usually, some unimportant requirements need not to be implemented. Therefore, if possible, leave unimportant requirements after the going to live.

4.9. Why a specialized agile method is required?

- Requirements gathering for business processes are more difficult.
- The developers deal with more than one project at the same time and some routine work. Daily meetings are frequent for them and e.g. three weeks period for an increment is short.
- Usually business processes have many actors.

4.10. A Typical Business Process Software Development Scenario

- Analysis
    - The customer and the known actors are determined.
    - The analysis is done with the customer.
    - Small-team meetings are organized.
    - Use-cases are determined.
    - The other actors are determined with the customer.
    - The analysis with other actors is realized.
        - Again small-team meetings are organized.
        - Use-case diagrams are drawn.

- The analysis work is combined into the overall use-case diagrams with whole-team meetings.
    - If there are conflicting use-cases, they are marked as conflicting use-cases and these use-cases are combined into conflicting use-case diagrams.
- The overall use-case diagrams and conflicting use-case diagrams are reviewed with all the actors.
    - Overall use-case diagrams are refined.
    - The conflicting use-case diagrams are lessened.
    - Whole-team meetings are organized.
    - Use-cases are prioritized.
- Site inspections are realized.
    - Each type of actor is analyzed in his own place.
    - Missing use-cases are examined.
- With each customer, the updated overall use-case diagrams are talked in small-team meetings.
    - Missing use-cases are examined in these meetings.
    - If new actors are found in these small-team meetings, site inspection stage is realized.
    - Overall use-case diagrams are updated with the missing use-cases.
- The overall use-case diagrams are reviewed with whole-team meetings.
- Sequencing of use-cases is realized.
    - With small-team meetings, use-cases are sorted if suitable.
    - With a final whole-team meeting, the order of use-cases is clarified.
- Grouping of use-cases into the process tasks.
    - Use-cases are grouped into tasks.
    - The process model is drawn with BPMN BPD.
- Approval of the customers is taken.
    - The BPD and the overall use-case diagrams are approved by the customers.

- Design
    - Data Modeling Stage.
        - The data objects are determined according to the tasks and overall use-case diagrams.
        - The data model is prepared.
    - Task Modeling Stage.
        - Each activity is modeled.
    - Spike solution is provided.
        - Spike solution is showed to the customers.
- Construction
    - Data model is implemented.
    - The process is implemented.
    - The task screens are implemented.
    - The coding is completed.
- Going to pilot
    - Measure the process before going live.
    - The scope of the pilot application is determined.
    - The deployment of the process is realized.
    - Pilot application is started.
    - Fix errors in the process according to the pilot know-how.
- Going to live
    - Production system is configured.
    - Users are informed and support is planned.
    - The live process is deployed.
    - The live process is started.
- Diagnosis
    - The process is monitored.
    - Measure the process after going live.
    - The lifecycle restarts according to the diagnosis results.

4.11. Comparison of the Approaches

In this section, the proposed methodology is detailed by comparing it with similar or related approaches. The compared methodologies are as follows:

- XP
- Scrum
- Waterfall model
- IBM Rational Unified Process (RUP)
- The proposed methodology.

As agile methodologies, the widely-known XP and Scrum methodologies are taken. The methodologies are investigated and their important properties are noted. Totally 51 important properties are found. These important properties of the methodologies, which will be called as methodology aspects, are divided into 5 categories as follows:

- Structural
- Communicational
- Managerial
- Organizational
- Technical

These methodology aspect categories are explained in the following sections.

As a future work, these methodology aspects may be used to compare other methodologies.

4.11.1. Structural Methodology Aspects

Structural methodology aspects are shown in the rows of Table 25.

Table 25: Comparison of structural methodology aspects

Methodology Aspect	XP	Scrum	Waterfall	RUP	The Proposed Methodology
Phased development	✘	✘	✔	✔	✘
Iterative development	✔	✔	✘	✔	✔
Incremental development	✔	✔	✘	✔	✔
Pilot application	✘	✘	✘	✘	✔

In Table 25, the compared methodologies are shown. The tick mark in these tables is used to show the emphasis on the corresponding methodology aspect for the related methodology. If there is a cross mark, it means that the related methodology aspect is not applicable to the methodology or is neglected. In this context, agile methodologies refer to the four agile methodologies except the Waterfall model and RUP.

The property “phased development” is a must for the Waterfall methodology. However, the other methodologies are iterative and they do not have rigid phases. In the proposed method, there are stages but these show only the set of related tasks of the stages. Agile methodologies deliver fully developed software that meets a subset of the requirements at the end of the each iteration. In each iteration, teams are left alone and they work. Their goal is to get work, not to think about working. The aspect “iterative development” is important for agile methodologies, RUP and the proposed one. However, for the Waterfall model there is no iterative development.

In RUP, there are four phases, which are inception, elaboration, construction, and transition. Each phase is concluded with well-defined artifacts. Therefore, RUP is a phased development methodology while the proposed method is an iterative methodology which has interactions with stages.

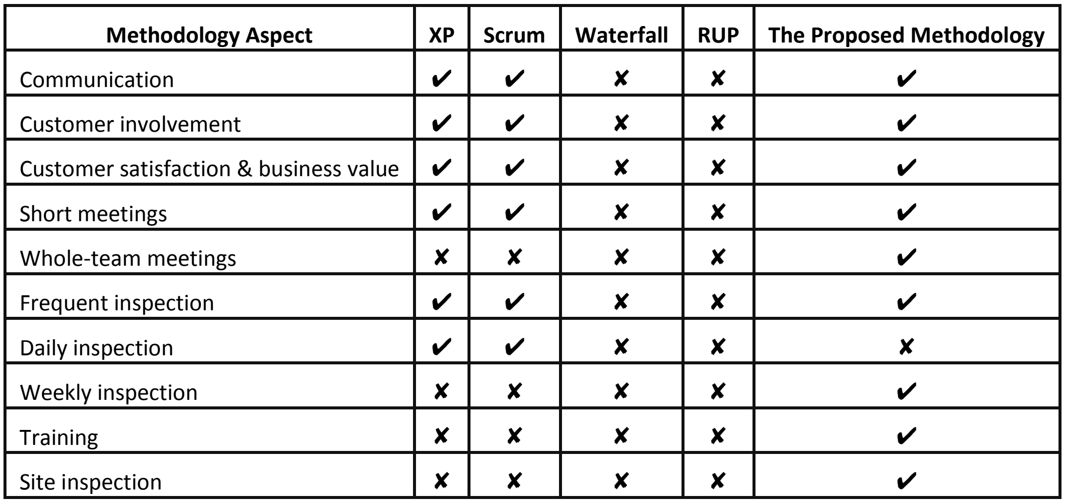
Incremental development is very important for the agile methodologies. At the end of each iteration, a set of requirements are developed and added to the delivered software. Until all the requirements of the customer are realized, this incremental development continues. Except the Waterfall model, at the end of each iteration an increment to the last product is achieved.

Pilot application is crucial in the proposed methodology because the complexity emerging from the number of the actors in business processes is taken under control before going live. The problems of the live application are seen earlier and precautions are taken before. However, in the other compared methodologies, pilot application is not emphasized.

4.11.2. Communicational Methodology Aspects

Communicational methodology aspects are shown in the rows of Table 26.

Table 26: Comparison of communicational methodology aspects



Methodology Aspect	XP	Scrum	Waterfall	RUP	The Proposed Methodology
Communication	✔	✔	✘	✘	✔
Customer involvement	✔	✔	✘	✘	✔
Customer satisfaction & business value	✔	✔	✘	✘	✔
Short meetings	✔	✔	✘	✘	✔
Whole-team meetings	✘	✘	✘	✘	✔
Frequent inspection	✔	✔	✘	✘	✔
Daily inspection	✔	✔	✘	✘	✘
Weekly inspection	✘	✘	✘	✘	✔
Training	✘	✘	✘	✘	✔
Site inspection	✘	✘	✘	✘	✔

The most effective and efficient method of conveying information is face-to-face conversation. Therefore, in agile methodologies, communication is highly emphasized. Especially, in XP, the open workspace concept is a means of communication. The whole team works in a large room to maximize the communication among each other. Also frequently held meetings in agile methodologies provide better communication between team members. In agile methodologies, customer is a natural member of the teams. Therefore, communication with the customer helps the understanding of the customer requirements and results in developing valuable software. The methodology aspect “communication” means tight face-to-face communication with the customer throughout the project. In agile methodologies and in the proposed methodology, communication is very important whereas in the Waterfall and RUP communication is limited or through written documents.

The property “customer involvement” means that customer should take role in the development of the software. In Waterfall methodology, this property does not exist, whereas in agile methodologies in order to deliver valuable software the customer should be a member of the development teams. Especially, in XP, customer should test the functions of the increments and accepts them. Customer involvement is not realized in RUP and the Waterfall model in which working with the customer is only in the beginning and/or at the end of the project.

The property “customer satisfaction & business value” aims to develop valuable product that satisfies the customer. It is important because of the valuable software. A methodology should aim to deliver software which is valuable for the customer. To realize this, the customer should be involved in the development of the project, the requirements of the customer should be understood well, and the delivered software should meet the real requirements of the customer. Therefore, each increment of the software is a step towards the realization of this property. In agile methodologies, this property is emphasized to get working software as early as possible. In XP, the concept metaphor describes the common vision how of the valuable software works. In RUP and the Waterfall model, this methodology aspect works only in the analysis phase and is forgotten in the rest of the project.

As a means of communication, meetings should be handled with the customer and the other stakeholders of the project. In agile methodologies, short meetings are emphasized [14]. However, the duration of the meetings is not a matter of RUP and the Waterfall model. The methodology aspect “whole-team meetings” means that nearly whole team of the project should make meetings infrequently for consolidation purposes. In the proposed methodology, this aspect is important whereas in the other compared methodologies there is no importance.

The property “frequent inspection” refers to monitoring the state of the project many times. In these meetings, each team member inspects each others’ activities and makes appropriate adaptations. The team takes a look at the requirements, considers the available technology, and evaluates its own skills and capabilities. Also, the team determines what needs to be done and selects the best way to do it. Frequency of these inspections is usually a day. Frequent inspection is emphasized very much in Scrum. Especially, in Scrum, the team starts the working day with a 15-minute meeting called Daily Scrum. In these meetings, three questions are answered:

- What have been done since the previous Daily Scrum?
- What to be done until to the next Daily Scrum.
- What are the problems in front of the team?

The daily scrums synchronize the team and help to adapt to the changing environment. In the proposed methodology, daily inspections take considerable amount of time because of handling more than one project at the same time. Therefore, a weekly period is found suitable for frequent inspections. However, in RUP and the Waterfall model are no frequent inspections like daily and weekly inspections.

The property “training” means that the customer is informed about the software being developed as much as possible. If there is no software to talk about, then similar software is explained. Training helps the customer to improve the vision. The aim of training is to gather requirements better because someone can describes something how much s/he knows it. Training is emphasized in the proposed methodology by including it in weekly meetings as an integral part. In the other methodologies, there is no counterpart.

Another important property for better requirements is “site inspection”. It means that the problem should be seen in its own place. Site inspection is important in the proposed methodology whereas there is no equivalent in the other compared methodologies.

4.11.3. Managerial Methodology Aspects

Managerial methodology aspects are shown in the rows of Table 27.

Table 27: Comparison of managerial methodology aspects

Methodology Aspect	XP	Scrum	Waterfall	RUP	The Proposed Methodology
Keeping history	✘	✘	✘	✘	✔
Product backlog	✘	✔	✘	✘	✔
Prioritizing of requirements	✘	✔	✘	✘	✔
Project management	✘	✘	✔	✔	✔
Sponsorship	✘	✘	✘	✘	✔
Sustainable pace	✔	✘	✘	✘	✔
Heavy weight methodology	✘	✘	✔	✔	✘
Light weight methodology	✔	✔	✘	✘	✔
Inclusive project planning	✘	✘	✔	✔	✘
Inclusive requirement management	✘	✘	✔	✔	✘
Change management	✘	✘	✘	✔	✘
Predictive methodology	✘	✘	✔	✔	✘
Verification of software quality	✘	✘	✘	✔	✘
Risk mitigation	✘	✘	✘	✔	✘

The property “keeping history” is to record important things during the development process. Keeping history helps to solve problems. In the proposed method, this property is emphasized. However, there is no emphasis on this methodology aspect in the compared methodologies.

The sum of the remaining tasks in a project is called “product backlog”. In Scrum, product backlog is important because the project is finished when it is completed enough. Also, it is used to estimate the remaining time to complete the project. Also in the proposed methodology, product backlog is important to prioritize the requests. However, in the other compared methodologies product backlog is not mentioned.

The property “prioritizing of requirements” provides doing important things first. In Scrum, requirements in the product backlog are prioritized and the requirements with high priority are implemented in the next iterations. In other words, the most valuable requirements are implemented first and the most valuable functionalities are produced first. This property eliminates some unnecessary requirements because they would be left to the end and at that time needlessness of them would appear. This methodology aspect is in the proposed methodology also whereas there is no prioritization in the other compared methodologies.

The methodology aspect “project management” is important both RUP and the Waterfall model whereas there is no solid project management in the other compared methodologies. Moreover, the big problems encountered during the development process can be solved easily by a sponsor. This methodology aspect is called “sponsorship”. In the proposed methodology, sponsorship is also a must. In the other compared methodologies, there is no obligation.

The methodology aspect “sustainable pace” means maximization of the productivity. The aim of this property is to get maximized productivity and to make it sustainable. In summary, during the project, the total productivity is maximized. Sustainable pace cannot be reached by working very hard. The team members should work as hard as their work is sustainable. In other words, they should rest enough to sustain their maximized productivity indefinitely. Moreover, well-rested teams produce quality work with the fewer defects in the software and they are more creative. Sustainable pace is emphasized only in XP and the proposed methodology among the compared methodologies.

The proposed methodology has some similarities with RUP [44]. Both methodologies are iterative and incremental. RUP is a heavy weight methodology [45] whereas the proposed method is a light weight methodology. The methodology aspect “heavy weight methodology” means that the project should have detailed documentation, inclusive planning, extroverted design, and sequential series of steps. The property “light weight methodology” stands at the other extreme end of this aspect. The Waterfall model is also a heavy weight methodology. However, the other compared methodologies are light weight methodologies.

The proposed methodology uses light project management but RUP applies comprehensive project management. The methodology aspect “inclusive project planning” means that the project should have a detailed project plan. RUP and the Waterfall model have inclusive project planning whereas the other compared methodologies do not have. In RUP, inclusive requirement management is done. The property “inclusive requirement management” means collecting all the requirements at the beginning of the project and managing all of them thereafter. The Waterfall model also focuses on inclusive requirement management. However, in the proposed methodology requirements are prioritized and only superior requirements are managed in related iteration bases. Moreover, other compared agile methodologies do not have inclusive requirement management. In the same way, RUP concentrates on change management. The methodology aspect “change management” means that every change request is handled and recorded carefully. However, in the proposed method, changes are not controlled strictly, only resembling aspect “keeping history” is conducted. In addition, the other compared methodologies do not concentrate on change management.

In RUP, as a heavy weight aspect, inclusive project planning is made. Therefore, everything is planned and it is considered as a predictive methodology. The methodology aspect “predictive methodology” means that the duration of the project is completely planned. The Waterfall model is also a predictive methodology. However, the other compared methodologies are adaptive.

The methodology aspect “verification of software quality” means that software quality should be verified. This aspect is very important in RUP. In the other compared methodologies, there is no counterpart. This methodology aspect “risk mitigation” means minimizing risks in the project. In RUP, risk assessment is emphasized very much. Therefore, RUP tries to mitigate risks. Risk mitigation is not a concern for the other compared methodologies.

4.11.4. Organizational Methodology Aspects

Organizational methodology aspects are shown in the rows of Table 28.

Table 28: Comparison of organizational methodology aspects

Methodology Aspect	XP	Scrum	Waterfall	RUP	The Proposed Methodology
Teamwork	✔	✔	✔	✔	✔
Small teams	✔	✔	✘	✘	✔
Self-organization	✔	✔	✘	✘	✔
Self-management	✔	✔	✘	✘	✔
Trust and self-motivation	✔	✔	✘	✘	✔
Cross-functional	✔	✔	✘	✘	✔
Retrospective	✔	✔	✘	✘	✔
ScrumMaster	✘	✔	✘	✘	✘

Teamwork is important for all kinds of software projects. Especially in agile methodologies, teams are also responsible for the success of the project. Therefore, teams are empowered by including customer and by granting management responsibilities. The success of the teamwork depends on communication firstly. Particularly, in XP, working in pairs shows the importance of this property. And in agile methodologies, communication channels among the team members are reinforced to get better teamwork. All the compared methodologies give importance to this methodology aspect.

The property “small teams” are encouraged in agile methodologies. Small teams have better communication among team members, and whenever they take responsibilities, they get success. In RUP and the Waterfall model, the size of the teams does not matter.

In agile methodologies, responsibilities are spreaded over among team members. And, the success of the project depends on the individuals in the project. Therefore, teams are self-organizing and self-managing. The methodology aspect “self-organization” contributes to produce the best architectures and designs by taking the best valuable requirements. The property “self-management” means that teams should manage their own work because the best people to make decisions about the project are those closest to the details. In order to empower self-management, team members should be equipped with an understanding of the overall goals of the project. Especially, in Scrum, the team manages the sprint by starting planning of it. In the proposed methodology, project management is light-weighted and includes the coordination of the efforts of the different teams. Iterations in the methodology are managed by the team members. However, in the Waterfall model and RUP management and organization are realized by project managers. The methodology aspect “project management” is important both RUP and the Waterfall model whereas there is no solid project management in the other compared methodologies.

The property “trust and self-motivation” is the core of the teamwork in agile methodologies. Self-motivation means that a person does his or her own work in motivated way independent of external agent and condition. Motivated individuals with enough responsibilities and managing capabilities results in the success of the project. Therefore, environment and support needed should be supplied to them and trust should be given to them in order to get the job done. This methodology aspect is encouraged in agile methodologies whereas there is no counterpart in the Waterfall model and RUP.

The property “cross-functional” means that individuals with different functional expertise work toward the common goals of the project. In agile methodologies, cross-functional teams are important because solving problems becomes easier by combining the powers of different perspectives. However, in RUP and the Waterfall model individuals with the same functional expertise are placed into the same teams.

The methodology aspect “retrospective” can be considered as all kinds of feedback. Especially, in Scrum, after the review of the previous iteration prior to the next iteration a retrospective meeting is realized to adjust the applied method. This method adaptation is important in XP and the proposed methodology also. However, it is neglected in the other compared methodologies.

In Scrum methodology, there is a special role as ScrumMaster. ScrumMaster is a consultant experienced on scrum methodology. According to the feedbacks taken from retrospective meetings, ScrumMaster guides the team members to adjust the behavior of the applied method. This methodology aspect is only seen in Scrum. In the other compared methodologies, there is no ScrumMaster.

4.11.5. Technical Methodology Aspects

Technical methodology aspects are shown in the rows of Table 29.

Table 29: Comparison of technical methodology aspects

Methodology Aspect	XP	Scrum	Waterfall	RUP	The Proposed Methodology
Frequent releases	✔	✔	✘	✘	✔
Adaptation	✔	✔	✘	✘	✔
Template usage	✘	✘	✘	✘	✔
Spike solutions	✔	✔	✘	✘	✔
Main input output screen design	✘	✘	✘	✘	✔
Use case diagrams	✘	✘	✘	✔	✔
Role modeling	✘	✘	✘	✘	✔
Simplicity	✔	✘	✘	✘	✘
Technical excellence	✔	✘	✘	✘	✘
Pair programming	✔	✘	✘	✘	✘
Collective code ownership	✔	✘	✘	✘	✘
Test-driven development	✔	✘	✘	✘	✘
Continuous integration	✔	✘	✘	✘	✘
Visually modeling software	✘	✘	✘	✔	✘
Component-based development	✘	✘	✘	✔	✘

Delivering working software frequently is called “frequent releases”. Each release meets the most valuable subset of the product backlog. This property makes software visible and keeps everything open and tangible. Therefore, all agile methodologies apply this property. However, there are no frequent releases in the Waterfall model and RUP.

In agile methodologies, adapting to the fast changing environment is a must. The methodology aspect “adaptation” welcomes changing requirements, even late in development. To reach to the valuable software, the needed changes in the requirements are done and the remaining most valuable requirements are implemented for the customer’s competitive advantage. In Waterfall methodology, change is forbidden. Similarly, change is difficult in RUP.

The property “template usage” means that any kind of template should be used as much as possible in the development. In order to develop fast, template usage is encouraged in the proposed methodology. However, importance is not given in the other compared methodologies. The property “spike solutions” means a technical investigation which is a small experiment to research the answer to a problem. Spike solutions, which are beneficial to gather requirements better, are supported in the agile methodologies whereas in RUP and the Waterfall model are no spike solutions.

In order to obtain the general requirements of user interactions, designing main input output screen is also utilized. The property “main input output screen design” is special to the proposed methodology. In other methodologies, there is no corresponding. The property “use case diagrams” refers to the employment of use cases in the development. Use case diagrams are used in the proposed methodology and RUP whereas in the other compared methodologies they are not employed. Use cases are crucial because of determining the roles in the process. The property “role modeling” means determining the roles in a business process. In the proposed methodology, role modeling is important because of finding missing actors in the business process. However, in the other compared methodologies it is not so important.

In XP, simplicity is emphasized. The methodology aspect “simplicity” means that simple design, programming practices, and simple methods give the team courage. Especially, simple design is crucial because of not wasting extra time for complexity. In the other compared methodologies, it is not a concern.

Technical excellence enhances agility. This property “technical excellence” means maximizing technical value. Therefore, in XP, continuous attention is given to technical excellence and good design. Especially in XP, working in pairs helps to improve technical excellence. Refactoring process removes duplications in coding, increases the cohesion, and lowers the coupling in the code. Also, applying coding standards makes the code as if only one programmer wrote it. However, in the other compared methodologies, technical excellence is omitted.

In XP, programmers work together in pairs and as a group, with simple design, obsessively tested code, and continually improving the design. This methodology aspect is called “pair programming” and there is no equivalent in the other compared methodologies. Any code in the project has no single ownership; all the codes are collectively owned by the programmers. This methodology aspect is called “collective code ownership”, which causes the codes to take attentions of all the programmers, increases code quality, and reduces defects in the codes. Again this aspect is only emphasized in XP among the compared methodologies.

Test-driven development is important in XP because good feedback requires good testing. The methodology aspect “test-driven development” means that testing code is written before writing the actual code. Test-driven development is special to XP among the compared methodologies. Moreover, XP teams keep the system fully integrated and running all the time. This methodology aspect is called “continuous-integration”. In continuous-integration, at least one build is taken daily. There is no corresponding in the other compared methodologies.

Another methodology aspect peculiar to RUP is “visually modeling software”. It means that software is modeled visually. This aspect is neglected in the other compared methodologies. The methodology aspect “component-based development” is also special to RUP among the compared methodologies. It means assembling applications from components, which are reusable, executable packages of software that provide their services through well-defined interfaces. RUP encourages component-based development whereas other compared methodologies omit this aspect.

4.12. The Impact of Methodology Aspects to Effort

In this part, the effects of the methodology aspects to the effort are investigated. For this purpose, a simple model is introduced. The effort spent for a project can be calculated simply with two parameters. One is the total number of the tasks in the project and the other is the average effort per task. Multiplication of them gives the total effort spent for the project.

In the proposed methodology, training, site inspection, role modeling, use case diagrams, main input output screen design, keeping history, template usage, pilot application, and weekly inspection are emphasized. Site inspection, role modeling, use case diagrams, and main input output screen design help to minimize the total number of the tasks in the project. On the other hand, template usage, pilot application and keeping history help to minimize the average effort. Training and weekly inspection contribute to minimize both the number of tasks and the average effort.

As a future work, effort models may be built from various methodology aspects. Their contributions to effort may be examined empirically. Moreover, their effects to the average effort per task and the total number of tasks may be experimented quantitatively.

4.12.1. The Impact of Structural Methodology Aspects

The impact of structural methodology aspects is shown in Table 30.

Table 30: The impact of structural methodology aspects to effort

Methodology Aspect	Minimize the total number of tasks	Minimize the average effort
Phased development	✘	✘
Iterative development	✔	✔
Incremental development	✔	✔
Pilot application	✘	✔

In Table 30, the impact of the structural methodology aspects to the effort is given. The total effort spent for a project can be minimized by minimizing the total number of the tasks and the average effort per task. In Table 30, the first column shows the methodology aspect, the second column shows its impact to minimize the total number of tasks, and the third column shows the impact to minimize the average effort.

The mark "tick" means that the corresponding methodology aspect has significant impact to either the total number of tasks or the average effort per task. On the other hand, the mark "cross" means that the corresponding methodology aspect has no impact or insignificant impact on either the total number of tasks or the average effort per task.

The methodology aspect "phased development" has no impact on the total number of tasks. In the phased development, all the requirements are determined at the beginning of the project and they are implemented later. However, the methodology aspect "iterative development" decreases the total number of tasks. Each iteration reviews the remained requirements and eliminates unimportant ones. Therefore, the total number of requirements would decrease. In the iterative development, know-how is obtained in early iterations, and this know-how is used later for the similar tasks. Hence, this decrease the average effort spent for a task. However, the phased development does not have this kind of know-how to decrease the average effort per task.

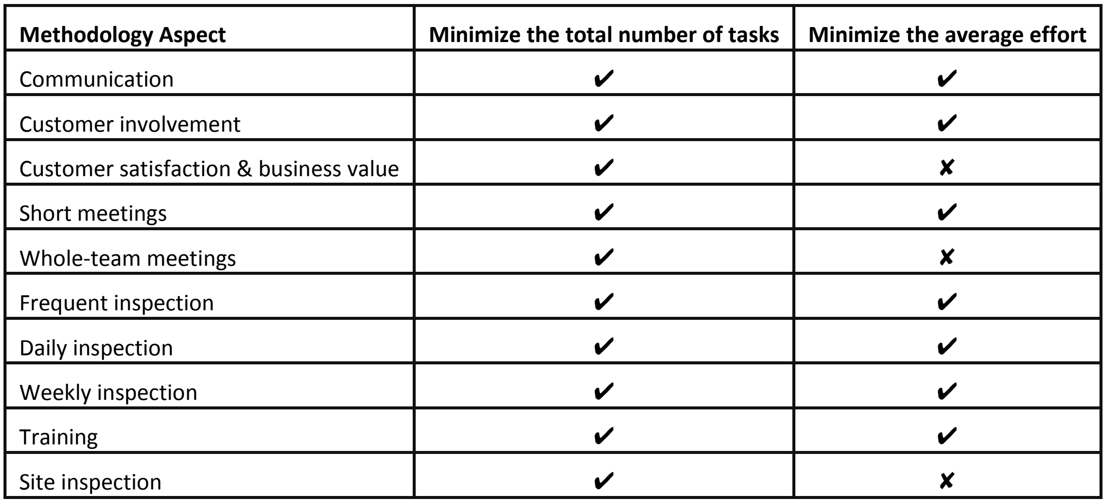
The methodology aspect "incremental development" has similar impact on effort with the aspect "iterative development". Each increment will add some value to the end product. Therefore, again valueless requirements will be eliminated causing a decrease in the total number of tasks. In addition, know-how obtained in early increments would be used later resulting in a decrease in the average effort per task.

The methodology aspect "pilot application" helps problems to be solved in shorter times. Hence, they contribute to minimize the average effort.

4.12.2. The Impact of Communicational Methodology Aspects

The impact of communicational methodology aspects is shown in Table 31.

Table 31: The impact of communicational methodology aspects to effort



Methodology Aspect	Minimize the total number of tasks	Minimize the average effort
Communication	✔	✔
Customer involvement	✔	✔
Customer satisfaction & business value	✔	✘
Short meetings	✔	✔
Whole-team meetings	✔	✘
Frequent inspection	✔	✔
Daily inspection	✔	✔
Weekly inspection	✔	✔
Training	✔	✔
Site inspection	✔	✘

The methodology aspect “communication” promotes valuable requirements and eliminates unimportant ones. It results in decreasing the total number of tasks. Communication helps to solve problems in a shorter time. Therefore, it also decreases the average effort spent for a task. In a similar way, customer involvement supports eliminating unimportant tasks resulting in a decrease in the total number of tasks. Also it shortens problem solving time. Hence, it decreases the average effort.

The methodology aspect “customer satisfaction and business value” causes prioritizing valuable requirements. Therefore, it eliminates unimportant requirements and helps to diminish the total number of tasks. However, this aspect does not shorten the average effort.

The methodology aspect “short meetings” increases the communication between stakeholders. Like the aspect “communication”, it decreases the total number of tasks and the average effort per task.

The methodology aspect “whole-team meetings” helps to consolidate the requirements and tasks. Therefore, it helps to decrease the total number of tasks whereas it does not a significant impact on the average effort spent for a task.

The methodology aspect “frequent inspection” reviews the state of the project. Therefore, it eliminates unimportant tasks and helps to reduce problem solving time. In brief, it contributes to decrease the total number of tasks and the average effort. In kind, the methodology aspects “daily inspection” and “weekly inspection” decreases both of them.

The methodology aspect “training” empowers with knowledge causing to eliminate unnecessary requirements and tasks. It also aids to be solved problems in shorter times. Hence, it decreases the total number and the average effort.

The methodology aspect “site inspection” helps to eliminate unnecessary tasks. Therefore, they decrease the total number of tasks. However, they do not have impact on the average effort per task.

4.12.3. The Impact of Managerial Methodology Aspects

The impact of managerial methodology aspects is shown in Table 32.

Table 32: The impact of managerial methodology aspects to effort

Methodology Aspect	Minimize the total number of tasks	Minimize the average effort
Keeping history	✘	✔
Product backlog	✔	✘
Prioritizing of requirements	✔	✘
Project management	✘	✘
Sponsorship	✘	✔
Sustainable pace	✘	✔
Heavy weight methodology	✘	✘
Light weight methodology	✘	✔
Inclusive project planning	✘	✘
Inclusive requirement management	✘	✘
Change management	✘	✘
Predictive methodology	✘	✘
Verification of software quality	✘	✔
Risk mitigation	✘	✘

The introduced aspect “keeping history” does not affect the total number of tasks. However, it reinforces problem solving ability. Consequently, it diminishes the average effort per task.

The methodology aspect “product backlog” contributes to eliminate unnecessary tasks. Hence, it helps to decrease the total number of tasks whereas it has no effect on the average effort. The methodology aspect “prioritizing of requirements” behaves like the previous aspect. Therefore, it decreases the total number of tasks without affecting the average effort per task.

The methodology aspect “sponsorship” helps problems to be solved in shorter times. Hence, they contribute to minimize the average effort. There is another aspect which contributes to neither the total number of tasks nor the average effort. It is the “project management” aspect.

The methodology aspect “sustainable pace” increases the development speed resulting in reducing the average effort. However, they do not affect the total number of tasks.

The methodology aspects “heavy weight methodology” and “light weight methodology” does not change the total number of tasks. However, the light weight methodology supports to increase the development speed. Therefore, light weight methodology reduces the average effort whereas the heavy weight methodology has no influence on it.

The methodology aspects “inclusive requirement management”, “change management”, “risk mitigation”, “inclusive project planning”, and “predictive methodology” do not affect both the total number of tasks and the average effort per task.

The methodology aspect “verification of software quality” assists to reduce problem solving time. Therefore, these aspects decreases the average effort spent for a task whereas they do not influence the total number of tasks.

4.12.4. The Impact of Organizational Methodology Aspects

The impact of organizational methodology aspects is shown in Table 33.

Table 33: The impact of organizational methodology aspects to effort

Methodology Aspect	Minimize the total number of tasks	Minimize the average effort
Teamwork	✘	✔
Small teams	✘	✔
Self-organization	✘	✔
Self-management	✘	✔
Trust and self-motivation	✘	✔
Cross-functional	✘	✔
Retrospective	✘	✔
ScrumMaster	✘	✔

The methodology aspect “teamwork” helps to solve problems in shorter times. Therefore it decreases the average effort per task whereas the total number of tasks does not change. The methodology aspect “small teams” affects like the previous one. Consequently, it decreases the average effort but not the total number of tasks.

The methodology aspects “self-organization”, “self-management”, trust and self-motivation”, and “cross-functional” reduces problem solving time. They do not affect the total number of tasks. Therefore, they decrease the average effort per task.

The methodology aspect “retrospective” adapts the applied method. Therefore, it aids to lessen the average effort by employing the know-how. This aspect does not affect the total number of tasks.

The methodology aspect “ScrumMaster” increases the development speed resulting in reducing the average effort. However, they do not affect the total number of tasks.

4.12.5. The Impact of Technical Methodology Aspects

The impact of technical methodology aspects is shown in Table 34.

Table 34: The impact of technical methodology aspects to effort

Methodology Aspect	Minimize the total number of tasks	Minimize the average effort
Frequent releases	✔	✘
Adaptation	✔	✔
Template usage	✘	✔
Spike solutions	✔	✘
Main input output screen design	✔	✘
Use case diagrams	✔	✘
Role modeling	✔	✘
Simplicity	✘	✔
Technical excellence	✘	✔
Pair programming	✘	✔
Collective code ownership	✘	✔
Test-driven development	✘	✔
Continuous integration	✘	✔
Visually modeling software	✘	✔
Component-based development	✘	✔

The methodology aspect “frequent releases” causes to eliminate the unnecessary and valueless requirements. Therefore, it decreases the total number of tasks. However, it does not influence the average effort. Another aspect is “adaptation”. The aspect helps to eliminate the unnecessary tasks and to lessen problem solving time. Consequently, adaptation has impact on the total number of tasks and the average towards to reduce them.

The methodology aspect “template usage” assists to reduce the average time by employing the prepared templates. This aspect does not change the total number of tasks.

The methodology aspects “spike solutions”, “main input output screen design”, “role modeling”, and “use case diagrams” help to eliminate unnecessary tasks. Therefore, they decrease the total number of tasks. However, they do not have impact on the average effort per task.

The methodology aspects “simplicity”, “technical excellence”, “collective code ownership”, “pair programming”, “test-driven development”, and “continuous integration” increase the development speed resulting in reducing the average effort. However, they do not affect the total number of tasks.

The methodology aspects “component-based development” and “visually modeling software” help to increase the development speed. Therefore, these aspects decreases the average effort spent for a task whereas they do not influence the total number of tasks.

CHAPTER 5

THE ESTIMATION FORMULA AND EFFORT COMPARISON

In this chapter, the proposed methodology is applied to the new two business process software development projects. The study is carried out as a case study [62]. The phases of a case study are applied one by one. This case study compares the classical method and the proposed methodology on the effort spent during the development. It is decided that an estimation formula would compare the efforts. After the objective is determined, the data sources are specified. They are developers and analysts of the projects, who are asked on monthly basis. The data collection phase is continued with the collecting evidence phase. In that phase, the collected values are recorded in the related tables monthly. Then the data is analyzed and the results are reported. The model of the estimation formula and the results are discussed against validity threats [63].

Agile methods [30] [31] decrease the development time where requirements are not specified fully and correctly. This proposed methodology is based on agile approaches so that expecting effort to be saved by applying the proposed methodology is reasonable. Therefore, the construct validity has been provided in this study [63].

In the following sections, why an estimation formula is needed is investigated firstly. The objective of the case study is based on the estimation formula. Then, the business processes implemented in classical method are explained. The estimation formula is built in the following section. Then, estimation accuracy section measures the accuracy of the formula. Lastly, the new business processes which are developed in the proposed methodology are talked and the results are given.

5.1. The Requirement for an Estimation Formula

An agile methodology is proposed for business process software development. It is argued that this methodology will decrease the development efforts compared with the traditional method. To measure the decrease in manpower a formula is needed to estimate the development efforts. This formula will give a manpower value as if a business process was implemented using the Waterfall methodology.

The formula should be simple in order to estimate the development efforts more accurately, and in order to avoid biasing. Therefore, only two parameters are used to represent business process automation effort. Actually these two parameters are used to express the size of a business process. From this point of view, this formula can be considered as a one parameter formula. The simple value showing the size of a business process is the number of steps in the business process. Therefore, the first parameter is the number of tasks in a business process. The second parameter is used to represent complex business processes. Complex tasks in a business process increase the efforts. Usually, tasks which require data input from users other than simply selecting an action are complex. These tasks will be called as interactive tasks throughout this work. Therefore, for interactive tasks the second parameter is used. In summary, an estimation formula is built with only one parameter. This parameter is simply the addition of the number of steps and the number of interactive steps in a process. Namely, this formula takes the size of the business process as a parameter and gives the effort spent for the business process.

Before applying the agile methodology, it was used to apply classical methods to develop business processes. It was a phased development. In other words, first of all analysis is done. Then, design comes. Lastly, programming, test, and deployment come. Therefore, this methodology is considered as the well-known Waterfall methodology. With the Waterfall methodology, nine business processes have been developed:

- The first one is for the approval of purchase requisition.
- The second one is for insurance claim.
- The third one is for material request from warehouse.
- The forth one is for the approval of purchase order.
- The fifth one is for local duties.
- The sixth one is for quality notifications.
- The seventh one is for quality notification tasks.
- The eighth one is form shipment.
- The last one is for plane tickets.

The manpower values spent for the development have been kept. And the models of the business processes have been kept. From these data, it was tried to derive a formula to estimate the development effort. Using this formula, the development effort of a business process can be estimated as if the business process was implemented using the Waterfall methodology.

New business processes will be implemented with the proposed agile methodology and recorded the development effort. On the other hand, the above formula will help us to estimate the development effort of the new business processes as if they were developed using the Waterfall methodology. As a result, two values of development effort for each new business process are obtained. It is argued that the proposed agile methodology will decrease the development efforts.

5.2. Implemented Business Processes

5.2.1. Purchase Requisition Business Process

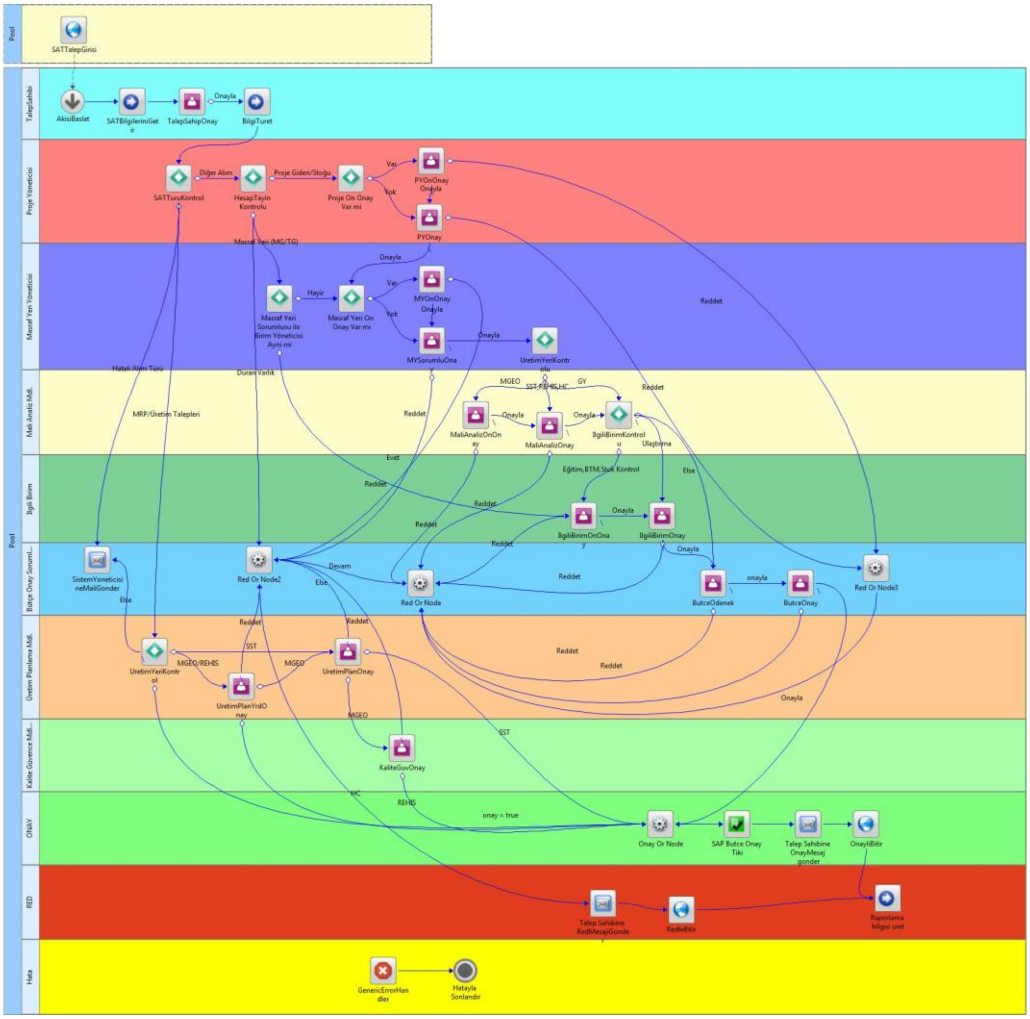


Figure 15: Purchase Requisition Business Process

This process, which is shown in Figure 15, has 14 task steps. In other words, it has 14 steps with human interface. All of them are for approval. Namely, these steps wait simply positive or negative input from users. Therefore, these steps are not categorized as interactive.

5.2.2. Insurance Claim Business Process

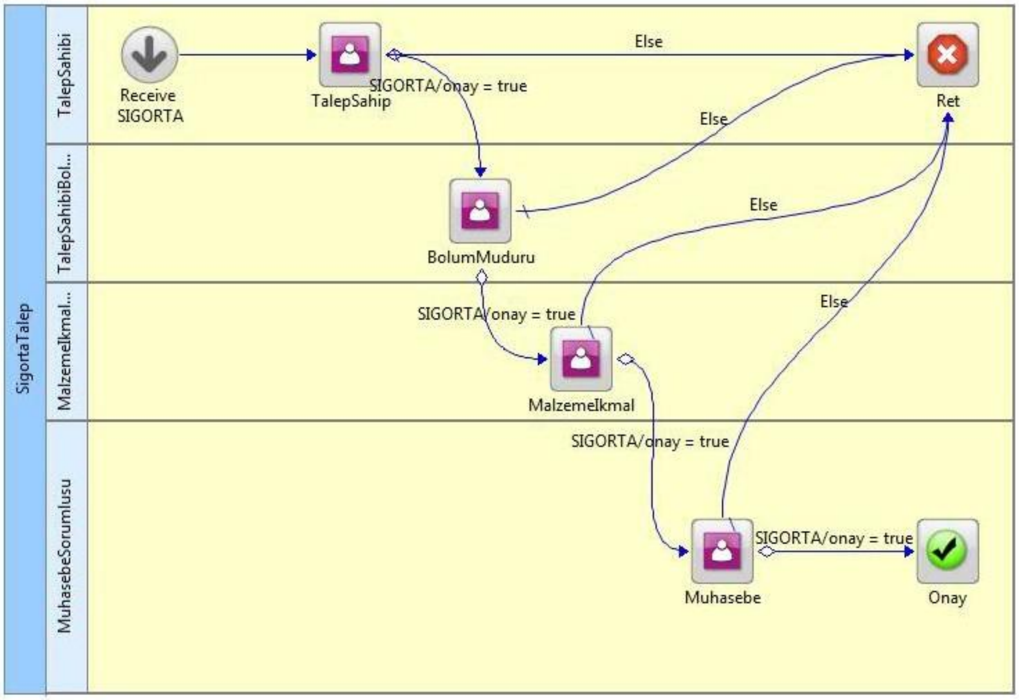


Figure 16: Insurance Claim Business Process

This business process, which is shown in Figure 16, has only 4 simple task steps.

5.2.3. Material Request Business Process

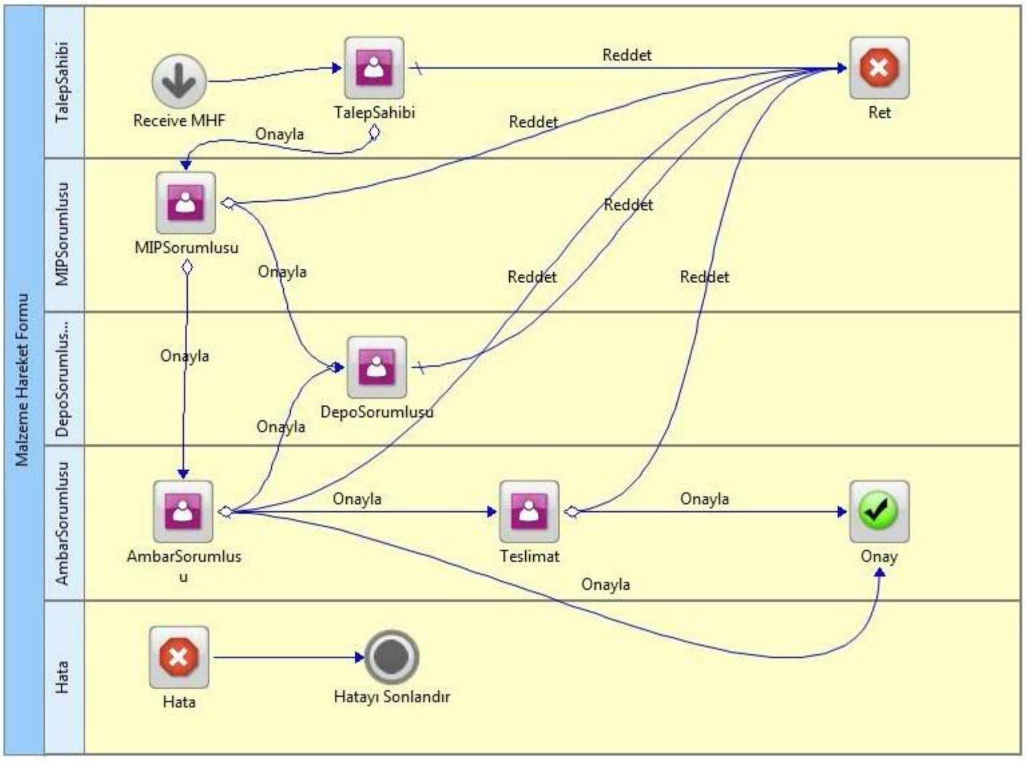


Figure 17: Material Request Business Process

This business process, which is shown in Figure 17, has 5 task steps. 4 of them are interactive. In other words, those steps wait many inputs from users. The last step is for approval and it is considered as a simple step.

5.2.4. Purchase Order Business Process

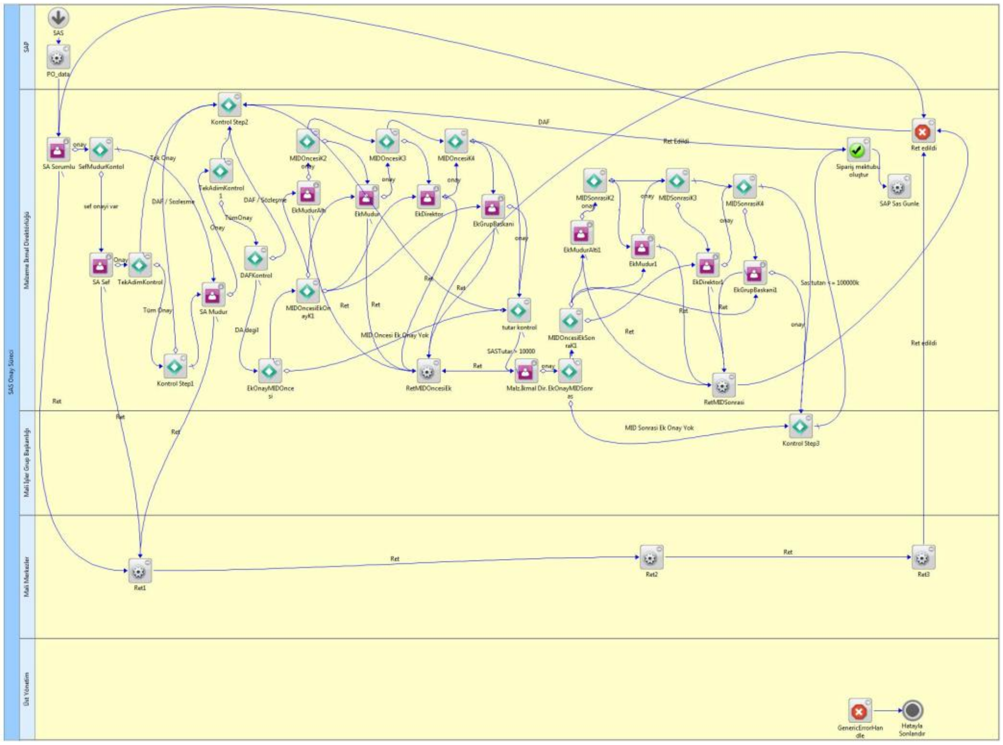


Figure 18: Purchase Order Business Process

This business process, which is shown in Figure 18, has 12 task steps. These steps are for approval.

5.2.5. Quality Notification Business Process

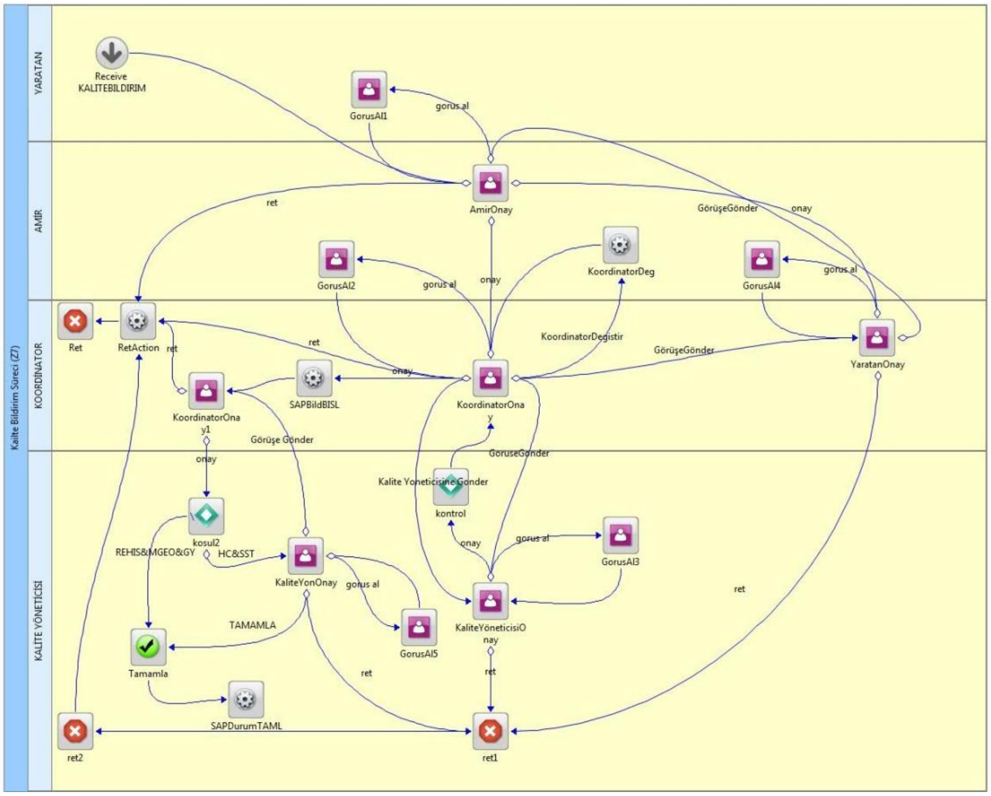


Figure 19: Quality Notification Business Process

This business process shown in Figure 19 has 11 steps. 5 of them are complex.

5.2.6. Quality Tasks Business Process

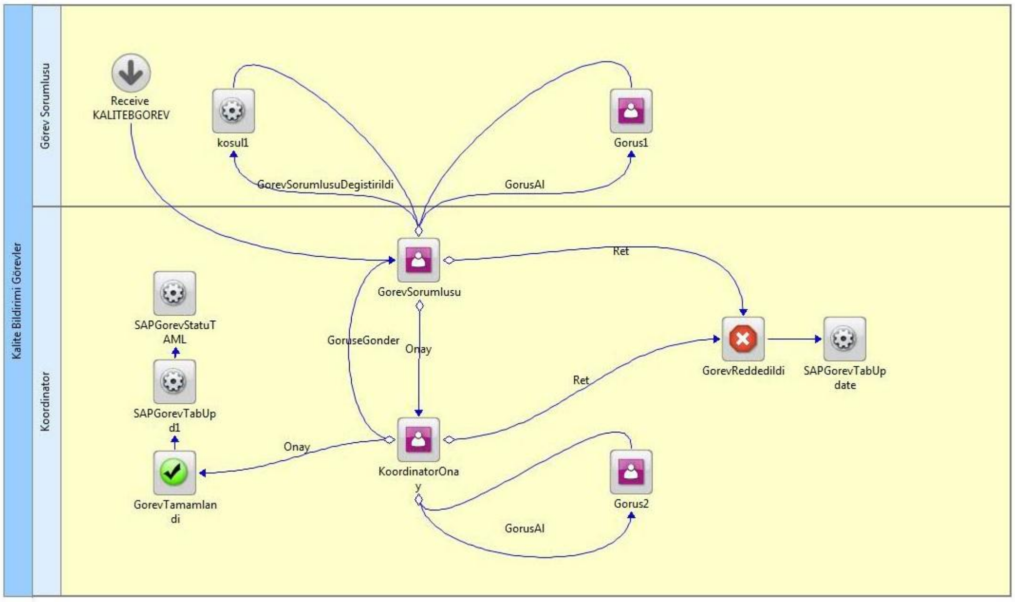


Figure 20: Quality Tasks Business Process

This process shown in Figure 20 and it has 4 steps of which 2 are complex.

5.2.7. Duty Order Business Process

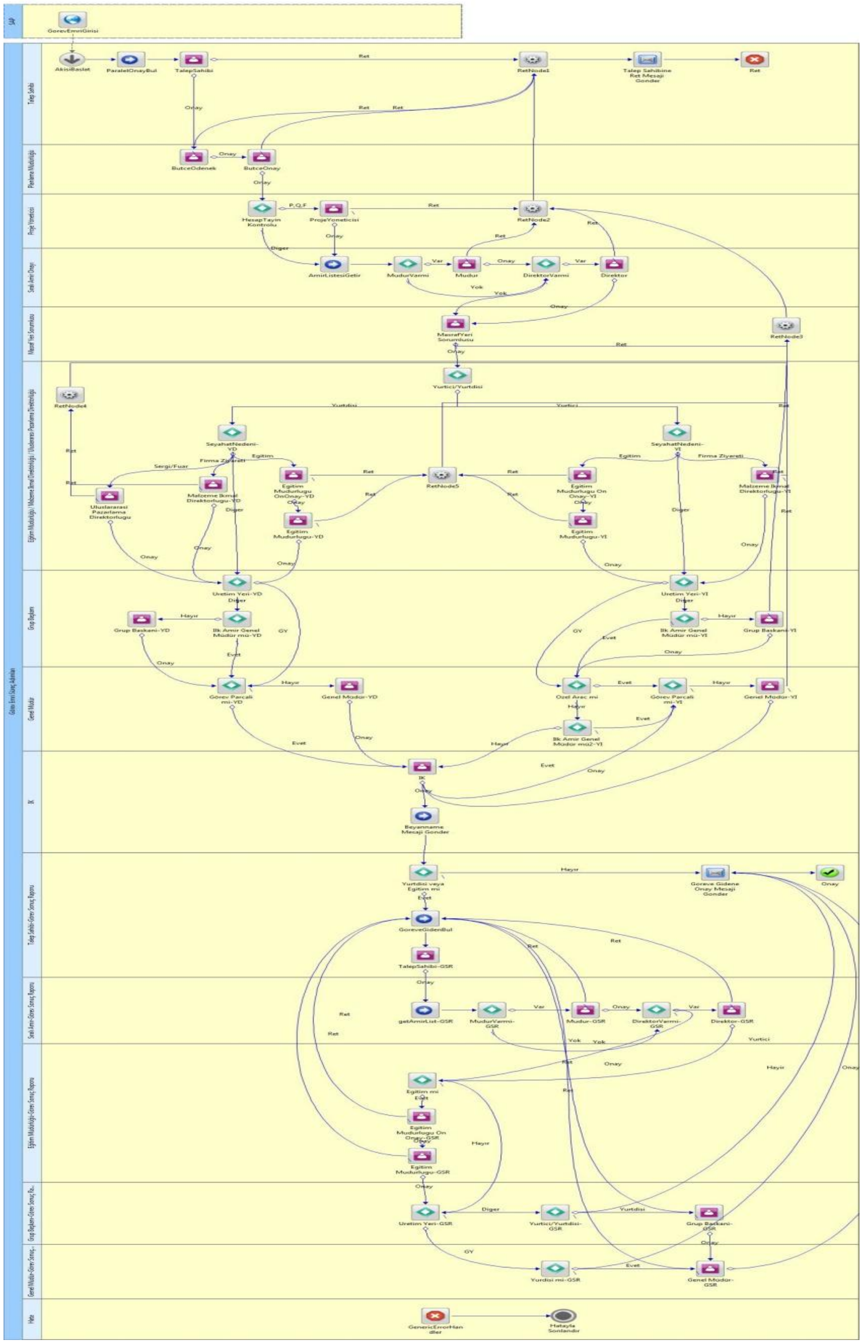


Figure 21: Duty Order Business Process

This process shown in Figure 21 and it has 26 steps.

5.2.8. Shipment Business Process

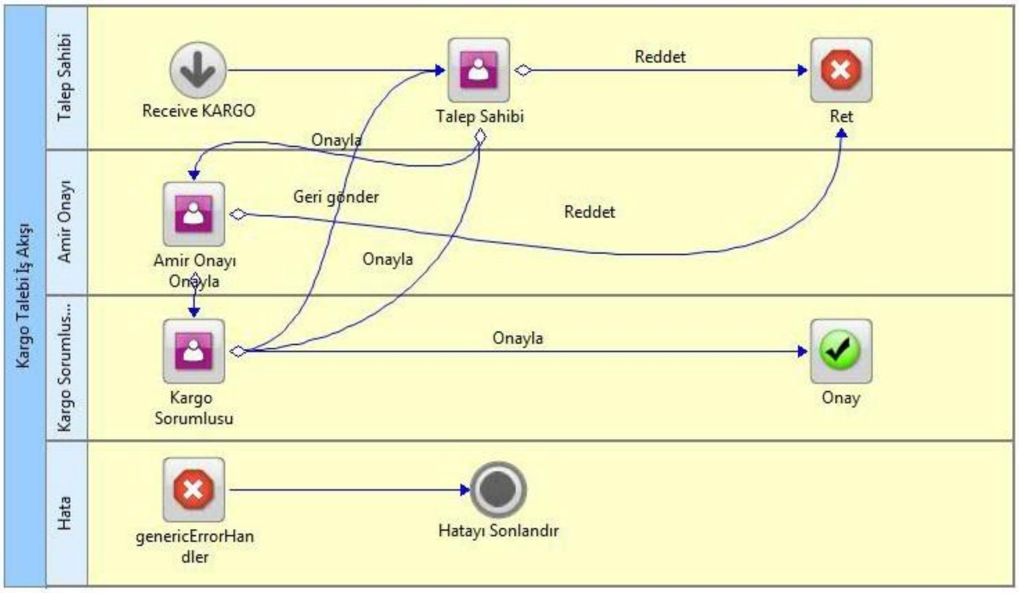


Figure 22: Shipment Business Process

This process shown in Figure 22 and it has 3 steps of which one is complex.

5.2.9. Plane Ticket Business Process

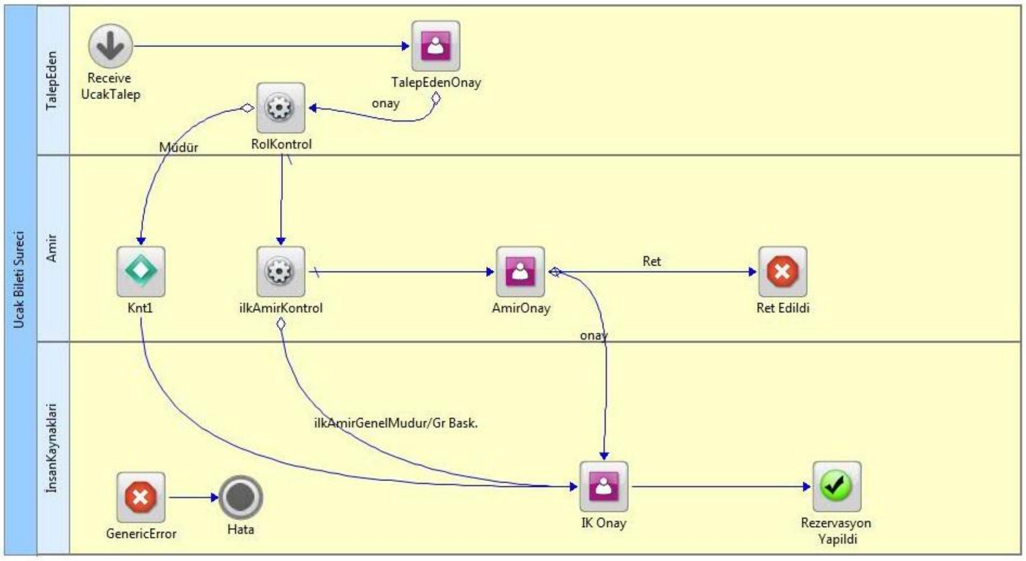


Figure 23: Plane Ticket Business Process

This process shown in Figure 23 has 3 steps of which one is complex.

5.3. The Estimation Formula

An agile methodology for business process software development is proposed. This methodology would decrease the development efforts compared with the traditional method. To measure the decrease in manpower, a formula is needed to estimate the development efforts. This formula will give a manpower value as if a business process was implemented using the Waterfall methodology.

The formula should be simple in order to estimate the development efforts more accurately. To avoid biasing, the simplest formula is chosen. In other words, the formula will take the size of the business process and will give the required effort for it as if it was developed using the Waterfall methodology. The number of steps in a business process is simply the best indicator for the size of the business process. Therefore, the number of steps is taken as an indicator for the size of the business process. Moreover, the complexity of a business process also affects the total effort very much. Hence, the number of complex steps is also taken into account. In short, the effort estimation formula should take two parameters and should give the estimated effort in person-day. First parameter is the number of simple steps, which is the difference between the number of steps and the number of complex steps. Second parameter is the number of complex steps. In (2), the model of the effort estimation formula is given.

$$\text{Effort} = a*s + b*s' + c \text{ person-day} \tag{2}$$

A step in a business process is taken as complex if the step requires user interactions except simple user decisions. Therefore the number of complex steps represents the complex part of the business process and as a result it shows the complexity of the business process.

Before applying the agile methodology, classical methods were applied in the same organization to develop business processes. It was a phased development. Analysis, design, coding, test, and deployment were conducted in sequence. Thus, the well-known Waterfall methodology has been applied for these business processes.

Nine business processes are developed using the Waterfall model. Table 35 shows the implemented business processes. The business process implementations are realized in a medium-sized organization. For this study, the guidelines described in [61] are used. The key business sector of the organization is electronics. Nevertheless, these business processes are not related to its business sector. These are common business processes which are related to purchasing, insurance, material request, duty order, quality control, and shipment. These are implemented typically by two analysts and two developers in cooperation with the customer.

Table 35: Implemented business processes

Process Name	# of Simple Steps(s)	# of Complex Steps(s′)	Effort (person-day)
Purchase Requisition	14	0	130
Insurance Claim	4	0	51
Material Request	1	4	102
Purchase Order	12	0	101
Duty Order	26	0	204
Quality Notification	6	5	156
Quality Tasks	2	2	61
Shipment	2	1	59
Plane Ticket	2	1	52

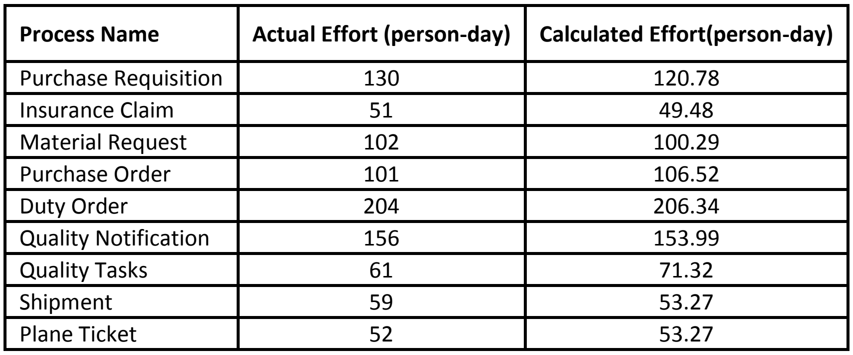
The effort spent for the implementation of business processes is recorded. Effort values are recorded on monthly basis by asking developers and analysts. From these data, a formula is derived to estimate the development effort. Using this formula, the development effort of a business process can be estimated as if the business process was implemented using the Waterfall methodology.

Table 35 shows also the number of complex steps in the processes. The last column shows the cumulative implementation effort in person-days. According to the table a simple estimation formula is derived using the least squares approach. The number of simple steps and the number of complex steps are taken as variables of the linear regression (2). Equation (3) shows the estimated formula in which s and s′ correspond to number of simple and complex steps, respectively.

$$\text{Effort} = 7.13*s + 10.91*s' + 20.96 \text{ person-day} \tag{3}$$

Table 36 shows the actual efforts and calculated efforts of the implemented processes.

Table 36: Calculated effort with formula



Process Name	Actual Effort (person-day)	Calculated Effort(person-day)
Purchase Requisition	130	120.78
Insurance Claim	51	49.48
Material Request	102	100.29
Purchase Order	101	106.52
Duty Order	204	206.34
Quality Notification	156	153.99
Quality Tasks	61	71.32
Shipment	59	53.27
Plane Ticket	52	53.27

5.4. Estimation Accuracy

The most frequently used method for software development effort estimations is expert estimation [34] [35].

Studies on software estimations show that the domain of the software is more important than the estimation method used [37]. In other words, the context is more important for more accurate estimates. Therefore for each software project, first similar projects should be found and the best estimation method for them may be chosen for the project. Probably, for this reason, the widely used estimation method is expert estimation. Another reason may be that there is no evidence that formal estimation models lead to more accurate estimates [34].

An interesting finding is that the accepted level of estimation accuracy is typically +/- 20% [34].

The reasons for schedule overruns are optimistic planning followed by changes from the original specifications [34].

According to [42] possible reasons for inaccurate estimates are respectively frequent changes, misunderstood requirements, and missing requirements.

Actually initial estimates for the software projects are more realistic. However, management and customer pressure makes to reduce estimates and these results in over-optimistic estimates [34].

The Mean Magnitude of Relative Error (MMRE) [36] is probably the most widely used evaluation criterion for assessing the performance of software prediction models. One purpose of MMRE is to assist us to select the best model. In paper [41], a simulation study has been performed, which demonstrates that MMRE does not always select the best model. Therefore, other evaluation criteria also should be used. In paper [38], two criteria are proposed: the weighted mean of quartiles of relative errors (WMQ) as a measure of accuracy and the standard deviation of the ratios of the estimate to actual observation (SDR) as a measure of consistency. In paper [39], instead of MMRE, boxplots or residuals are suggested in order to assess the accuracy of a prediction system. At present, there is no universal replacement for MMRE. Therefore, a combination of evaluation criteria should be used to assess the estimation models. Moreover, a framework may be used for the analysis of estimation errors. In paper [40], a 5-step algorithm is described for the analysis.

The effort estimation formula (3) can be analyzed using MMRE. In (4), the formula for the Magnitude of Relative Error (MRE) is given.

$$\text{MRE} = |\text{actual\_effort} - \text{estimated\_effort}| \,/\, \text{actual\_effort} \tag{4}$$

Here, the parameter “actual_effort” is used for the actual effort spent for a business process, and the parameter “estimated_effort” is used for the estimation of the effort to be spent for the business process. In (5), the formula for MMRE is given.

$$\text{MMRE} = \text{average percentage of MRE} \tag{5}$$

Table 37 shows the calculated MMRE of the estimation formula. In the first column, actual efforts of the business processes are given. In the second column, the estimated efforts are shown. In the last column, the calculated MRE values of the corresponding processes are given. The average of the MRE values gives the MMRE of the estimation formula.

Table 37: MRE's of the efforts

Process Name	**Actual Effort**	**Calculated Effort**	**MRE**
Purchase Requisition	130	120.78	0.0709
Insurance Claim	51	49.48	0.0298
Material Request	102	100.29	0.0168
Purchase Order	101	106.52	0.0547
Duty Order	204	206.34	0.0115
Quality Notification	156	153.99	0.0129
Quality Tasks	61	71.32	0.1692
Shipment	59	53.27	0.0971
Plane Ticket	52	53.27	0.0244
			MMRE = 5.4%

The MMRE of the estimation formula is found as 5.4%. This means that the error of the estimation formula is approximately less than 5%.

Another analysis method for estimation accuracy is PRED(N). In (6), the formula of PRED(N) is given.

$$PRED(N) = \frac{100}{n} \sum_{i}^{n} \begin{cases} 1 & if\ MRE_i < N/100 \\ 0 & otherwise \end{cases} \tag{6}$$

For PRED(30), the accuracy of the estimation formula is calculated. The result is shown in (7). This means that all the estimates are within 30 percent of the actual.

$$PRED(30) = 100\% \tag{7}$$

5.5. Discussing Software Quality Factors

In this section, the proposed methodology is discussed according to the software quality factors. In this dissertation, main objective is to define the proposed methodology. As a secondary objective, the proposed methodology is compared with the Waterfall methodology based on the effort spent during development. As a supplementary part, this section is included to discuss the proposed methodology based on the software quality factors.

The widely-used software quality factors are firstly defined in [70]. There are totally 11 factors which are correctness, reliability, efficiency, integrity, usability, maintainability, testability, flexibility, portability, reusability, and interoperability. In the following part, correctness, reliability, efficiency, integrity, usability, maintainability, flexibility, and portability will be discussed according to the observations.

Correctness means that a program should satisfy its specifications and fulfill customer's objectives. The new implementations require less number of versions of the business process implementations. In other words, after developments of old business processes, approximately 3 versions were released in the first 6 months. However, for the new business processes only one version was released in the similar period. Therefore, requirements were gathered better. Therefore, the software quality factor "correctness" is increased with the proposed methodology.

Reliability is another software quality factor. It means that a program should function with required precision. This kind of general business processes is not directly related to precision. Therefore, this quality factor is not applicable to the applied methodology. In other words, the reliability did not change.

The software quality factor “efficiency” means that the amount of computing resources will be needed to run a program. According to the observations, there is no significant change between the old and new implementations.

Another quality factor is integrity. It is the extent to which unauthorized access to software or data can be controlled. In fact, this is not an objective of the proposed methodology. Therefore, it is not applicable for comparison.

An important quality factor is usability. It is the effort required to learn, operate, prepare input for, and interpret output of a program. This factor is increased with the proposed methodology because number of user training sessions is decreased. In other words, in the old developments 2 user training sessions were arranged. However, after the new developments, one session was sufficed.

Maintainability is a software quality factor which measures the effort required to locate and fix errors in a program. The problems sent by users are easily fixed because the customer also knows the design and the coding of the program in a general sense. It makes easier to locate and fix the errors. For old developments, effort spent for a problem request was 0.39 person-days. For new developments, this number decreased to 0.32 person-days.

Flexibility means the effort required to modify an operational program. During the new implementations, the customer’s probable requests had been discussed. Therefore, the architecture of the software has been designed accordingly. For old developments, effort spent for a new issue request was 1.02 person-days. However, for new developments, this number decreased to 0.8 person-days. Therefore, flexibility is increased with the proposed methodology.

Another software quality factor is portability. Portability is the effort required to transfer a program from one hardware or software to another. This factor was not addressed in this research because the business environment did not require any porting.

CHAPTER 6

RELATED WORK

In this chapter, the related work is given. In the first section, the related methodologies are discussed. In the following section, the motivation is expressed. In the last section, concerns in developing a new methodology are provided.

6.1. Related Work

6.1.1. Business Process Software Development Methodologies

There are very few specialized methodologies for business process software development, and they are not complete and used widely. Therefore, detailed information about them is not available, only basic information about them is bounded to the related papers. In the following sections, these methodologies such as project-oriented development methodology, service-oriented development methodology, UML-based methodology, process-based methodology, and policy-based methodology are mentioned.

6.1.1.1. Project-oriented Development Methodology

In [8] and [3], a project-oriented development methodology is discussed. The methodology differs from the BPM lifecycle which includes the following phases:

process design, system configuration, process enactment, and diagnosis. The phases of the lifecycle are for the management of the business process related technologies whereas the phases of this project-oriented methodology are for development. The development methodology starts with a survey phase. In this phase, the goals of the project are defined, the project team is established, and initial information on the application environment is gathered. The project managers then decide which business processes will be selected. The main result of the Survey phase is a reviewed as-is business process model. The design phase is next, in which the as-is business process model is analyzed and optimized to reflect the overall goals of the business. The output of the design phase is the to-be business process model. Then the implementation phase starts. The test phase is next, which includes lab test and field test. If the tests are successful, the operational phase is reached. This methodology takes the as-is model and tries to optimize it to reach the to-be model.

As it has been mentioned before, the Design phase aims at analyzing and optimizing the as-is business process model. For that reason, this model serves as an input for this phase. The first set of activities deal with organizational and technical modeling of the so called to-be business process model, which represents the optimized business process which will be supported by the new application. The to-be workflow modeling activity is subdivided into workflow process modeling, organizational modeling and data modeling. The output of this activity is a workflow schema which reflects the contents of the to-be business process model, enhanced by workflow-specific features.

6.1.1.2. Service-oriented Development Methodology

The service-oriented business process software development methodology is based on a roadmap that comprises one preparatory phase to plan development, and five distinct phases that concentrate on business processes: analysis and design (A&D), construction and testing, provisioning, deployment, and execution and monitoring [18].

The stepwise transition through these phases tends to be incremental and iterative in nature and should accommodate revisions in situations where the scope cannot be completely defined a priori.

The methodology defines a foundation of development principles for web services which are assembled to construct real-world business processes. Currently, the methodology is enriched with patterns, and design and coding templates that are derived from empirical material. Next, an integrated toolset supporting the phases of the methodology will be developed.

6.1.1.3. UML-based Methodology

UML is a general-purpose modeling language. UML is also used in business process software development. In [11] a business process software development methodology has been developed that uses UML. In addition to the methodology, a process template development methodology has been developed to allow the practical reuse of business process models. These works reduced the business process software development time.

6.1.1.4. Process-based Methodology

Another business process software development methodology rests on processes. The process approach [16], where process is the new first class entity, can be applied to all the tasks. Typical business applications today are not adaptable. New versions have to be developed whenever the processes being supported change. Applications written in BPML [43] will be direct representations of business processes. By producing applications in BPML, within the framework of a suitable business process management system, organizations should be able to produce applications which adapt to organizational change. Process and application effectively will converge, the mutability of each reflected in the other.

Early practical experience with BPML in real-world applications has shown that it offers a considerable step forward in supporting a wide variety of dynamic processes and process tools, including process discovery, design, deployment, execution, operations, optimization and analysis. If the language is taken up by the IT industry, a wide variety of new ‘process technologies’ is likely to emerge, in the same way that previously, in the era of data management, myriad applications have been built upon the RDBMS platform.

6.1.1.5. Policy-based Methodology

The work presented in [19] proposes an innovative approach for business process modeling and enactment, which is based on a combination of protocols and policies. The key idea is to capture meaningful interactions reusably as protocols.

In order to design processes, first protocol specifications are determined. Then, roles are extracted from them, and augmented with business policies to come up with a business process. The protocols and policies are separate rule-bases. However, there is a clean, uniform interface between the two; they are tied together through the use of policy variables in the protocol rules. This greatly enhances the modularity of the software. New policies can be plugged in with only local changes. In addition, the methodology advocates refining protocols to yield more robust protocols.

Protocols are registered to protocol repositories. Composite protocols can be constructed or existing protocols can be reused. As the messaging interface, JADE (Java Agent Development Framework) is employed. And for rule system for both the protocol rules and the policy rules, Jess is used. Jess offers seamless interoperability with Java objects.

Developing business processes for open systems poses significant challenges because of the complexity of the interactions and the autonomy of the partners. Traditional technologies such as workflow systems lack the flexibility and agility that modern business processes need.

6.1.2. Comparison of the Business Process Software Development methodologies

The introduced business process software development methodologies are not detailed in the respective papers. Therefore, comparing them was not possible. This situation also addressed the need for a detailed methodology for business process.

6.2. Motivation

Agility has become a key competency today as businesses face changing and uncertain environment [74]. In order to position themselves to respond to changing business conditions, organizations must be able to adapt quickly by managing the necessary business processes [75]. Adapting to the changing conditions keeps organizations to stay aligned with defined business goals and to adapt to the new business goals. Moreover, organizations grow with their business processes. Therefore, business processes have to be extended while keeping them efficient.

Business process agility has increasingly become essential for survival for today's organizations [74]. Even if organizations gain a simple cost reduction, agility is a critical characteristic for today's organizations. Business process agility is defined as the ability to manage a business process to accommodate required needs of the organizations [74] [75]. Business processes must be agile to improve the ability to exploit opportunities for innovation and competition [76]. Information technologies today are considered a significant digital platform for business process agility and are expected to facilitate agility [74].

The need for a fast BPM development technique has been felt in the organization for long. For the last decade, there has been 3 to 5 BPMs developed every year, each taking about one-year to develop and enact. Any reduction in the development time of those BPMs would definitely benefit the organization very much. The fundamental motivation for undertaking this research has been definitely, the speed up in BPM development.

For the purpose, other methodologies were slow and not open enough to adapt. The proposed approach in developing a new methodology exploited some shortcomings of other general approaches that could be adapted for BPM development. An analysis of such properties of general approaches are provided in depth, in Section 4.11. Also supported with the Software Quality Factors discussed in Section 5.5, we devised a new methodology considering the existing shortcomings:

- Speed of development for BPMs
- More efficient capturing of BPM requirements

6.3. Concerns in Developing a New Methodology

6.3.1. Additional Requirements for Methodology

A business process model should be directly enacted or at least should be converted into executable languages like BPEL [8].

Traditional technologies such as workflow systems lack the flexibility and agility that modern business processes need. Therefore, business process management should be agile and flexible [19].

Prototyping should be applied to most of the projects. Prototyping is a very helpful activity in building workflow applications. The main reasons are as follows: Firstly, the organizational requirements of the business process can be validated in prototypes by the users of the system. It turned out that system usability and front-end design are important factors for the acceptance of the new technology by the workflow participants. Secondly, technical restrictions can be tested in an early project stage, which can save considerable resources in later project stages [8].

Process templates should be employed to develop business processes fast and to adapt to fast-changing environment [11].

6.3.2. Agile Development and BPM

Agile software development methodologies [30] [31] are based on iterative development. Agile methods encourage frequent inspection, adaptation, teamwork, and self-organization to allow for rapid delivery of high-quality software.

Scrum [14] is an agile development methodology which is a simple process for managing complex projects. Scrum's rules and practices are few, straightforward, and easy to learn.

All the practices of Scrum are based on an iterative and incremental process skeleton. The skeleton consists of iterations and daily inspections. Development is done through iterations one after another. The output of each iteration is an increment of product. Daily inspections occur in iterations. In daily inspections, the individual team members meet to inspect each other's activities and make appropriate adaptations. Driving the iteration is a list of requirements. This cycle repeats until the project is no longer funded.

Scrum implements this iterative, incremental skeleton through three roles. These are the Product Owner, the Team, and the ScrumMaster. All management responsibilities in a project are divided among these three roles. The Product Owner is responsible for representing the interests of everyone with a stake in the project and its resulting system. The Product Owner achieves initial and ongoing funding for the project by creating the project's initial overall requirements, return on investment (ROI) objectives, and release plans. The list of requirements is called the Product Backlog. The Product Owner is responsible for using the Product Backlog to ensure that the most valuable functionality is produced first and built upon; this is achieved by frequently prioritizing the Product Backlog to queue up the most valuable requirements for the next iteration. The Team is responsible for developing functionality. Teams are self-managing, self-organizing, and cross-functional, and they are responsible for figuring out how to turn Product Backlog into an increment of functionality within iteration and managing their own work to do so. Team members are collectively responsible for the success of each iteration and of the project as a whole. The ScrumMaster is responsible for the Scrum process, for teaching Scrum to everyone involved in the project, for implementing Scrum so that it fits within an organization’s culture and still delivers the expected benefits, and for ensuring that everyone follows Scrum rules and practices.

Extreme Programming (XP) [31] [32] is another agile software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent releases in short development cycles where new customer requirements can be adopted.

Other elements of Extreme Programming include: programming in pairs, unit testing, a flat management structure, simplicity, and frequent communication with the customer and among programmers. The methodology takes its name from the idea that the beneficial elements of traditional software engineering practices are taken to "extreme" levels, on the theory that if some is good, more is better.

In recent studies, agile approaches are applied to BPM and social media are used to provide BPM with agility. In [27] there is a mention of an agile methodology but it is not completely reflected, except for only a feel of it. In the paper the embedding of social software features, such as collaboration and wiki-like features, in the modeling and execution tools of business processes is proposed. These features will foster people empowerment in the bottom-up design and execution of business processes.

In [28], social software is advocated to satisfy the key requirements for enabling agile BPM development. The four features of social software are applied: weak ties, social production, egalitarianism and mutual service provision. Organizational and semantic integration and responsiveness have been identified as the main requirements for implementing an agile BPM life cycle. This paper does not seem to contain a complete methodology. It is a nice study for the mostly social and business related principles in the design of agile processes. Specifically, they promote the usage of social software as a tool.

In [29] agile principles have been applied to an existing business process. It investigates how stakeholder involvement affects business process redesign. Constant stakeholder involvement in redesigning processes aligns the redesigned process with stakeholder needs and requirements. Though at times meeting with stakeholders may be difficult due to geography, language, or other barriers, working with stakeholders at each step of the redesign yields invaluable results during a business process redesign cycle. It reports that business process redesign cycle time is improved based on a short case study. This paper does not offer a general process but offers to use agile principles in redesigning processes.

In [33] an agile approach is presented for business process management. It depends on “sense and respond” loops. This infrastructure is event-based and has 5 stages which are sense, interpret, analyze, decide, and respond. Its aim is to develop an agile business process management platform with “sense and respond” capabilities. This platform will take existing processes and adapt them to the fast changing environment using agile approaches.

At the beginning of this dissertation studies, the combination of the concepts “agile” and “business process management” was proposed in 2008. Then, in the same year, the dissertation is shaped under the title “An Agile Business Process Software Development Methodology”. In the following years, some studies are appeared related to the two concepts. For example, in 2012, a case study is conducted, which investigates the suitability of the agile methodologies to business process software development [67]. In that paper, an agile method is applied to a business process software development project. As a result, it is recommended that agile methods are appropriate for complex business process software development projects, which need flexibility and adaptability. In that work, the standard outcomes of the agile implementations such as “customer satisfaction” and “cost saving” were reported as achieved. This kind of work does not propose a specialized methodology but only presents application of the well-known agile methods to the projects.

6.3.3. Method Engineering

Method Engineering was investigated as an approach for this research. However, due to the need for developing a new methodology this avenue was abandoned. Method

Engineering suggests the selection of method fragments out of existing methodologies for each new problem.   This research does not suggest a different approach per every new problem. A new methodology was developed that included parts that were not existing in other approaches. However, for its being a related topic Method Engineering will be introduced in this section.

Information Systems (ISs) affect people excessively. Moreover, ISs are becoming more complicated and expensive. Therefore, they should be developed in an effective and efficient way. For this purpose, many methods have been described. However, every IS development project has its own special properties. Hence, a special method should be provided. In real life, actually this happens. In other words, every project is developed with some differentiation from other projects. The reason for this is that one method does not fit all the projects. Therefore, for each project a method should be tailored. Building a special method from existing methods is called method engineering. This field is named Methodology Engineering initially.

In [48], Methodology Engineering is defined as a meta-modeling for designing and implementing Information Systems Development Methodologies (ISDMs). However, this term is renamed as Method Engineering in [46] and it is defined as an engineering discipline to design, construct and adapt methods, techniques and tools for the development of ISs. The term “Method Engineering” is generally accepted since the latter definition. In this context, a technique is a procedure to perform a development activity using a prescribed notation. In addition, a tool is a means to support a part of a development process, which is possibly automated. Method engineering deals with all engineering activities related to methods, techniques and tools. The application of ISs development methods is effective with a proper automated support tool.

In [51], a survey on Method Engineering is made with a view to identifying the important problems and research directions being followed to solve these. Bare bones of Method Engineering are described in both practice and theory in [50].

Method Engineering constructs methods from existing methods. The place where the existing methods are stored is called Method Base [49]. A method can be divided into its building blocks. These building blocks are named method fragments. Alias, method engineering is choosing suitable method fragments from Method Base for a project. The combination of these method fragments builds a new method for the project, and this is also stored in Method Base for later projects.

Method fragments are divided into two categories: technical method fragments and conceptual method fragments [49]. Technical method fragments are divided into three subcategories: tool fragment, repository fragment, and process manager fragment. Tool fragment describes the CASE tool. Repository fragment describes the database of the CASE tool. Process manager fragment is the guidance for the CASE tool user. On the other hand, conceptual method fragments are divided into two as product fragment and process fragment. A product fragment is a product generated during the ISs engineering process, whereas a process fragment is the activities of that engineering process.

In addition to method fragments, Method Base houses other concepts like persons in methods, their roles, and their functions [49]. Method Base also keeps the properties of method fragments, which help to choose suitable fragments for the project.

In [52], a Computer Aided Method Engineering (CAME) tool is explained. The central repository of it contains the Method Base in which method fragments are kept. With a CAME tool, a method engineer builds the specific method for the desired project. A CAME tool has its modeling environment which is called meta-model. In [50], a meta-model is proposed for Method Engineering.

In order to make Method Engineering efficient, some concepts, strategies and frameworks are utilized. In [48], four design strategies are proposed, which guarantee that a method would fit the situation, be complete, and its fragments are proved.

In [59], the importance of reuse is emphasized. Reusing existing knowledge accelerates building of situational methods. The reuse mechanisms which are used in different research areas can be transferred to Method Engineering. The relevant reuse mechanisms are identified by reviewing the literature. A preliminary analysis showed that at least three commonly accepted approaches can be used in Method Engineering. These reuse approaches are patterns, components, and reference models. In [51], Method

Engineering is investigated using four worlds framework. These worlds are subject, representation, development and usage worlds. In the subject world, methods are situated. In general, methods are considered as domain independent. However, they are context dependent. In the representation world, the meta-modeling around which the whole area of Method Engineering is developed is investigated. In the development world, the problems in the existing CAME environments are mentioned. The refinements are needed for the CAME environments and their integration with CASE environments. In the usage world, it is pointed out that the straight-forward modeling of current methods is inadequate for solving any unsolved problems of IT.

In [53], multi-perspective software development is examined. In multi-perspective software development, there exist multiple development participants who hold multiple views on a system and its domain. Each participant has its own client who has different, conflicting and complementary requirements of the requested software system. The developers should elicit, specify, analyze and validate these requirements, and then they should design and build a software system to satisfy the requirements. For this conflicting problem, the paper proposes Viewpoint Frameworks which describes multi-perspective development.

For each IS development, the situation in which the IS is developed should be taken into account. This situational approach is called Situational Method Engineering (SME). The basic concepts of SME are described in [49] [52] [57]. SME tries to optimize standardization and flexibility. Namely, every IS should be developed in standards as much as possible. On the other hand, every project has its own situation and its situational factors should be considered. This brings the IS development to a tailored method which is standard as much as possible and is flexible enough. This requirement is defined in [49] as controlled flexibility. Controlled flexibility is achieved by SME. The resultant method is called situational method.

In [47], the SME literature is surveyed. In [57], the feasibility of SME is studied. In order to get a successful situational method, the prerequisites which have to be fulfilled are identified and their inherent complexity is investigated.

There are meta-models for SME and high level process models for method construction. In [54], the MEMA-model is described. It is a method engineering approach. The core part of the approach is the modeling framework. It can also be considered as a set of guidelines to infer a method advice. In [56], the Noesis meta-modeling technique is defined. This technique captures recursive and decompositional structures with a complete and minimal family of transformations in order to obtain situational methods by the assembly of the method fragments. In [58], a meta-model is proposed. This meta-model differs from the existing meta-models while paying attention to procedural information. Such information provides guidance to developers.

In [50], practices from SME are given. In [55], the challenges of developing software for mobile systems are examined. The characteristics of mobile systems and proper methodologies for them are reviewed. It has been shown that agile methodologies are appropriate for the development of such systems. Based on this assumption, a new agile method is engineered using the Hybrid Methodology Design approach. The proposed agile method is a risk-based method built from the existing agile methodologies Adaptive Software Development (ASD) and New Product Development (NPD).

In [60], a set of high level process patterns for agile development are provided. These process patterns are derived from seven agile methodologies. The process patterns depend on the generic proposed methodology called Agile Software Process (ASP). A method engineer can tailor an agile method from these process patterns based the generic template ASP. This method engineering is only for agile development and it is a kind of paradigm-based SME.

Method Engineering and the proposed methodology have some similarities. In the construction, they both use existing methods. The proposed methodology is a generic agile methodology for its special domain, whereas in Method Engineering every time a new method is generated for a project. In addition, Method Engineering may use the proposed methodology to build special methods for similar domains. This usage may be a future work.

6.3.4. Case Study and Validity Threats

Case study is one way of doing social science research. In [62], application of case study to software engineering is explained. In case studies, subjects and objects are not selected from statistically representative samples. Instead, researchers obtain the findings through the analysis in depth of typical or special cases. A survey on case study, an introduction to case study methodology, and guidelines about case study can be found in [61]. The natural context is crucial for a case study. Moreover, the case study should be contemporary. The researchers try to understand this contemporary phenomenon in its natural context and the interaction with its context.

In general, case studies have four characteristics [62] where:

- “How” or “why” questions are answered.
- The researcher has little control over events.
- The case is a contemporary phenomenon within a real-life context.
- Boundaries between the phenomenon and the context are not clear.

Case study research is conducted in the following phases [62], which may be iterated:

- Design: objectives are decided and the case is defined.
- Data collection: data collection techniques and data sources are planned. Data collection methods include interview, observation, and usage of archival data.
- Collecting evidence: the case study is executed for data collection.
- Analysis: collected data is analyzed. A chain of evidence is maintained from the findings to the original data.
- Reporting: the report should include sufficient data and examples to allow the reader to understand the chain of evidence.

Researchers and reviewers in conducting and reviewing case studies may consult to the checklists for case studies [65], which are derived using a systematic qualitative approach.

In [63], empirical methods in software engineering are investigated. The research methods “controlled experiments”, “case studies”, “surveys”, and “post-mortem analysis” are briefly introduced. There are mainly two types of research paradigms which have different approaches to empirical studies. One is qualitative research concerned with studying objects in their natural setting. The other is quantitative research concerned with quantifying a relationship or to compare two or more groups.

In case studies, confounding factors should be minimized [63]. A confounding factor is one that cannot be distinguished from another factor in a case study. Moreover, data should be validated as correct before making any analysis. In other words, descriptive statistics should be applied before analysis. For example, this covers identifying and handling outliers. An outlier is an unexpected value in the data set. Every outlier must be investigated in order to determine whether it should be discarded, corrected, or included. In addition, it should be decided whether there is an effect of independent variables to the dependent variables.

Before presenting the results of a case study, the results should be assessed whether they are valid [63]. In other words, the study and the results should be validated against threats in empirical research. These are called validity threats. A validity threat is a specific way where the results might be wrong. In [66], a survey on validity threats can be found. In [63], four main types of validity threats “conclusion”, “internal”, “construct”, and “external” are discussed.

- Internal: Internal validity is concerned with assuring that there is a cause-effect relationship between the independent and dependent variables. There should be a statistically significant relationship between the treatment and the outcome.
- External: External validity is concerned with the generalization of the results of the study. It is important to generalize the results outside the scope of the study.
- Conclusion: Conclusion validity is concerned with the conclusions. The conclusions should be correct regarding the relationship between treatments and the outcome of the experiments.
- Construct: Construct validity is concerned with the relationship between the concepts and theories behind the experiment and what is measured and affected.

In [64], the importance of external validity threats is emphasized. In making generalizations, a series of questions are outlined to better address external validity.

6.3.5. Data Analysis

The present situation where the classical Waterfall methodology is applied can be analyzed by On-Line Analytical Processing (OLAP) tools. OLAP is a category of software technology that enables analysts and managers to gain insight into data [24]. A wide variety of possible views of information that has been transformed from raw data is accessed by OLAP, which reflects the real dimensionality of an enterprise. OLAP tools provide multidimensional analysis to the underlying information.

In [24], different proposals for multidimensional data cubes, which are the basic logical model for OLAP applications, are surveyed. Traditional relational data models are not powerful enough for deep analysis. Data cubes provide this functionality by summarizing, viewing, and consolidating the data available in the databases.

There are basically 3 forms of multidimensional model, which are star, snowflake, and fast constellation [25]. In a star schema, there is a large central table (fact table) containing the bulk of the data, with no redundancy, and a set of smaller tables (dimension tables), one for each dimension. The snowflake schema is a variant of the star schema model, where some dimension tables are normalized. Further splitting the data into additional tables results in a shape similar to a snowflake. Sophisticated applications may require multiple fact tables. In a fast constellation schema, there are more than one fact tables, which share dimension tables. A fact constellation schema can be considered as a collection of star schemas.

The analysis of the application domain has been conducted with an OLAP analysis. As a multidimensional model, a star schema has been used.

6.3.6. Software Quality Factors

Software quality is related to the excellence of the software. In [72] [71], various definitions of software quality can be found. One of the best definitions is that software quality is the extent to which an industry-defined set of desirable features are incorporated into a product so as to enhance its lifetime performance. From the definition, it can be extracted that the product has quality features which have been incorporated from the beginning and enhance its lifetime performance.

Characteristics that influence software quality are called software quality factors. Different quality factors and models are examined in [69] [71] [72]. The well-known quality factors are firstly defined by [70]. Its quality model is called McCall Quality Model. In McCall Quality Model, there are 11 quality factors, which are a reduced set of the 55 quality characteristics [70]. These quality factors are correctness, reliability, efficiency, integrity, usability, maintainability, testability, flexibility, portability, reusability, and interoperability.

In [71], three major perspectives of the McCall Quality Model are described. These are product revision, product transition, and product operations. Product revision which is the ability to undergo changes includes maintainability, flexibility, and testability. Maintainability is the effort required to locate and fix an error. Flexibility is the ease of making changes. Testability is the ease of testing the program.

The other perspective is product transition which is the adaptability to new environments. It includes portability, reusability, and interoperability. Portability is the effort to transfer a program from one environment to another. Reusability is the ease of reusing software in other applications. Interoperability is the effort required to couple the system with another.

The third perspective of the McCall Quality Model is product operations which are the operation characteristics. It includes correctness, reliability, efficiency, integrity, and usability. Correctness is the extent to which a program fulfils its specifications. Reliability is the ability of the system not to fail. Efficiency is the usage of the computing resources. Integrity is the protection of the system from unauthorized access. Usability is the ease of the software.

CHAPTER 7

APPLICATION OF THE PROPOSED AGILE METHODOLOGY: A CASE STUDY

Case study is a kind of research method which is defined in detail in [62]. A research method can be used in one of the three purposes: explanatory, descriptive, and exploratory. Similarly, case studies may be explanatory, descriptive, and exploratory. In explanatory case studies, the “how” and “why” questions are answered mostly. The cases in this chapter are based on an explanatory case study.

According to [62], a case study should have 6 phases. These are plan, design, preparation, data collection, analysis, and reporting. In the plan phase, the research questions are identified. In the design phase, the method which will be used in the case study research is specified. Also, the procedures to maintain case study quality are defined. In the preparation phase, the case study protocol is developed. In the data collection phase, the protocol is followed, and the case study data is collected from the sources. If the sources are multiplied, the evidences would be more confident. Also, the chain of evidence is maintained. In the analysis phase, the collected data is analyzed in either qualitative or quantitative way. Rival explanations are explored. The data is displayed apart from interpretations. In the report phase, the audience of the study is defined, and enough evidence is displayed for each conclusion. All these phases will be explained in the following sections consecutively.

Case study research should be protected against threats to validity [61], maintain a chain of evidence, and investigate and test rival explanations. Validity threats will be discussed in a separate section after the section “Analysis Phase”.

7.1. Plan Phase

In the plan phase, first of all the case is described. The case is the application of the proposed agile methodology. There are two cases:

- Case A: Consumable Goods Business Process Software Development Project
- Case B: Local Duties Business Process Software Development Project

These cases are selected because of being general in all organizations in order to support the generalization of the proposed methodology.

Each case is applied to a business process software development project in an organization. In the organization, the stakeholders of the cases have more than one project concurrently.

The research questions of the case study are identified. There are two main research questions:

- Is the proposed methodology applicable and generalizable?
- How much effort can be saved using the proposed agile methodology?

The proposed methodology has been defined in Chapter 4. Basically, it will be applied to the business processes development projects. Effort values will be recorded and compared with the values of the Waterfall methodology. The latter values will be estimated using an estimation formula derived in Chapter 5. Effort saving would be obtained with the proposed methodology.

7.2. Case Study Design

The research is simply proposing an agile methodology for business process software development. The proposed methodology is compared with the Waterfall methodology to measure the effort saving.

The domain of cases is business process software development projects. These kinds of projects usually have many actors. Being many actors increases the complexity of the projects. And, the stakeholders in the projects have more than one project concurrently. In other words, they are business process software developments projects with stakeholders having more than one project concurrently.

The followings are the rationale to select the business processes:

- The business process should be a general business process for nearly all organizations.
- The business process should have different organizational units. Actually the owner of the business process should be different from the implemented ones. In short, the cases have different organizational units.
- The business process should have at least 3 steps.

The effort estimation formula is obtained using the effort values of the business process software development projects implemented using the Waterfall methodology. Those projects have the following criteria:

- The projects were developed usually by two developers and two analysts with the related people from the customer.
- The developers and the analysts were trained before beginning the projects about business process software development.
- The implemented projects were selected from the common business processes. Actually, the organization, where the business processes had been implemented, deals with electronics as a main business. However, these selected processes are not related to its main business. Actually, they are general and standard business

processes nearly for all companies. Therefore, every organization uses very similar types of the selected business processes. They are related with purchasing, shipment, duties, quality, insurance, material, and travel.

- These business processes have been done in Waterfall methodology. First of all, analysis is done. Then design and coding are realized.
- The customers of these projects are the people from different organizational units. And, they nearly all changed from one project to another. Therefore, each project was implemented with new customers.

In this chapter, bound to these criteria, two business process software development projects are implemented by changing only the applied methodology. In other words, instead of Waterfall methodology the proposed methodology is used in the case study.

There are a pool of 5 developers and a pool of 8 analysts. These pools are employed to select the stakeholders of the business processes. The customer of the cases changed every time. Therefore, for each case there are trained developers and analysts with new customers.

The cases are implemented using the webMethods BPM Suite. The business data is held on the SAP ERP System. The connections to the ERP system are realized using web services. The business processes have strong relations with the SAP ERP system. The connections are realized using web services. There are also other systems with which the business processes have little relations. Their connections are also realized using web services. The webMethods system is installed on the SQL Server as a database. The programming is usually realized in java. Also, to develop supporting components, SAP ABAP and c# programming are also employed.

The proposed agile methodology depends on agile methodologies. Agile methodologies decrease development time. Therefore, this proposed methodology would also decrease the development time and save effort.

In the proposed methodology, all the tasks are done in iteration bases. Duration of an iteration base is selected as 5 weeks according to the observations. The cases would be implemented in many iteration bases. In each iteration base, there would be small-team meetings arranged weekly. In small-team meetings, training sessions would be conducted to improve the vision of the customer. If consolidation of the works is needed, whole-team meetings would be organized. In the first iteration bases, use cases would be used for determining the requirements. Also, site inspections would be realized. Then, business process model would be drawn in BPD diagrams. Main input output screen would be designed to determine basic input output requirements.

The effort estimation formula predicts effort of a business process software development project implemented using the Waterfall methodology. The estimation formula in these cases will be used to predict the effort values of the cases as if they were implemented using the Waterfall methodology. The effort values would be collected from the stakeholders at the end of each iteration base. Moreover, the effort values are also entered into the help desk system by developers and analysts. These values would also be used to check the effort values. If there is a mismatch between these two, then, the effort values would be checked by the developers and analysts again. Therefore, the correction of the effort values would be provided by two sources.

7.3. Preparation Phase

The protocol is a major way of increasing the reliability of case study research and is intended to guide the investigator in carrying out the data collection from the cases. The protocol is defined in the following part.

First of all, cases are determined. The proposed methodology described in Chapter 4 will be applied. Site inspections are planned in the first iteration bases. Actually, missing actors would be investigated in the site inspections as well as missing requirements. Site inspections are selected according to heavy or special usage of the business processes. The users of the business processes will be interviewed and the paper forms of the processes will be examined. For Case A, Administrative Affairs, Marketing, and Warehouse Departments are planned to visit. For Case B, Human Resources, Logistics, and Material Supply Departments are planned to visit.

The cases will be implemented by selecting developers and analysts from a pool. There are totally 5 developers and 8 analysts in the pool. Their skills and experience are nearly the same and they were trained before business process software development projects.

Effort values will be collected for the cases. They will be collected from the developers and analysts by asking at the end of each iteration base. Moreover, there is also help desk system and the effort values are entered also by each stakeholder. These two resources will be compared to refine obtained effort values.

The effort estimation formula will be used to calculate estimated effort values. The results will be compared and analyzed. Iteration bases will be analyzed. The results will be reported.

7.4. Data Collection Phase

The case study should present adequate data and the raw data should be available for independent inspection. The reader of the case study should be allowed to follow the derivation of any evidence from initial research questions to ultimate case study conclusions. Moreover, the reader should be allowed to do further analysis.

Two business processes have been implemented with the proposed agile methodology and recorded the development effort. On the other hand, the estimation formula will help us to estimate the development effort of the new business processes as if they were developed using the Waterfall methodology. As a result, two values of development effort would be obtained for each new business process. It was argued that the proposed agile methodology will decrease the development efforts. The business processes are general business processes on consumable goods and local duties. And it is developed in the same conditions with the previously developed business processes except the methodology.

Table 38: Characteristics of the cases

	Start Date	End Date	Duration (days)	# of Steps(s)	# of Complex Steps(s')	Actual Effort	# of Iteration Bases	# of Whole team Meetings
CASE A	02.12.2011	23.07.2012	231	3	0	34	7	2
CASE B	08.05.2013	01.01.2014	232	2	1	41	7	1

In Table 38, the characteristics of the cases are shown. The start dates and end dates of the cases are displayed. Case A is implemented in 231 days, whereas Case B is implemented in 232 days. Both cases have 3 steps in their business process models. However, only Case B has a complex step in its process model. Effort spent for Case A is 34 days, whereas 41 days are spent for Case B.

In Figure 24, the process model of the Case A is shown. It has 3 task steps, which are simple.

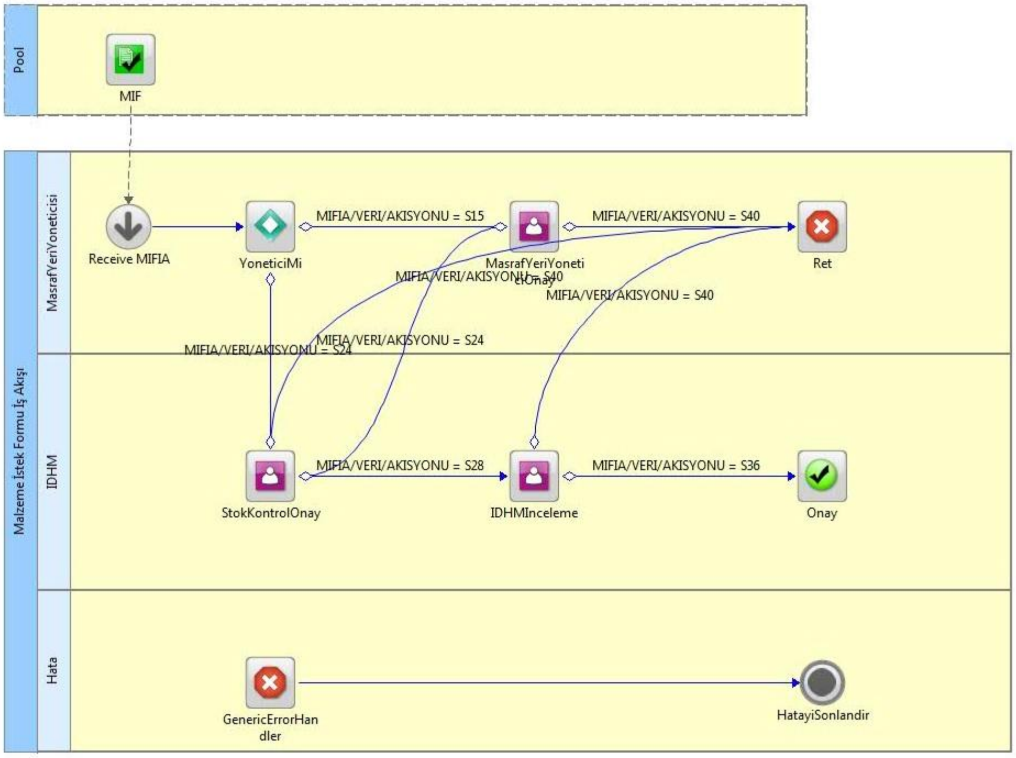

Pool
MIF
Malzeme İstek Formu İş Akışı
MasrafYeriYoneticisi
Receive MIFIA
YoneticiMi
MIFIA/VERI/AKISYONU = S15
MasrafYeriYoneticiOnay
MIFIA/VERI/AKISYONU = S40
Ret
MIFIA/VERI/AKISYONU = S40
MIFIA/VERI/AKISYONU = S40
MIFIA/VERI/AKISYONU = S24
MIFIA/VERI/AKISYONU = S24
IDHM
StokKontrolOnay
MIFIA/VERI/AKISYONU = S28
IDHMInceleme
MIFIA/VERI/AKISYONU = S36
Onay
Hata
GenericErrorHandler
HatayiSonlandir

Figure 24: Consumable Goods Request Business Process

In Figure 25, the process model of Case B is shown. It has 3 steps. One of them is complex.

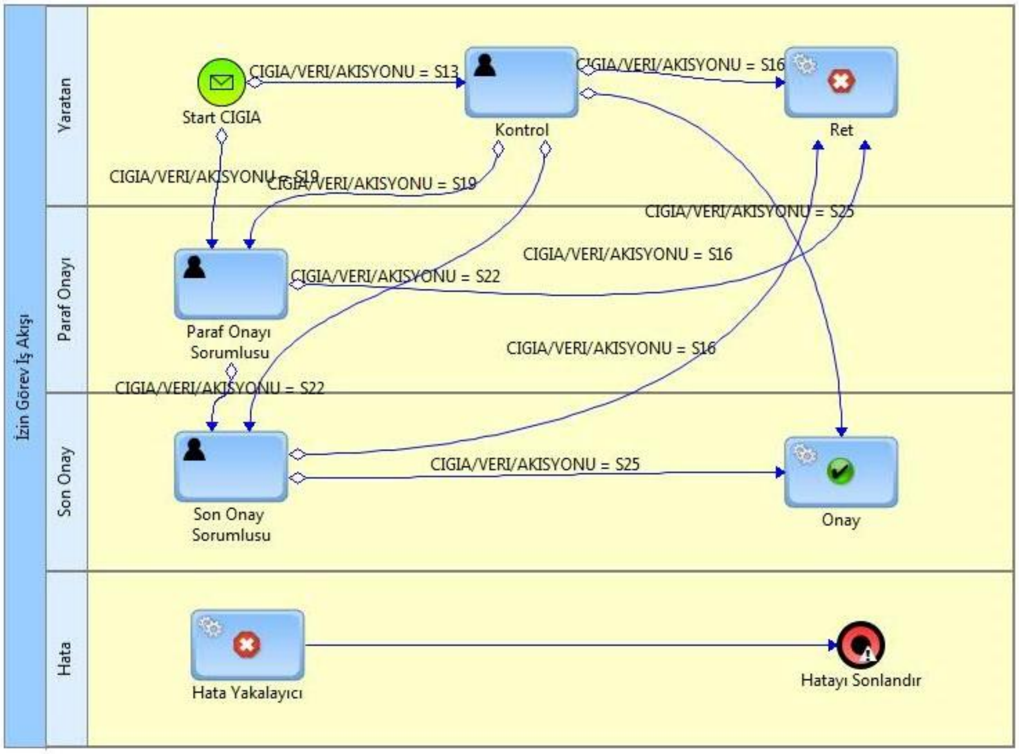

İzin Görev İş Akışı
Yaratan
Paraf Onayı
Son Onay
Hata
Start CIGIA
CIGIA/VERI/AKISYONU = S13
Kontrol
CIGIA/VERI/AKISYONU = S16
Ret
CIGIA/VERI/AKISYONU = S19
CIGIA/VERI/AKISYONU = S19
CIGIA/VERI/AKISYONU = S25
CIGIA/VERI/AKISYONU = S16
CIGIA/VERI/AKISYONU = S22
Paraf Onayı Sorumlusu
CIGIA/VERI/AKISYONU = S16
CIGIA/VERI/AKISYONU = S22
Son Onay Sorumlusu
CIGIA/VERI/AKISYONU = S25
Onay
Hata Yakalayıcı
Hatayı Sonlandır

Figure 25: Local Duties Business Process

Both cases are implemented in 7 consecutive iteration bases. In Table 39, the collected effort values are shown based on iteration bases.

Table 39: Collected effort values based on iteration bases

	Iteration Base 1	Iteration Base 2	Iteration Base 3	Iteration Base 4	Iteration Base 5	Iteration Base 6	Iteration Base 7	Total (in days)
CASE A	5	8	6	3	4	3	5	34
CASE B	6	10	8	5	4	3	5	41

These effort values are collected from the developers and analysts of the cases at the end each iteration base. Moreover, the same stakeholders are also entered their effort values to the help desk system of the organization. Whenever there is an inconsistency between the collected and entered effort values, the stakeholders are asked to reconcile to effort values. In other words, Table 39 is obtained by reconciling effort values of two sources.

In Case A, in the first iteration base use cases are drawn. In the second iteration base, the process model is drawn. Moreover, a whole-team meeting is organized to consolidate the process model. For this case, a second whole-team meeting is needed before going to pilot. Therefore, in the 6th iteration base another whole-team meeting is organized to arrange the organization units which take part in the pilot application phase. In the second iteration base, also main input output screen design is built. In the third iteration base, this design is refined.

Site inspections are realized in the first and second iteration bases. The requirements of the users are examined in these site inspections in detail. These requirements are added to the product backlog. In addition, determined requirements in iteration bases are added to the product backlog.

Coding works of the Case A begin in the second iteration base and they continued until end of the project.

Analysis works started in the first iteration base and ended at the 5th iteration base. Whereas, design works began at the second iteration base and ended in the 6th iteration base. Construction works started also in the second iteration base and ended with the project. In the last two iteration bases, pilot application works are realized. In the last iteration base, the going to live phase was carried out.

Case B was also realized in 7 iteration bases. The first iteration bases needed more effort. Use case diagrams are drawn in the first iteration base. Site inspections are realized in the first and second iteration bases. The business process model is drawn in the second iteration base. In addition, main input output screen is designed in this iteration base. Consolidation of the business process model is needed again in the second iteration base. Therefore, a whole-team meeting was organized.

Analysis works were taken part in all iteration bases. In other words, whenever new requirements are collected, they were added to the product backlog and later completed in the following iteration bases. Design works were handled starting from the second iteration base until to the 6th iteration base. Except the first iteration base, coding works took part in all iteration bases. The going to pilot phase was realized in the last two iteration bases. In the last iteration base, live preparation is realized.

The estimation formula will help to estimate the development effort of the new business processes as if they were developed using the Waterfall methodology. As a result, there are two values of development effort for each new business process. Table 40 shows the calculated efforts for these processes. The effort values are calculated using the estimation formula derived in Chapter 5. For Case A, the calculated effort value is 42.35 days and the calculated effort value of Case B is 53.27 days. These effort values are estimated as if these two cases were implemented using the Waterfall model.

Table 40: Estimated effort values

	# of Steps(s)	# of Complex Steps(s')	Actual Effort	Calculated Effort
CASE A	3	0	34	42.35
CASE B	2	1	41	53.27

7.5. Analysis Phase

The collected data is analyzed to determine whether any meaningful results are emerging. In addition, rival explanations are explored. The data should be displayed apart from interpretations.

The case study is analyzed based on the theoretical propositions. The original objectives and design of the case study are based on the defined propositions. In turn, these propositions reflect a set of research questions, reviews of the literature, and new hypotheses and propositions.

In this case study, pattern matching is used as an analytic technique [62]. In this technique, the empirically obtained pattern is compared with a predicted one. The obtained results help to strengthen the internal validity.

Table 41: Calculated effort of the implementation using the proposed method

Process Name	Actual Effort (person-day)	Calculated Effort(person-day)	Effort Saving Proportion
Case A	34	42.35	0.1972
Case B	41	53.27	0.2303

In Table 41, effort saving is displayed. According to the formula Case A would be implemented in approximately 42 person days using the classical Waterfall methodology. Likewise, Case B would be implemented in approximately 53 person days. However, these processes are implemented in 34 and 41 person days respectively using the proposed methodology. This says that there is an effort saving with the proposed methodology. The proportions of the effort saving are calculated and shown in Table 41. The mean of the proportions is 0.2137. Therefore, approximately 21% effort saving is obtained.

In both cases, more effort was spent in first iteration bases. Small-team meetings were arranged weekly. Use case diagrams were employed in the first iteration bases. Site inspections were realized in the first and second iteration bases. In both cases, a whole-team meeting was needed in the second iteration base to consolidate the process model. In second and third iteration bases, process model and main input output screen were designed.

In both cases, analysis stage took part mostly in the first iteration bases. Design stage took part mostly in the middle iteration bases. Construction stage took part in all iteration bases except the first one. Pilot stage took part in the last two iteration bases. The going to live stage took part in the last iteration base.

Developers and analysts of the cases were selected from a pool. Therefore, individual bias would be eliminated when collecting effort values.

7.6. Validity Threats

There are four widely accepted categories of validity related to a case study: construct validity, internal validity, external validity and reliability [61]. Each validity category will be explained in the following sections.

7.6.1. Construct Validity

Construct validity refers to what extent the design of the study represents the investigation of the research questions. The construct validity is concerned with the relation between the research and the observations.

In this case study, the cases were selected to represent the general business process software development projects. In other words, the cases are not specific to the organization. The cases are general to all organizations. Therefore, the obtained results would be valid for other organizations. The cases were developed by analysts and developers who were trained. They are capable of developing business process software development projects. Therefore, application of the proposed methodology can be applied in other organizations.

Another threat to construct validity is whether the right data is collected and measured. This threat is mitigated by collecting the effort values from different sources. The collected effort values were compared with the estimated effort values. The collected effort values were strengthened by collecting from two sources. And obtained data is refined if there is a mismatch between the two values. The estimation values depend on the estimation formula which has been obtained in the same organization with the same pool of analysts and developers. Therefore, effort comparison is meaningful. The results show an effort saving. This was expected because the agile methodologies save effort usually.

7.6.2. Internal Validity

Internal validity [63] of a case study refers to the degree to which independent variables affect the dependent variables. In industrial case studies, to obtain internal validity is difficult because of not fully achieving control over variables.

The cases were applied to business process software development projects. Similar projects were applied using the Waterfall methodology before. From those projects, an estimation formula is obtained. The cases also implemented in the same organization and in the same conditions except the methodology. Pool of developers and analysts were the same. The customer was new for each project. Therefore, the obtained effort saving depends on the proposed methodology.

7.6.3. External Validity

External validity is related to generalization. In other words, external validity is the degree to which the results are generalizable.

The observations and conclusions presented here are based on two case studies, which can limit the power of generalization. Although these business processes are general and representative, we recognize that the number of the cases is low. However, the results can be generalized arguably. It is hoped that other researchers could add to the evidence in this area by performing additional case studies.

Since the case study was executed in only one organization, it is difficult to generalize the outcome to include other organizations. However, it is possible to conduct such studies in other organizations and reach a more generalized confidence.

Since the selected cases are general to all organizations. The collected and estimated effort values would be nearly the same in other organizations because these values are refined from multiple sources. Moreover, it is hoped that effort saving would occur in other organizations.

The cases have 3 steps in their models. This number is good for small business processes. However, for generalizations there should be more cases with many steps in their models. Moreover, in these cases there is only one complex step. Again, more cases with many complex steps should be implemented to generalize.

The proposed methodology was applied according to its description. This application was simple and for similar cases the application would be generalizable. However, for complicated cases we cannot say it is generalizable. For complicated cases, there would be concurrent iteration bases. In addition, more effort would be needed for consolidations. The proposed methodology should be applied to complicated cases to support its generalization.

7.6.4. Reliability

Reliability means whether the study is conducted in a robust manner and can be repeated by other researchers with the same results. The proposed methodology is defined in detail. Therefore it can be implemented in other organizations for the defined domain. Effort saving would be obtained similar to this case study. To improve reliability the same data was collected from two different sources.

Using a pool of developers and analysts also supports reliability. The duration of the iteration bases is optimized for business process software development projects. Therefore, for the same domain, the iteration bases would be implemented successfully in other organizations. Moreover, the duration and structure of small-team meetings are optimized. This also supports the reliability.

7.7. Case Study Report

In the reporting phase, the audience of the case study is defined. And, enough evidence is displayed for reader to reach own conclusions.

The report of the case study is structured as linear-analytic [62]. In other words, the issue being studied is stated. Then, relevant literature is reviewed and the methods used are explained. Data collection and its analysis are performed to obtain findings. Lastly, the conclusions are made from the findings.

The audience of the case study is the stakeholders of the business process software development projects. Developers, analysts, customers, users, and researchers can benefit from this study.

The proposed methodology was applied to business process software development projects. An effort saving of 21% is obtained. The cases are simple and implemented in an organization. More cases should be implemented to generalize the findings.

CHAPTER 8

CONCLUSION

An agile business process software development methodology was developed and applied in the enactment of two new business processes. The goal was to address requirements better and to develop the business process faster. A 21% reduction in the development effort was achieved. Also the more efficient requirements handling in agile approaches were incorporated.

Agile principles help to adapt to the fast changing environment. In agile methods, requirements are gathered periodically so that they are determined with high quality. The number of versions shows this result. After developments of old business processes, approximately 3 versions were released in the first 6 months. However, for the new business processes only one version was released in the similar period. Therefore, requirements gathered better. Moreover, requirements are kept at minimum because they are prioritized and unimportant ones are postponed or eliminated. That the most valuable requirements with high priority are developed causes fast development.

Adaptation to change requests increased with the new implementations. Since change requests were welcomed in small-team meetings, adaptation of change requests is realized. The training sessions also supported to gather requirements better. This kind of customer involvements resulted in cooperation in software design and change request were handled easier.

Applying classical agile methodologies causes problems too. Although agile methodologies solve the “missing requirements” problem, they do not fit to business process software development projects where the stakeholders have more than one project at the same time.

An agile business process software development methodology is defined for business processes and stakeholders dealing with more than one project concurrently. It is iterative and incremental. Customer involvement is realized and new requirements are gathered at short development cycles called iteration bases. The project task list is prioritized frequently to queue up the most valuable requirements for iteration bases. While spike solutions increase the communication between the stakeholders, small teams ensure frequent communications through periodic meetings.

In this agile business process software development methodology, sponsorship, lightweight project management, use case diagrams, site inspection, role modeling, BPMN BPD diagrams, using templates, and main input output screen are encouraged. This specialized methodology gives great emphasis on training, keeping history, going to pilot stage, and determining the roles. Especially training is important because it creates awareness about business processes so that it supports the success of the development.

A formula is derived for estimating the effort required in the development of a business process based on the traditional approaches. The actual effort used in the development of a new business process is compared to the estimated effort if it was developed using the traditional approach. The latter value was obtained using the developed formula.

The proposed method showed an effort saving. According to this study, 21% effort saving is obtained. The development environments for the old and new projects were the same except the methodology. Therefore, this saving points out that the proposed methodology achieved the saving. Therefore, the conclusion validity [63] is provided. Implementation of new business processes using the proposed methodology are expected to support this improved efficiency. Hence, the external validity [63] would be satisfied.

As future work, the methodology can be applied at different organizations and further statistical analysis can be conducted for more detailed evaluation of the success of the methodology.

REFERENCES

[1] W.M.P. van der Aalst, A.H.M. ter Hofstede, M. Weske. Business Process Management: A Survey. BPM 2003, LNCS 2678, pp.1-12, 2003. Springer-Verlag Berlin Heidelberg 2003.

[2] C. Moller, C.J. Maack, R.D. Tan. What is Business Process Management: A Two Stage Literature Review of an Emerging Field. IFIP, Volume 254, pp.19-31.

[3] M. Weske. Business Process Management Concepts, Languages, Architectures. 10.1007/978-3-540-73522-9_8

[4] M. Owen, J. Raj. BPMN and Business Process Management, Introduction to the New Business Process Modeling Standard.

[5] W.M.P. van der Aalst. Three Good Reasons for Using a Petri-net-based Workflow Management System. Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC'96), pp.179-201, Cambridge, Massachusetts, Nov 1996.

[6] W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, P. Wohed. Pattern-based Analysis of BPML (and WSCI). QUT Technical report, FIT-TR-2002-05, Queensland University of Technology, Brisbane, 2002.

[7] C.A. Ellis, G. Nutt. Workflow: The Process Spectrum. Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems, pp.140-145, Athens, Georgia, May 1996.

[8] M. Weske, T. Goesmann, R. Holten, R. Striemer, Analysing, Modelling and Improving Workflow Application Development Processes. Software Process: Improvement and Practice 6(1):35-46, 2001

[9] R.K.L. Ko, A Computer Scientist's Introductory Guide to Business Process Management (BPM), www.acm.org/crossroads, Vol. 15, No.4, Summer 2009

[10] M. Havey, Essential Business Process Modeling, O'Reilly August 2005, ISBN 0-596-00843-0

[11] R.Yamamoto, K.Yamamoto, K.Ohashi, J.Inomata, Development of a Business Process Modeling Methodology and a Tool for Sharing Business Processes, Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC'05), 0-7695-2465-6/05, IEEE, 2005

[12] R.K.L.Ko, S.S.G.Lee, E.W.Lee, Business process management (BPM) standards: A survey. Bus. Process Manage. J. 15, 5. To appear, 2009.

[13] M.Kirikova, J.Makna, Renaissance of Business Process Modelling, Information Systems Development: Advances in Theory, Practice and Education Edited by O. Vasilecas et al., Springer, 2005

[14] K.Schwaber, Agile Project Management with Scrum, Washington, Microsoft Press, 2004

[15] Beeri, C., Eyal, A., Milo, T. & Pilberg, A. (2007) "Monitoring Business Processes with Queries". Proceedings of the 33rd International Conference on Very Large Data Bases 2007 (VLDB 2007). Vienna, Austria, pp. 603-6 14.

[16] H.Smith, Business process management—the third wave: business process modelling language (bpml) and its pi-calculus foundations, Information and Software Technology, Volume 45, Issue 15, 1 December 2003, Pages 1065-1069

[17] Koubarakis, M. & Plexousakis, D. (1999) "Business process modelling and design-a formal model and methodology". BT Technology Journal, Vol. 17 No. 4, pp. 23-35.

[18] M. P. Papazoglou, W. van den Heuvel, Business process development life cycle methodology, October 2007, Communications of the ACM, Volume 50 Issue 10

[19] Nirmit Desai , Ashok U. Mallya , Amit K. Chopra , Munindar P. Singh, Processes = Protocols + Policies: A methodology for business process development (2004),Proceedings of the 14th International World Wide Web Conference (WWW'2005)

[20] Barry Povey, The development of a best practice business process improvement methodology, Benchmarking: An International Journal, 1998, Volume: 5, Issue: 1, Page: 27 – 44

[21] R. Pressman, Software Engineering a Practitioner's Approach, 6th ed. New York, New York, McGraw Hill, 2005.

[22] W. M. P van der Aalst, Business Process Management: A Personal View, Business Process Management Journal, Vol. 10 No. 2 pp. 5, 2004.

[23] C. Beeri, A. Eyal, S. Kamenkovich, T. Milo, Querying Business Processes with BP-QL, Proceedings o f the 31st International Conference on Very Large Data Bases. pp. 1255-1258, 2005.

[24] P. Vassiliadis, T. Sellis, A Survey of Logical Models for OLAP Databases, SIGMOD Record 28(4), Dec. 1999.

[25] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, CA, 2006.

[26] C. Møller, C.J. Maack, R.D. Tan, What is Business Process Management: a Two Stage Literature Review of an Emerging Field, IFIP International Federation for Information Processing, Volume 254, Research and Practical Issues of Enterprise Information Systems II Volume, 1, pp19-31, 2007.

[27] A.R. Silva et al, AGILIPO: Embedding Social Software Features into Business Process Tools, BPM 2009 Workshops, vol. 43, 2010; 219–230.

[28] G. Bruno et al, Key Challenges for Enabling Agile BPM with Social Software, Journal of Software Maintenance and Evolution: Research and Practice, 23:297-326, 2011.

[29] S. Larson, Applying Agile Software Development Methodologies to Business Process Redesign/Management (BPRM), Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA March 26th-27th, 2010.

[30] Agile Manifesto, accessed 12 January 2012 at http://agilemanifesto.org/principles.html.

[31] L. Lindstrom, R. Jeffries, Extreme Programming and Agile Software Development Methodologies, Information Systems Management, summer, 41-52, 2004.

[32] Extreme Programming, accessed 18 January 2012 at http://www.extremeprogramming.org/.

[33] A. Schatten, J. Schiefer, Agile Business Process Management with Sense and Respond, IEEE International Conference on e-Business Engineering (ICEBE'07), pp. 319-322, 2007.

[34] K. Molokken, and M. Jorgensen, "A Review of Surveys on Software Effort Estimation", in International Symposium on Empirical Software Engineering, 2003, pp. 223-231.

[35] M. Jørgensen, “A Review of Studies on Expert Estimation of Software Development Effort,” J. Systems and Software, vol. 70, pp. 37-60, 2004.

[36] S. Conte, H. Dunsmore, and V.Y. Shen, Software Engineering Metrics and Models. Menlo Park, Calif.: Benjamin Cummings, 1986.

[37] Shepperd, M. and Kadoda, G., “Comparing software prediction techniques using simulation”, Software Engineering, IEEE Transactions on, Volume: 27 Issue: 11, Nov. 2001, Page(s): 1014 -1022.

[38] B. Lo and X. Gao, Assessing Software Cost Estimation models: criteria for accuracy, consistency and regression, The Australian Journal of Information Systems, v.5:1, September 1997, p: 1-27.

[39] Kitchenham, B. MacDonell, S. Pickard, L. and Shepperd, M. “What accuracy statistics really measure” IEEE Proceedings - Software Engineering, 48, pp.81-85, 2001.

[40] S. Grimstad, M. Jørgensen, A framework for the analysis of software cost estimation accuracy, in: The International Symposium on Empirical Software

Engineering (ISESE), Rio de Janeiro, Brazil, September 21–22, ACM Press, 2006, pp. 58–65.

[41] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrtveit, “A Simulation Study of the Model Evaluation Criterion MMRE” IEEE Trans. Software Eng., vol. 29, no. 11, pp. 985-995, Nov. 2003.

[42] Lederer, A.L. and J. Prasad, Information systems software cost estimating: a current assessment. Journal of Information Technology, 1993(8): p.22-33.

[43] A. Arkin et al. Business Process Modeling Language (BPML), Version 1.0, 2002.

[44] P. Kruchten, A Rational Development Process, Crosstalk, 9 (7) July 1996, pp.11-16.

[45] A.I. Khan, R.J. Qurashi, U.A. Khan, A Comprehensive Study of Commonly Practiced Heavy and Light Weight Software Methodologies, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 2, ISSN (Online): 1694-0814, www.IJCSI.org.,July 2011.

[46] S. Brinkkemper, Method Engineering: Engineering of Information Systems Development Methods and Tools. Information and Software Technology, 38 (4). pp. 275-280. ISSN 0950-5849, 1996.

[47] B. Henderson-Sellers, J. Ralyté: Situational Method Engineering: State-of-the-Art Review. Journal of Universal Computer Science, 16(3): 424-478 (2010).

[48] K. Kumar, R.J. Welke: “Methodology Engineering: A Proposal for Situation Specific Methodology Construction,” Challenges and Strategies for Research in Systems Development, Cotterman, W. and J. Senn (eds.), J. Wiley, Chichester, UK, 1992, pp. 257-266.

[49] A.F. Harmsen, Situational Method Engineering, Doctoral dissertation University of Twente, 1997.

[50] B. Henderson-Sellers. Method engineering: Theory and practice. In D. Karagiannis and editors Mayr, H. C., editors, Information Systems Technology and its Applications., pages 13–23, 2006.

[51] C. Rolland, A Primer For Method Engineering. Proceedings of the INFORSID Conference. 1997

[52] F. Harmsen, S. Brinkkemper, J. L. Han Oei, Situational method engineering for informational system project approaches. Methods and Associated Tools for the Information Systems Life Cycle, 1994: 169-194.

[53] B. Nuseibeh, A. Finkelstein, J. Kramer, Method Engineering for Multi-Perspective Software Development, Information and Software Technology Journal, 38(4): 267-274, Elsevier Science B.V., April 1996.

[54] T. Punter, K. Lemmen. The MEMA-model: towards a new approach for Method Engineering. Information and Software Technology 38 (1996) 295-305.

[55] V. Rahimian, R. Ramsin. Designing an agile methodology for mobile software development: A hybrid method engineering approach. In O. Pastor, A. Flory & J.-L. Cavarero (eds.), RCIS (p./pp. 337-342), : IEEE. ISBN: 978-1-4244-1677-6, 2008.

[56] E. Domínguez, M.A. Zapata. Noesis: Towards a situational method engineering technique. Inf. Syst., 32, 181-222, 2007.

[57] A.H.M ter Hofstede, T.F. Verhoef. On the Feasibility of Situational Method Engineering. Inf. Syst., 22, 401-422, 1997.

[58] M. Saeki. A meta-model for method integration. Information & Software Technology, 39, 925-932, 1998.

[59] J. Becker, C. Janiesch, D. Pfeiffer. Reuse Mechanisms in Situational Method Engineering. In J. Ralyté, S. Brinkkemper & B. Henderson-Sellers (eds.), Situational Method Engineering (p./pp. 79-93), : Springer. ISBN: 978-0-387-73946-5, 2007.

[60] S. Tasharofi, R. Ramsin. Process Patterns for Agile Methodologies. In J. Ralyté, S. Brinkkemper & B. Henderson-Sellers (eds.), Situational Method Engineering (p./pp. 222-237), : Springer. ISBN: 978-0-387-73946-5, 2007.

[61] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in soft-ware engineering. Empirical Software Engineering, 14(2):131–164, 2009.

[62] R.K. Yin, Case study research. Design and methods, 4th edn. London, Sage, 2009.

[63] C. Wohlin, M. Höst, K. Henningsson, Empirical research methods in software engineering. In: Conradi R, Wang AI (eds) Empirical Methods and Studies in Software Engineering—Experiences from ESERNET, Springer, 2003.

[64] H.K. Wright, M. Kim, D.E. Perry, Validity concerns in software engineering research. In G.-C. Roman & K. J. Sullivan (eds.), FoSER (p./pp. 411-414), : ACM. ISBN: 978-1-4503-0427-6, 2010.

[65] M. Höst, P. Runeson, Checklists for Software Engineering Case Study Research. ESEM (p./pp. 479-481): IEEE Computer Society, 2007.

[66] R. Feldt, A. Magizinius, Validity Threats in Empirical Software Engineering Research - An Initial Survey. Proc. of the Int'l Conf. on Software Engineering and Knowledge Engineering (p./pp. 374-379), 2010.

[67] G. Zheng, Implementing a business process management system applying Agile development methodology: A real-world case study, thesis, May 2012.

[68] D. Çulha, A. Doğru. Towards an Agile Methodology for Business Process Development, S-BPM ONE, LNBIP 170, pp. 133–142, 2014.

[69] John J. Marciniak (ed.), Encyclopedia of Software Engineering. Taylor & Francis. ISBN:0-471-54004-8, 1994

[70] J.A.McCall, P.K.Richards, G.F.Walters, Concepts and definitions of software quality. Factors in Software Quality NTIS, 1, 1977.

[71] P. Berander et al, Software Quality Attributes and trade-offs, Blekinge Institute of Technology, June, 2005.

[72] R. Fitzpatrick, Software Quality: Definitions and Strategic Issues. School of Computing Report, Advanced Research Module, Staffordshire University. April 1996.

[73] M. Papazoglou, W.J. van den Heuvel, Service Oriented Architectures: Approaches, Technologies and Research Issues. The VLDB Journal, 16, 389--415, 2007.

[74] R. Seethamraju, J. Seethamraju, Enterprise systems and Business Process Agility - A Case Study, Proceedings of the 42nd Hawaii International Conference on System Sciences – 2009.

[75] M.E. Kharbili, T. Keil, Bringing Agility to Business Process Management: Rules Deployment in an SOA, Whitestein Series in Software Agent Technologies and Autonomic Computing, 157–170, 2009 Birkhauser Verlag Basel/Switzerland

[76] Dan, F. Gittler, and F. Toumani (Eds.): ICSOC/ServiceWave 2009, LNCS 6275, pp. 370–384, 2010. © Springer-Verlag Berlin Heidelberg 2010.

CURRICULUM VITAE

PERSONAL DETAILS

Name surname: Davut Çulha

Profession: Computer Engineer

Sex: Male

Nationality: Turkish (TC)

Date and place of birth: 19.07.1973, Ankara

Marital status: Married with two children

Phone: +90 312 592 30 42

e-mail: culha@aselsan.com.tr

Military service: Completed

EDUCATION

Ph.D. Computer Engineering, Middle East Technical University, Ankara, 2006-2014, (CGPA: 3.79 out of 4), An Agile Business Process Software Development Methodology

M.Sc. Computer Engineering, Middle East Technical University, Ankara, 1995-1998 (CGPA: 3.88 out of 4), Two Matching Algorithms for Aerial Images

B.Sc. Computer Engineering, Middle East Technical University, Ankara, 1990-1995 (CGPA: 3.50 out of 4)

High School, Ankara Lisesi, Ankara, 1987-1990

PUBLICATION

D. Çulha, A. Doğru. Towards an Agile Methodology for Business Process Development, S-BPM ONE 2014, LNBIP 170, pp. 133–142, 2014

PROFESSIONAL EXPERIENCE

2003 November – Present; Aselsan, Ankara; Team Leader

Took part in the installation of SAP ERP as the leader of the development team. Nearly all SAP Modules are implemented during the installation. Attained deep knowledge on SAP ABAP, SAP Basis.

Took part in the implementation of Aselsan Workflow System. As a team leader of the developers, currently 8 workflows have been implemented using the Software AG BPM system webMethods. webMethods relies on Java. Gained deep Java experience.

Led the upgrade project of the webMethods business process management system.

Developed an SAP SQL Server adapter using Java. Java threading, SAP Java connector JCO, and SQL JDBC are used. With the adapter, SQL statements are executed in SAP.

Developed an SAP webMethods adapter using Java. With the adapter, webMethods IS services are called in SAP.

Developed a webMethods SAP adapter using Java. With the adapter, SAP RFC function modules are called in webMethods Java services.

First usage of SAHAB (Aselsan’s Virtual Air Gap product) is realized using .Net c#. SAHAB connects different security level networks. Deep experience is gained on web services.

Developed a custom mail server using c#. This mail server is used to send messages to other Networks through SAHAB.

Developed a file transfer system using #c web services. The system transfers and logs files to other Networks. Gained deep experience on c# threading, web services, and Microsoft IIS.

Took part in ALM (Application Lifecycle Management) studies using Microsoft TFS.

October 2000 – 2003 November; CyberSoft, Istanbul; Team Leader

Developed an IDE (Integrated Development Environment) in Java for the AURORA Project that is a framework for rapid application development to be used for enterprise applications. The IDE named as EDS is a multilingual screen design and test environment based on JavaBeans technology. Developed without significant disruption to programmers developing on EDS. EDS 1.0 delivered 2 months ahead of schedule. Experienced in Java, AWT/Swing, JavaBeans, XML and Multi-threaded programming. Used Eclipse.

Finansbank and our company developed a Core Banking Project. The project was a web-based, 4-tier and scalable project. There were six software development teams in the project. I was the team leader of the Foreign Trade Group, giving directions to 9 programmers. Involved in analysis, design and development phases of the project. Used object oriented approaches in all phases of the project. Software development time was reduced by 25% using the framework AURORA.

Experienced in IBM Visual Age for Java, Oracle DBMS, Rational Rose, UML, Microsoft Visual Source Safe, Eclipse, CVS, Erwin, Microsoft Project.

Team Leader of 5 programmers developing Credits Module of the Core Banking Project. Re-engineered the old credits module. Used Erwin.

May 2000 – October 2000; Opus Bilgi Sistemleri, Ankara; Computer Engineer

Took role in developing Medical Information System Project. Used Visual Basic and Oracle.

November 1998 - March 2000; General Staff, Health Division, Ankara; OBİ Officer

Studied on medical information systems. Prepared Specifications of Medical Information System of the Turkish Armed Forces.

December 1997-September 1998; NETAŞ (NORTEL), Research and Development Department, Ankara; Computer Engineer

Took role in the Project RMS-5T which is a switch management project. Developed some scheduling and encryption modules using Visual Basic and Access on Windows NT domain.

November 1996-November 1997; SFS Stratejik Fonksiyonel Sistemler, Ankara; Analyst Manager

Developed Basic Health Statistics Module (BHSM) for Ministry of Health (MOH). The aim of this project is to collect and evaluate health statistics data in provinces, and to compute required statistics, and to report and transfer consolidated data to MOH periodically.

Used Visual Basic and Oracle

August 1995 - October 1996; Department of Computer Engineering, Middle East Technical University, Ankara; Research Assistant

Performed researches and implementation using Neural Network Approaches on Computational Vision to find the depth information of stereo images. The objective of this project is to simulate human visual system on robotics.

Developed two matching algorithms for aerial images (M.Sc.Thesis) using C and C++ on Linux.

Studied on a query optimizer generator (Cascades) source code in order to modify, and enhance its capabilities. The source code is in C++ language on Sun OS (UNIX) environment.

Developed Company Querying System for KOSGEB using Visual Basic and Access. This system allows to the users to design their own queries and generate user-defined reports. The developed software has two different modes; advanced and simple user interfaces. The software was also multi-lingual (Turkish, English, French, German, and Italian).

1994 July – August (Summer Practice); NACE Makina Sanayi A.Ş., Ankara

Studied on Novell Netware and Real World Processing (Genesis).

1993 July - August (Summer Practice); M.E.B. Bilgisayar Eğitimi ve Hizmetleri Genel Md., Ankara

Developed Personnel Information System using Dbase.