العنوان:	Software Configuration Management for Embedded and Real Time Software in Software Engineering Domain
المؤلف الرئيسي:	الزواهرة، حنان حمدان أحمد
مؤلفين آخرين:	الصرايرة، خالد تركي، المخادمة، خالد عبدالكريم(مشرف)
التاريخ الميلادي:	2017
موقع:	الزرقاء
الصفحات:	1 - 52
رقم MD:	856632
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	عمادة البحث العلمي والدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات، برمجيات الزمن الحقيقي، إدارة تكوين البرمجيات، برمجيات الأنظمة المضمنة
رابط:	https://search.mandumah.com/Record/856632

CHAPTER ONE
INTRODUCTION

Currently, software products intended to grow exponentially in market. Thus, the complexity of software has a huge increment. This makes a big challenge for the software development. That leads the strongest companies to force the software engineers to develop their product in minimal time with limited size and high quality while maintaining the same customer's requirements. Therefore, we must impose more control on software product development to avoid failure.

The normal Software Configuration Management (SCM) considered as service function in software development process (SDLC). Where the SDLC is a continuous process of change, due to many reasons, changes in requirements, schedule, budget, design programs and others [1]. SCM is interested in managing the evolution of software systems during the initial stages and during all other phases in SDLC. That means when made modify in embedded software must be efficiently control. In order to ensure that the modifications do not turn the process into chaos. Therefore, the modifications controlled by a software engineering activity called SCM [2].

This introductory chapter presents an overview of the problem statement of this thesis. It discusses the importance of introducing more research effort in order to improve framework to develop the embedded and real time software using SCM .Moreover it also discusses the objectives that achieved throughout this thesis. Finally, this chapter concludes with the chapters' organization throughout this thesis.

1.1 Problem Statement

Software configuration management (SCM) is a special case of configuration management (CM). Where software is faster and easier to change more than hardware. SCM can deal with some problems that related with the evolution of software and lack of control. So when developing the embedded system we face some challenges. Risks from malfunctions of embedded software are much higher than the application software.

There are many frameworks deal with evolution of embedded software. However, these frameworks have some weakness such as their development cycle is not clear enough, other types used in specific domain and they developed based on using other tools and frameworks to support them. Our framework built to overcome this weakness.

Based on the above, this research studies proposed a new and generic steps of SCMF to develop the embedded and real time software's and their role in increasing the efficiency, reliability, and traceability of software product in industrial sector.

1.2 Importance of Thesis

Many organizations deal with complex embedded and real time software. But it face some problems when deciding to improve these types of software, that return to lack of controlling the changes which introduce on the ES through development phase, where changing in ES without controlling and tracking can lead into chaos and losses in software product. Thus, this thesis is to contribute the improvement of SCM .Regarding developing the embedded and real time software to insure the process goes correctly, by the adoption of the SCMF to identify the configuration items and components. Then place them under configuration management in order to control the changes.

1.3 Thesis Objectives

The study aims to achieve the following objectives:

1. Investigate the Software Configuration Management Frameworks of embedded software from previous work proposed in literature.
2. Proposed a generic steps on the basis of the previous and explored frameworks
3. Manage the change and development of embedded and real time software system during software life cycle.

1.4 Thesis Organization

This thesis consists of seven chapters. The first chapter discusses the current problem statement, which is the lack of control changes in embedded software when developing it. **Chapter 2** presents the main motivation and the objectives of this thesis; it also presents a detailed methodology that followed to carry out the objectives of this thesis. Furthermore, **Chapter** 3 shows an overview of the previous work of

SCM in embedded and real time software. **Chapter** 4 shows the requirements of business domain of embedded and real time software. **Chapter 5** relies on Chapters 3 and 4, where this chapter proposes a software configuration management SCMF and its elements to support the development process for the new or enhanced products of the embedded and real time software. **Chapter 6** introduces a case study using the proposed SCMF in order to validate the controlling and tracking changes through developing the embedded and real time software. Finally, a brief conclusion and future work pointed out in **Chapter 7**.

العنوان:	Software Configuration Management for Embedded and Real Time Software in Software Engineering Domain
المؤلف الرئيسي:	الزواهرة، حنان حمدان أحمد
مؤلفين آخرين:	الصرايرة، خالد تركي، المخادمة، خالد عبدالكريم(مشرف)
التاريخ الميلادي:	2017
موقع:	الزرقاء
الصفحات:	1 - 52
رقم MD:	856632
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	عمادة البحث العلمي والدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات، برمجيات الزمن الحقيقي، إدارة تكوين البرمجيات، برمجيات الأنظمة المضمنة
رابط:	https://search.mandumah.com/Record/856632

CHAPTER TWO
RESEARCH METHODOLOGY

This chapter aims to discuss the main motivation behind conducting this research work and discusses the detailed research methodology that followed to scope the thesis objectives as well as organizes the research process in order to ensure validity of this thesis. It discusses the research motivation, goals, objectives and a detailed methodology to achieve this research. Moreover, this research methodology contains four phases as follows to extend the research objectives.

2.1 Overview of Research Methodology

This part describes an overview of the research methodology that used to achieve objectives. Moreover, it provides clear steps to demonstrate how the thesis carried out, and it consists of four phases as seen in Figure 1.

Phase1: SCM and SCMF in the literature Review.

This phase includes surveying the literature on the previous and published frameworks of SCM from academia and the industry.as well as it presents an overview of the earlier and recent studies in this domain. For more details. See chapter 3.

Phase2: Analyzing the Requirements of ES in Business Domain and Explain the SCRUM Process

This phase analyzed the identified steps that guided to build the generic SCMF and their elements. It divided into two parts:

1) Defines and analyze the requirements of embedded software in business domain.
2) Shows how the software developed and changed in the organization requirements by using the SCRUM process .In addition this chapter includes the SCRUM principle and activities. See chapter 4.

Phase 3: Software Configuration Management Framework (SCMF)

Is a construction phase. After analyzing the SCM and ES requirements in the previous phases, then propose SCMF based on previous frameworks to support the development process of embedded and real time software through specific steps. See chapter 5.

Phase 4: Case Study

Applying Software Configuration Management Framework SCMF to validate this solution by the case study in RICE COOKER. (See chapter 6).

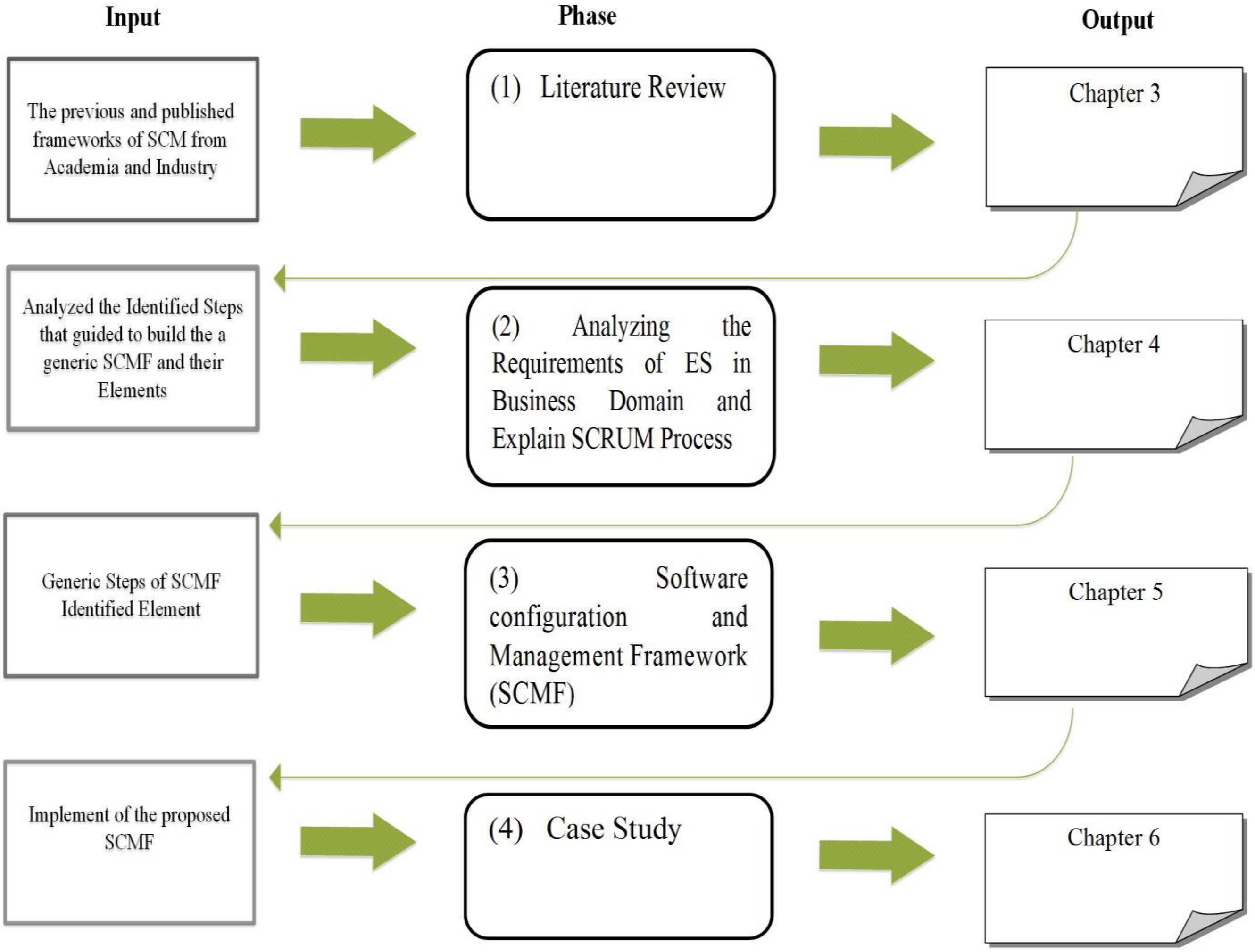
Input
Phase
Output
The previous and published frameworks of SCM from Academia and Industry
(1) Literature Review
Chapter 3
Analyzed the Identified Steps that guided to build the a generic SCMF and their Elements
(2) Analyzing the Requirements of ES in Business Domain and Explain SCRUM Process
Chapter 4
Generic Steps of SCMF Identified Element
(3) Software configuration and Management Framework (SCMF)
Chapter 5
Implement of the proposed SCMF
(4) Case Study
Chapter 6

Figure 1: Research Methodology

CHAPTER THREE

LITERATURE REVIEW

Currently, software products are becoming more complex and larger than before. In the same time during the development phase, we make some changes to develop it. In addition, any change that occurred in the environment causes change in the software requirements. Moreover, customers demand a product with high quality in minimal time. Thus, software changes are unavoidable in software development products. Where each change occurred during the development phase can lead to a better quality, or may lead to failure if the software of product development did not put under good control.

Software development process aims to obtain good embedded software with high level of quality. However, the main issues and challenges that face the developers are how they can control these changes during software lifecycle. Therefore, to ensure these changes do not return to the chaos, we can use configuration management (CM). The essential purpose of software engineering is to obtain the highest level of the improvement in an easy way to introduce new changes to software.

CM is an essential task to develop the complex embedded software. It is a management task. Moreover, it provides administrative and technical guidelines for the lifecycle of embedded software.CM needs some systems to support its process such SCM. SCM is a special case of CM [3]. In addition, SCM provides a way for identifying, tracking, and controlling the version of each item in embedded software.

SCM considered as a method for controlling the changes of embedded and real-time software during the life cycle. The essential activity in SCM is version control and change management. Change management defined as "the process of continually renewing an organization’s direction, structure, and capabilities to serve the ever changing needs of external and internal customers" [4]. Version control is defined as" the assignment of either unique version names or unique version numbers to unique states of software configuration items, usually for a specific purpose, such as a release of the software product to an external group or the identification of a specific baseline"[5].

Embedded systems developers face some problems when dealing with different versions of software. For this reason, the developers tend to use the SCM to manage large files.

Embedded software changed through the development process. Referring to many reasons one of them return to the stakeholder, He/she demands updates of data, service delivered, or functionality. Sometimes the scheduling constraints or budget reasons redefine the system. In addition, the business reorganization causes changes in team formation. Furthermore, the market conditions or new businesses impose changing the requirements or rules of the business [6]. In software, development process has three main classifications applied on the object: controlled, pre-controlled and uncontrolled [7].

Controlled object its configuration control. Software configuration identification (SCI) determines items to be controlled. Because it is impossible to control anything, which is not, known. That means breaking the system down into a number of parts. Then, make relations among the SCI to facilitate the traceability of these relations. After that grouping similar configuration items into the baseline. It, which separated between different versions. Then, understand the status of each item through the development process [7].

The electronic control units (ECU) in automotive industry have integration between hardware and software. ECU with Driver Assistance systems (DAS) can provide control and safety performance. However, the main problem at ECU was in HW/SW. Which cannot make any change at the run time. This contrasts with coming days; in future, the DAS distributed over the all automobile. Thus, the automobile needs to interact the components with each other's .in order to obtain high level of performance, provide ability to change the hardware and software configuration, and allow the incorporation among components at the run time. To get all of these benefits the interface layer for ECU such as radar and video, was designed [8].

Real-time system identification tool (RealSysId). Considered a computational tool. It provides the ability to select and compute through a real time of flight. The main purpose of this tool is developing the selected coefficients online then identifying it and validating all these online. It was the first version created in the aircraft domain.

Moreover, this tool has the ability to select and compute through real time with analytical and visual indications by using some methods. While the most of aircraft systems define all procedures by using pre-planned flight tests offline [9].Recently the cyber-physical system (CPS) is an embedded system. It used to monitor the physical processes .CPS used for multiple tasks in various environments where different constraints and rules need to maintain the system and it often leads to develop several of products variant. It can increase the efficiency and the effectiveness of the development of variability by using the product line engineering (PLE) [10].

PLE tools and methods provide the ability to reuse the existing software. Evolution management is the critical thing for CPS. It achieved by software configuration management (SCM). It focuses on changing control to keep traceability and integrity. So, it was needed the version control. Thus, the combinations between SCM with PLE increase the level of controlling and managing the software evolution. The main point behind this combination is version control to define what and where the change was occurred [10].Ship Power System is a full and independent system. In addition, it demands a high level of quality, especially run time to develop the power quality of ship system using the combination of techniques as configuration software and virtual instrument technology [11].

The electric vehicle spreads in the global market. However, it still faces some challenges. The big challenge is the short of autonomy, where, this vehicle can deal with limited hundred kilometers. The embedded energy software in these vehicles put it under monitoring. Therefore, it used Energy Management System (EMS) [5].EMS has a high level of controlling with a high quality of managing. Therefore, each hybrid and full vehicles use EMS in order to provide the best vehicle efficiency. Other researchers tend to focus on a prediction or the trip knowledge to reach the optimal battery and engine. However, the lasting research interested in embedded software to obtain global Quality of Service through used some Quality Assurance (QA) frameworks [5].

QA framework provides two different ways effectively configure electric vehicle of EMS: on-line and off-line, where, the off-line uses to define the characteristics of the vehicle and then matching it with their abilities. The on-line is the execution of

QA framework. It used to define which driving strategy to choose the best from alternatives. The two ways lead to minimize the computation time that needed to choose the optimal solution from space on-line. QA framework provides Quality of Service as much as possible in addition provide EMS, which has global, features [5].

Industrial embedded systems have long lifetime also demand a high level of reliability and safety requirements, where HW/SW exposed to several changes. So that, it must control these changes over the lifecycle of the system. Embedded software is a critical part; it plays the main role to distinguish a product from another. The dilemma of industrial embedded systems is how to manage and control the changes. Several tools and models used to manage the development process for Programmable Control and Communication Platform (PCCP), which built through Linux. In addition, SCM used in this kind of machine to identify and control software configurations. It provides many benefits to a product such the ability to trace changes, increasing productivity, and more safety [12].

Configuration Management (CM) is a combination of techniques, which coordinate and control with each other to construct the system. Which was developing the CM principles to enable the hardware engineers design and assemble the components [13]? Many authors explain the SCM through many of definitions. The first definition is "SCM, like CM, is defined as the discipline of identifying the configuration of a system at discrete points in time for purposes of systematically controlling changes to this configuration and maintaining the integrity and traceability of this configuration throughout the system life cycle"[14].

According to IEEE standards, the SCM Is " Configuration Management is a discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements"[15]. SCM is a development process with some of standards and procedures to provide a good managing of an evolving the software system. In addition, SCM considers as a Software quality assurance activity that applied during the software process [16]. Thus, SCM is a collection of activities integrated with each other's to manage and control every change throughout the life cycle of computer software. SCM is an integral part of software quality assurance SQA. It participates to assess the impact of the changes that occurs during SQA activities and defined decisions based on cost and benefit analysis. In addition, SCM is a helping software life cycle process. It used mostly in the development and maintenance of the organization [17] [18].

Customers need to change their requirements at any phases in the life cycle. So these changes addressed, for this reason, organization tend to use SCM to ensure these changes approved or disapproved. The SCM activities divided into two main parts: planning and controlling of the SCM process. Traditionally these activities grouped as series of tasks. These tasks have many functions, software configuration identification (SCI), configuration control (CC), configuration status accounting (CSA), and configuration auditing (CA) [19].

The good software configuration management plan (SCMP) for controlling changes can lead to avoid the chaos and unexpected results. So, to success, SCM must analyze, manage and plan all requirements correctly. Otherwise, the software project fails. SCM is a critical element in software engineering because it must keep the high quality of product and continue to improve the productivity of software through the lifecycle, which achieved through organized and planned change management. Moreover, it can include the change management in SCM .In order to obtain high stability, reliability, better quality of time and provide the way to detect the bug and tracking it. In addition, SCM known as an inseparable part of quality oriented [20] [21].

Risk management (RM) is imperative to success software project. Because it was used to catch the modification and the roles, responsibility, controlling, and tracking. It defines each risk and classify it in risk log. Therefore, we use SCM to manage all potential risks. The first step in SCM Repository is to save all risks in the risk log then stored it in a specific way to determine which can occur before the other. In addition, it provides the ability to track for each risk by new members; this applied in the spiral model. Each risk recorded in a repository. The previous data used as input to the next step while File versioning considered as a major risk under SCM [22].

Software product line engineering (SPLE) is a technique for developing all types of software products in the specific domain. SPLE does not have SCM tool to improve their evolution; SCM tool designed to improve a single product. Therefore, the researcher suggests a new prototype of configuration management called Molhado SPL. It designed to solve the evolution process in SPL [23]. When developing a software it may occur many changes. These changes put into controlling and tracking in order to achieve a high level of quality and improvement of software productivity. Therefore, they use SCM in order to obtain global tractability for configuration management and individual artifact's changes through the combination of individual working environment and configuration management system. Thus, traceability can be control more effectively [24].

The essential feature in SCM is a version control (VC). It is a stand- alone but in most cases, it embedded in different types of software.VC stores versions based on increasing updates rather than entire copies. In addition, it has the ability to retrieve the previous version during debugging and testing phase. In addition, it provides tracking the document to correct the error by editors. All versions stored in a repository. While repository must have the ability to manage, all object types such text, bitmaps, and complicated documents. All of these is to enable effective controlling and management of software product [25].

VC is divided into two types based on the environment are Distributed version control systems (DVCS) and centralized version control systems (CVCS) such as current version control (CVS) and subversion (SVN). However, DVCS have a more popularity than CVCS [25]. Furthermore, VCS an essential role in the tracking system and releases management. VCS also known as revision control. It provided historical views of source code and all changes on it; so VCS named a source control. There are various kinds of VCS such as git, mercurial, bazaar, CVS and SVN, these types aim to develop the merging and branching capabilities and provide more secure to repository [26].

SCM is a discipline of software engineering where SCM consists of procedures and technical. It used to manage the changing of software product. In addition, it considered an umbrella activity because it applied through the software development lifecycle. In addition, the modification can occur in any phase. Every system in software engineering domain contains many components. These components integrated with each other to success specific goals of the system. Thus, SCM is a system contains many items; it used in some organizations to identify each element then tracking it to extract any changes by change control. In addition, it can verify any modification by using some tools that used in versioning .in order to follow bugs and maintain the right changes.

العنوان:	Software Configuration Management for Embedded and Real Time Software in Software Engineering Domain
المؤلف الرئيسي:	الزواهرة، حنان حمدان أحمد
مؤلفين آخرين:	الصرايرة، خالد تركي، المخادمة، خالد عبدالكريم(مشرف)
التاريخ الميلادي:	2017
موقع:	الزرقاء
الصفحات:	1 - 52
رقم MD:	856632
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	عمادة البحث العلمي والدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات، برمجيات الزمن الحقيقي، إدارة تكوين البرمجيات، برمجيات الأنظمة المضمنة
رابط:	https://search.mandumah.com/Record/856632

CHAPTER FOUR

Requirements Business Domain of Embedded and Real Time Software

Embedded software (ES) products spread all over the world. It was strongly connect with physical environment. Moreover, ES merge in people's life and use it in daily job such as mobile phones, airplanes, satellites, washing machines and cars [27]. On the other hand, many organizations increase the search of the way to develop the embedded software with better quality and deliver faster .But it is still facing some of challenges when developing it due to distinct features, which distinguish it from other software's [28].

The business environment means anything that surrounding the business corporations and affect by their performance and decisions. The business requirements express about the objectives of organizations or users requesting for developing software. Moreover, business domain can help the software architect to understand the business requirements and business goals. Also, provide an easy way to decide which architecture pattern may be appropriate with the specific needs of the corporation. The managers in the corporation must interact with environmental factors external or internal. Corporation's success and survival relies on the adaption of changing requirements [29].

Companies, which are looking for developing the embedded software, interest to obtain the maximum level of profit with high level of quality. However, Changes of original requirements in software development process are inventible. Requirements maybe modified, deleted and added during life cycle. This can leads the software developers in these companies complain from these changes in requirements. Moreover, it may arise some problems with other phases in the development process [30].

Software evolution of embedded software in market place needs continuous management of changing requirements to minimize the complexity and chaos of their environment. When developing embedded software in business environment, we must define the business requirements. Which determine the business problem to solve it in the flaws in embedded software product. Moreover, business requirements define the reasons for why the software product is being developed [31].

The requirements of embedded software are a critical; it effects on the types of the hardware and software of the system. Thus, we must classify the requirements of embedded software into two types: hardware requirement and software requirement. Thus, good implementation of requirements in embedded software is a sensitive factor to the success of the development process [30].

Users and customers form essential role in software development. Requirement process (RE) focuses on collecting all requirements from users to make the requirements specification document before preparing design phase [32]. Where these requirements must be defined and accepted from users and suppliers of software product before software is generated [33].But these requirements can be changed through the development cycle [32].

Requirements change over lifecycle due to changing in customer needs .Maybe customer tend to change the existing requirement or identifying new requirement. However, the requirements depend on each other in a complicated way. When make change in requirements it must be trace the relations between requirements .Because these changes can effect on other requirements and appears a new relations based on a new changes. Thus, it was reflecting on architecture and code [32].Furthermore, managing the customer is difficult. Customers demand many changes with better quality of software product and minimal cost .Moreover, the customer needs at the beginning is not the really he/she wants. Therefore, we must try to better understand their business [34].

The requirements divided into two general types in all software products: function requirement (FR) and non-function requirement (NFR).The embedded software product look to FR and NFR are equally important. On the contrary, the traditional software product considers the NFR is less important than FR. Moreover, the success of any embedded device or application in this environment is NFR such as performance, quality and reliability. In addition, these products provide additional requirement, as Potential interactive with other application. Also, provide high level of security where most embedded software cannot attack it and effect on operation [35].

Real time access is a critical thing in business performance data for organization to response of changing in business environment and run a competitive business. Analysis the time and controlling of all processes are essential to determine the non-compliant cases and make decision of these inconsistencies to obtain the response from competitors. Therefore, some organizations combine between the integration of business intelligence (BI) systems and Process Aware Information Systems (PAIS) and use it as a key tool to support the business users in decision-making [36].

The requirements collecting by the Requirement Elicitation (RE) activity. It extracts and defines the requirements from the stakeholders, through the elicitation techniques. In addition, it considered the first step to define the problem and to solve it. The developers in RE must analyze the customer's needs in a deeply way in order to define the importance of requirements needed and risk achieving the success of project, and decreasing the time consumer. The RE allows the stockholders to add new requirement, updates on the existing requirements or remove some requirements if needed it. After that, the analyst defines the critical customer requirements [37]. As seen in figure2.

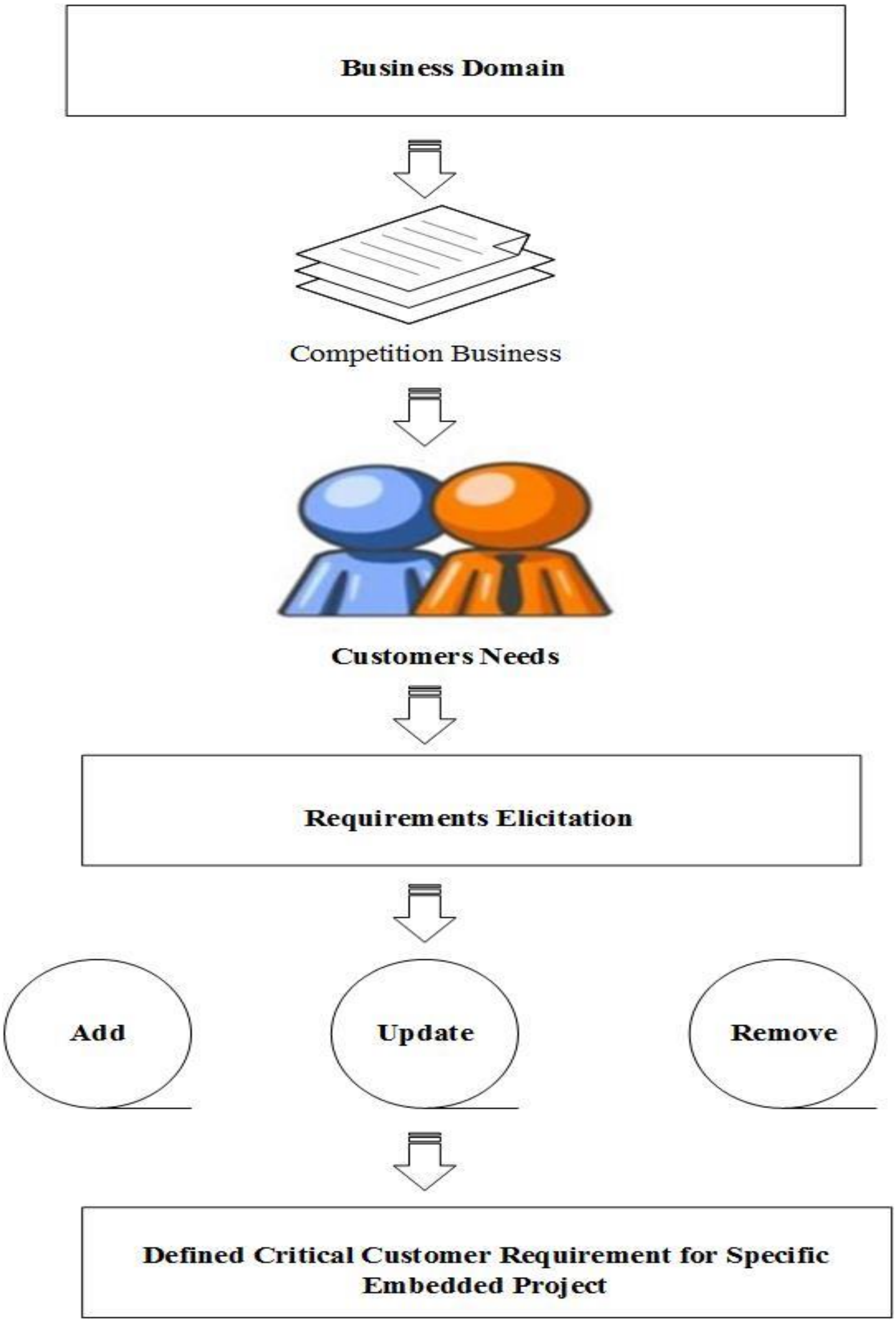

Figure 2: Requirements Business Domain of Embedded and Real software

The agile approach is the most appropriate and the best solution from all other software models. Agile has remarks on the embedded software [38]. In addition, it can reduce the cycle times when developing the embedded software. In addition, the development of embedded software is a difficult process because this software is a part of the physical environment which demanding more time and complicated dynamic [39].

Currently Agile Software Development (ASD) used to deal with increasing complexity in software development process [40]. In addition, the agile model focusing on the involvement of users and stakeholders in requirements phase .The highest priority of agile model is the user satisfaction through early delivery of software .Moreover, agile have many methods such: DSDM, SCRUM and Extreme Programming [41].

The traditional software development models demand of understanding of each requirement before moving it to the next phase [42]. However, the agile approach allows enter a new requirement even in late time in the development process .And also agile method is adaptive process more than traditional methods. Especially SCRUM methodology. It considered as control and a management mechanism with flexibility and productivity. It focusing to develop software product based on business needs [43].

4.2 SCRUM Software Development Process

4.2.1 Introduction

Users in the complex software project cannot have the complete picture of the final product. Thus, the requirements are insufficient. In this case, the iterative model is the most appropriate to allow the user to introduce new requirements in late phase [44].The agile methodology is the main type of iterative model. The most agile method used in software industry is the scrum. Scrum the most commonly used for development process for small team. In addition, it works well for complex software project [45].

Scrum is the most common agile methodology for completing the complex project. That has become the most popular methodology in distributed software development for small teams in software industry [46]. Furthermore, Scrum considered as iterative and incremental project. Moreover, it controls and manages the process in order to achieve the building software that appropriate with business needs. In addition, it provides ability to modify the requirements of project, even in late time in development process. The scrum process has ability to modify smoothly the organization requirements [47].

Scrum delivers the software as Sprints. In each sprint, it begins with the sprint-planning meeting. In this meeting, the product owner and team discuss which item moved from product backlog to the sprint backlog. Finally, the scrum team creates potentially shippable product increment at the end of each sprint [48]. Between these phases cannot introduce new requirements. In order to ensure the final product complete development in a successful way. Furthermore, each member in scrum team must understand his or her role. Thus, all members in scrum team must focus on single goal [49].

Scrum framework is iterative, adaptive and flexible. In addition, it designed to deliver the project with more productivity. Furthermore, scrum does not demand the full understanding of the requirements before transferring to the next phase. Conversely, in traditional approaches, however, the requirements are clear and known [50].

Scrum has been especially useful in software project with specific proprieties: short development calendars, systems based on new technologies, small and medium teams and constantly modify in requirements [44].

4.2.2 SCRUM Activity

4.2.2.1 Scrum Team

The scrum team is a small group of people work on geographically distributed that provides a good result. All team members make daily meeting during each sprint. This meeting called the daily scrum. It formulated as a five or nine member's .all members in teamwork on the projects as one unit to complete the specific work to complete the sprint [51].

The scrum team is self-organizing. That means there is no existing one person as a leader in Scrum team. The Scrum team members contain product owner, Scrum master, and some developers to develop sprint. The most critical member in scrum team is the product owner. The Product owner was deal directly with stakeholders then represents their needs on the scrum team. Moreover, the first person responsible for the team’s success because he was driving the entire process also defines the goals and direction. In addition, the product owner must achieve a good result in order to get satisfaction stakeholders [52].

The second role in scrum team is a Scrum Master. Scrum Master controls the relationship between the product owner and others members of the team, he/she works as a moderator. Furthermore, he/she can work as the coach, understands the scrum process, and explain how the rest of members can use it in order to achieve a high level of performance. In addition, the Scrum Master has the ability to extract and remove the impediments in the scrum process through communication between product owner, stakeholders and business owners [52]. The scrum team deals with business owner and stakeholders while the stakeholders consider the main source of validation for the software product [51]. Each member of scrum team have skills to do the specific jobs. Moreover, all members in scrum team have a certain task such as answer phone calls, maintain old projects and take bug [46].

4.2.2.2 Scrum Process

In scrum environment, the project moves from step to another through series of iterations known as sprints. The first step in scrum process is defining the stakeholder requirements and storing it in Product Backlog items. Product Backlog items considered as a specific repository in scrum process [53]. It stores all desired features of a product such as requirements, bugs, customer requested enhancements and other features. The product owner (PO) must be aware of all types of requirements in the software product. In addition, he/she is responsible for controlling the requirements and ensures the all relevant requirements written in product backlog [50].

The PO considers the project key customer. He/she makes negotiations with stakeholders to learn what you need to build the software product. The customers explain what you want. Then the PO defines the customer's requirements. After that PO analyzes, the requirements into FR and NFR. All types of requirements put into requirement control process [54].

The requirements control process contains all requirements. The PO defines the relevant requirements. In addition, determines the state of the requirements by using specific methods. After that, write the important requirements in natural language this known as Documentation phase. Finally, the verification phase where must compare the current requirements with the quality aspects. The output of the requirements control process used as input to preparing product backlog [55]. Moreover, the PO makes the list of necessary requirements. Maybe changes are FR or NFR for project development .This list known as a Product backlog. As shown in figure 3.

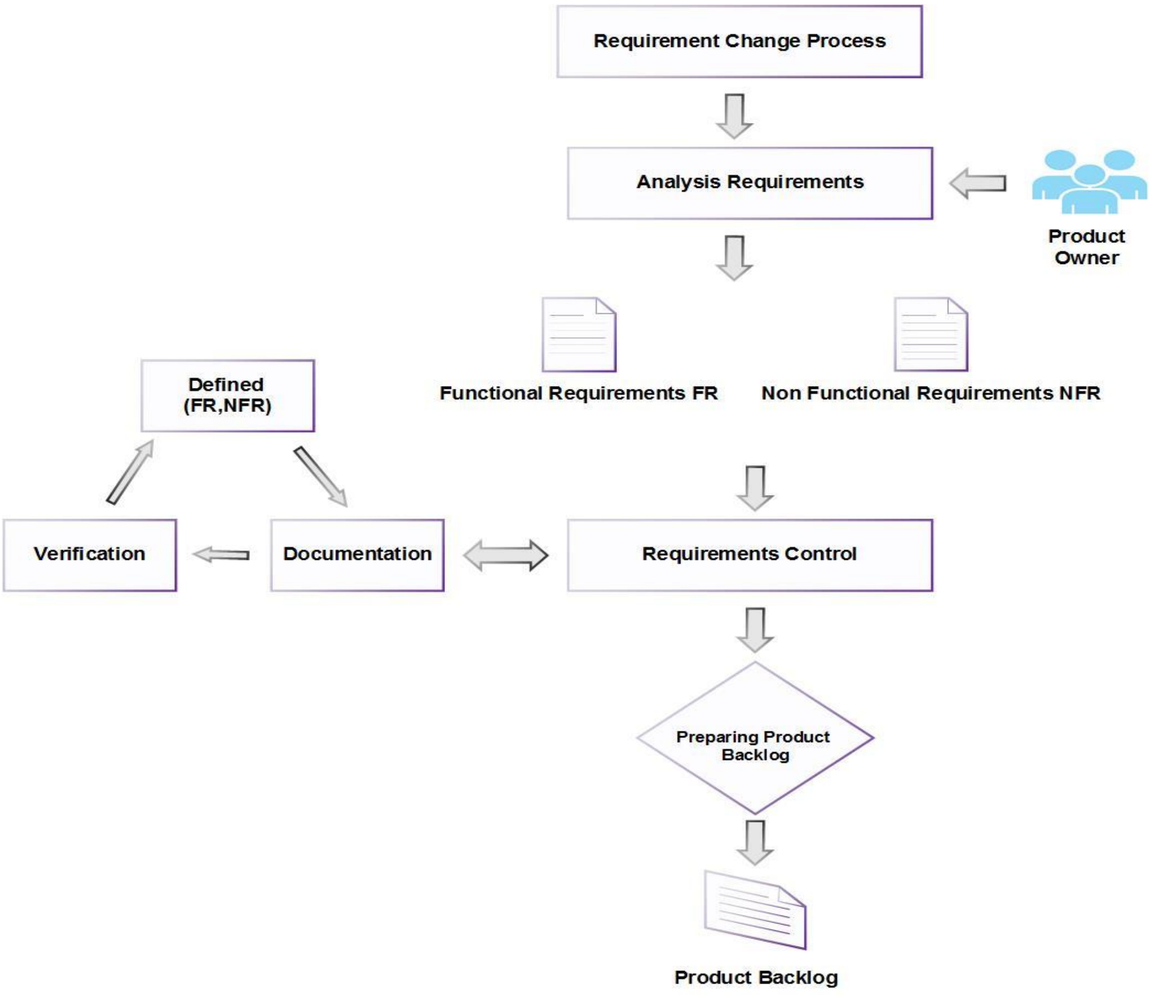

Figure 3: Generate Product Backlog

The product backlog is the only source of the requirement for the product. It ordered according to some criteria. The top items in product backlog analyzed and executed before other items. After the completion of the preparation the product backlog, the product owner holds the sprint-planning meeting with scrum team to prepare the sprint backlog and decides which items must move from the product backlog to the sprint backlog. The Sprint Backlog contains the collection of particular items category as a list of tasks. After that, it enters it into sprint execution [46].

Sprint is period, meaning that the period to execute the sprint is defined and date of delivering is determined. Through sprint execution, the developer's team make daily meeting from one to four week. Daily meeting allows the members to understand the team's status and know what you have to do. The Scrum Master (SM) is responsible for teaching all members of the team about the project. Moreover, SM helps the team to solve any problem in their jobs. Thus, the team can work on sprint in an efficient way as much as possible [56].

In fact, the actual work in scrum process is sprint. It is the heart of the scrum process (see Figure 4). In execution, process, all members of the teamwork daily on sprint. In addition, the team has the ability to choose the best possible tools to support the execution sprint such as bug tracking systems, managing Backlog Items and other tools [46].

At the end of each sprint execution, the developer's team held the sprint review meeting. It is an informal meeting. The attendees include the product owner, stakeholders, developer's team and scrum master. The product owner invites them to make modification on the product backlog if needed. The meeting show the completed work to the stakeholders. In addition, define the work that was not completed [55].

The developer's team in this meeting shows the potential shippable product to the product owner and stakeholders. That means the product is ready to deliver to a customer. Scrum team can introduce new enhancement on the next sprint through holding the Sprint Retrospective. The Sprint Retrospective held after the Sprint Review and before the next Sprint Planning [57]. (See Figure 4)

Finally, the Scrum team provides their prototype to allow the customers to test the process and ensure their requirements achieved or not. In addition, this prototype shows the new functionality to take the feedback from the interested people.

Feedback gives the scrum team opportunities to improve. Then, being accepted or rejected.

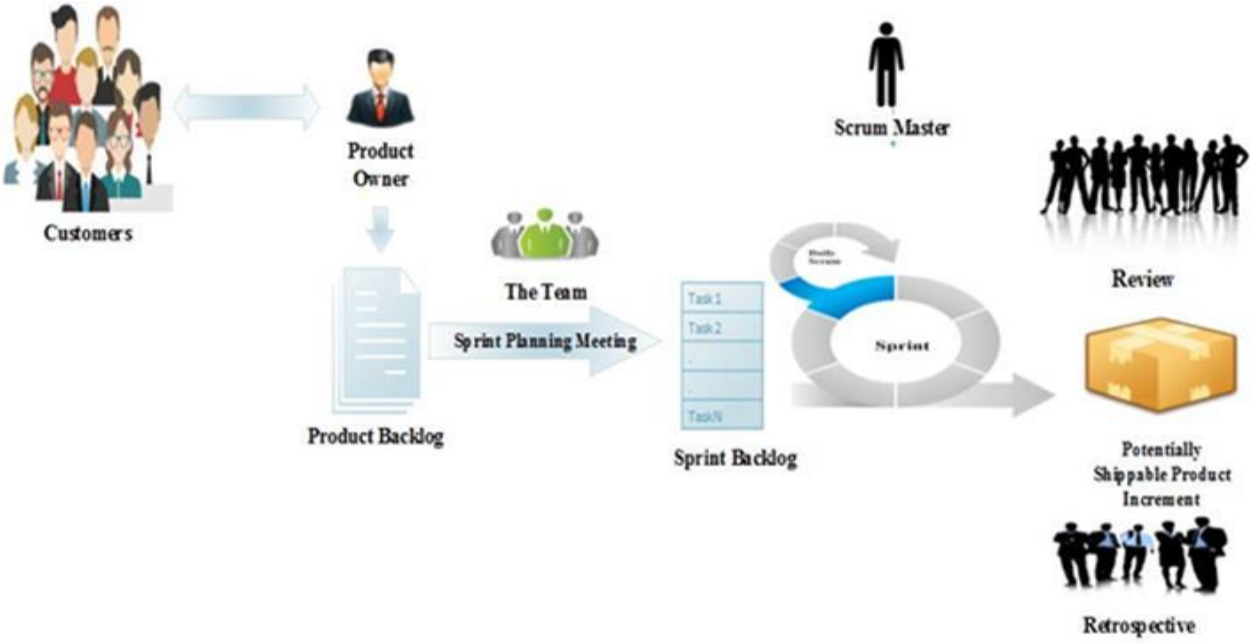

Figure 4: Scrum Process

4.2.2.3 SCRUM Principles

Scrum principles are guidelines for helping the scrum process. In addition, it is mandatorily use in all scrum process. The most important principles are Artifacts, Events and Roles. The main artifact in scrum process is a product itself [58].The expected from Scrum framework is delivering the product. The other artifacts are product backlogs is a list of product change that will add on the product.

In addition Backlog items is a part of product backlog with the necessary changing and Sprint backlog is a list of demands which the team must perform it during sprint execution[59].

The next principle is roles. The scrum process has specific roles. There are three main roles in scrum framework during development process. They represent

Scrum team where each member in the team does not have a job description; they must have the ability to do any task in the process. These roles contain Product owner, Chief product owner, Scrum master and some developers. All members must have trust between them and work in a closed sessions [60].

Third principle is Event. The scrum process passes in sequence of events. These events called time-boxed events. That means every event has a specific duration to execute it. The scrum framework has vital events. The sprint consider as the heart of scrum process. New sprint starts after the end of the previous sprint. In addition, the sprint includes Daily Scrum meeting, Sprint review meeting, Sprint retrospective meeting and Release the product [53].These principles applied in any project in corporation.

العنوان:	Software Configuration Management for Embedded and Real Time Software in Software Engineering Domain
المؤلف الرئيسي:	الزواهرة، حنان حمدان أحمد
مؤلفين آخرين:	الصرايرة، خالد تركي، المخادمة، خالد عبدالكريم(مشرف)
التاريخ الميلادي:	2017
موقع:	الزرقاء
الصفحات:	1 - 52
رقم MD:	856632
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	عمادة البحث العلمي والدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات، برمجيات الزمن الحقيقي، إدارة تكوين البرمجيات، برمجيات الأنظمة المضمنة
رابط:	https://search.mandumah.com/Record/856632

CHAPTER FIVE
SOFTWARE CONFIGURATION AND MANAGEMENT FRAMEWORK (SCMF)

The purpose of the Software Configuration and Management Framework (SCMF) generic steps is to establish and maintain the integrity of the products of the software project throughout the project's software life cycle. Software Configuration Management Framework SCMF involves identifying configuration items for the software project, managing and controlling these configuration items and changes to them, and recording and reporting status and change activity for these configuration items.

Due to the lack of the “Software Configuration and Management Frameworks to support, a nontrivial product line requires tool support, and there is no shortage of available commercial CM systems” [21]. However, most of them do not directly support the functionality required for the CM to be useful in a product line context. This chapter proposed a Software Configuration and Management Framework and its elements to support of the development process for the new or enhanced products of the embedded and real time software.

Software Configuration and Management Framework Phases:

- Business Domain System of Embedded and Real time Software.
- Critical Customer Requirement for Embedded and Real Time Software.
- Requirements Change Process and Design Using Scrum.
- Software Configuration and Management Process for the Designed and Implemented Systems.

Essential Elements for Software Configuration and Management Framework (SCMF)

The most used elements of the Software Configuration and Management Framework in the previous work include the following steps illustrated in Figure 5.

- “Identify the configuration items, components, and related work products that will be placed under configuration management.
- Establish and maintain a configuration management and change management system for controlling work products.
- Create or release baselines for internal use and for delivery to the customer.
- Track change requests for the configuration items.
- Control changes in the content of configuration items.
- Establish and maintain records describing configuration items.
- Perform configuration audits to maintain the integrity of the configuration baselines” [21].

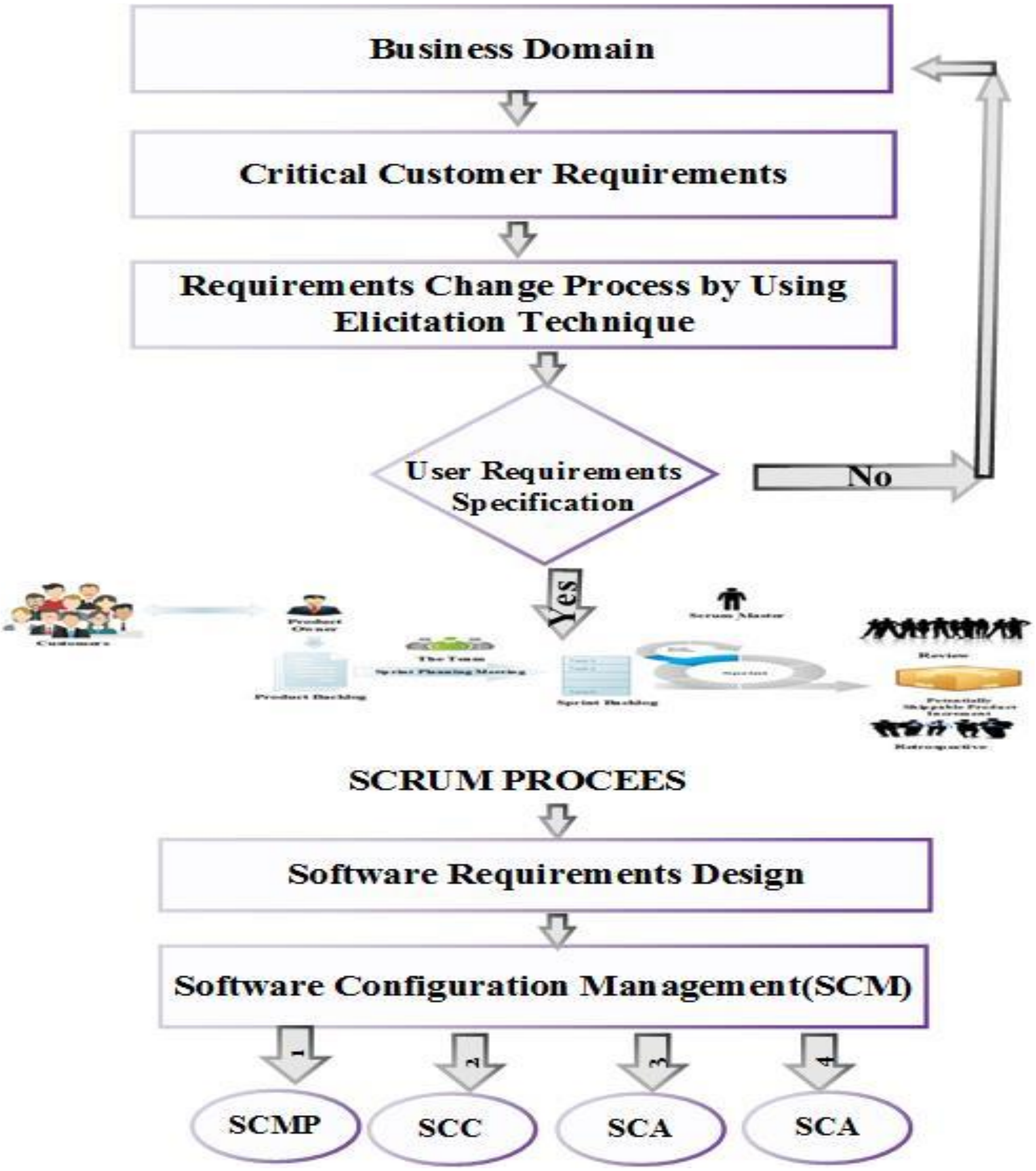
Business Domain
Critical Customer Requirements
Requirements Change Process by Using Elicitation Technique
User Requirements Specification
No
Yes
Customers
Product Owner
Product Backlog
The Team
Sprint Planning Meeting
Sprint Backlog
Scrum Master
Sprint
Review
Potentially Shippable Product Increment
Retrospective
SCRUM PROCEES
Software Requirements Design
Software Configuration Management(SCM)
1
2
3
4
SCMP
SCC
SCA
SCA

Figure 5: Software Configuration Management Framework (SCMF)

5.1 Business Domain System of Embedded and Real time Software

Business Domain System of Embedded and Real time Software built on the bases of business competition and customer needs. Furthermore, the software industry is growing rapidly in the world. In addition, it is becoming the base of other industries. For example, e-business and other sectors increasing rely on the software system, especially in a complex project. When analysis any system in a business domain we have two types of systems: Embedded software and Real-time software.

5.1.1 Real-Time Software (RTS)

RT designs the hardware and software to compatible with the RT requirements. Moreover, RT projects can control the surrounded environment by extracting the data from it. Then, it makes processing on it and return the result quickly at that moment. Such anti-lock brakes system on the vehicle. It divided into two main classes based on react of the time: Soft RT (SRT) and Hard RT (HRT).

RT is a system interacts with the external environment with a limited amount of time (time constraints), where the main feature that distinguishes it from other system is, and time “deadline”. In order to achieve a high level of RT correctness, they must guarantee implementing the operation on time. In addition, RT plays a vital role in our society.

RT is known as HRT system if the delay in time can cause great losses in human life or catastrophic. Thus, HRT guarantees the important task completed on specific time. Conversely, the SRT the delay in time (deadlines) are not fatal, which means it does not lead to catastrophic problem.

5.1.2 Embedded Software (ES)

Embedded software ES is a type of software industry. It is in everywhere. Customers think that these devices are hardware but actually, this is belongs to the software product. The development process of ES and real time is difficult. These systems are a portion from physical environment whose complicated structure and demanding many requirements must adhere to. Such as reliable, protected from many target manipulation, safety, power consumption, and timing requirements all these NFR leads to the success of the ES.

ES designed to do specific functions or support operational function of complex software. Moreover, it does not necessary input to work but execution through external controls. We take in consideration the secure and safety requirements.

5.2 Critical Customer Requirement for Embedded and Real Time Software

The Software critical system (SCS) is any system which failure causes a threat on human life. SCS considers people and operational processes are essential factors to its success. Failure in SCS maybe primary or secondary. The primary failure critical system causes problem in association hardware and threaten people in the system itself. Conversely, the failure in secondary critical system, which can cause problem in other system.

The requirements of ES are sensitive; it is effects on the types of the hardware and software of the system. In addition to the customers and users who plays a main role to define and modify the requirements. Thus, the requirements classify of ES into two types: hardware requirement and software requirements. The nature ES system is adhere some strict requirements. Customer expectation of this type of system meets the most demanding performance. The result of failure in safety critical software can cause the physical accident or lead to harm to some people (e .g controlling traffic signal). As for the business, critical software does not lead to physical damage that means the failure is not the grave. However, can causes threats the constancy of a business (e. g banking).

5.2.1 Real-time and ES (RTES)

When real -time software designed as embedded element, it known as a real-time embedded system (RTES). It is a computer system on customer hardware boards that control and response the external environment. This system interacts closely with the external world. It exists on the end user software product such as anti-lock brake system. In addition, it plays the main role in the modern application domains. When defining the requirements of RTES, we must understand the software area of these devices. In addition, analysis the current cases carefully are necessary to define the specific needs. The most important requirement of RTES is the first one is a time constraint. That means, the RTES should be provided not only the right output but also provide them at the correct time. The second one, Predictable real-time performance guarantees the response time of system even in the worst case. Hence, the good implementation of RTES requirements considers the critical factors to success the development process. This type of system must be the right response in correct time. If the response is late, it can lead to failure.

5.2.2 Real-time, safety-critical, embedded software (RTSCES)

RTSCES is embedded software focusing on avoiding undesirable situations more than required features. In addition, it adhered to qualification requirements and constraints. Which means it was whose failure in it cause damage in human life. When we say safety critical software, we imagine any failure or malfunction in software can lead to injury or death of human life (catastrophic).

RTSCES deals with the system in avionics and automotive where these systems have the ability to expand from one -core architectures to many-cores. In order to improve and introduce new features of quality attributes such maintainability, performance, safety and other attributes. Thus, the poor requirements can cause many failures in a software project. Currently, many aspects tend to operate multiple applications with high level of speed. These types of applications designed to execute it under all conditions.

5.2.3 RTES and mission Critical (RTMES)

Currently, RTES introduced in many mission critical settings. Because it deals cautiously with deadlines. Hence, RTMES concern with time constraint and considers it as a critical factor to success the mission critical. Thus, it can cause catastrophic if any failure has been occurred in main the components of software or in some goals of activities. System can effect on business.

5.2.4 RTES and business Critical (RTESBC)

RTESBC is a system deals with another system, which have many transactions, or other critical data of business, any failure in critical business function system can cause financial losses. Requirements phase are essential phase in software development process. The software build based on customer needs. Good determined requirements of software functionality can obtain satisfaction of customer needs. When building the critical software, the developers define the level of functionality of the system.

5.3 Requirements Change Process and Design Using Scrum

Scrum is an incremental and iterative framework managing the development process of software product. Each Sprint starts with negotiating between the Product Owner (PO) and stockholders to define the requirements. PO receives the list of requirements of ES from previous step. Then PO puts the requirements in product backlog. It contains all new features that will be added into the system. After that, the PO starts to prepare the sprint-planning meeting.

The PO makes a contract with the development team in order to arrange the tasks in sprint backlog based some criteria. They entered the sprint backlog into sprint execution. During sprint execution, the Scrum Master (SM) makes a daily meeting with the developer team from one to four weeks. In this meeting, they discuss the problem and solve it. Also, introduce the new enhancement. After finishing the sprint meeting, the SM held a review meeting with the end user to show the software product with new modifications. The most common problems appear after the system delivered to the client. These problems resulted from elicitation process, not from implementation process [61].

Business sector tends allowing the change in practices of the companies by determining needs with the suggestion solution that provides return to stakeholders. When defining requirements, we use cases for elicitation and the analysis of the functional requirements; it is representing the performance and behavioral requirement of a software system. One of the most important requirements is time constraints; the end user sends a request to RTS to operate a specific function, RTS receiving and analysis the request. After that, RTS make the needed processes with limited time (deadlines).

Finally, returns the result quickly to the end user to affect the external world at that time. Limited function of embedded software designed to limit functionality. This can provide less control it compared with other software.

5.4 Software Configuration and Management Process for the Designed and Implemented Systems

The organization uses SCM to manage the changing in the software product. Moreover, SCM considered as an umbrella activity for application through the software development lifecycle.

5.4.1 SCM Planning

SCMP is the main plan of the SCM process. It starts in early phases of a software project. SCMP defines the mandatory policies, responsibilities, and procedures for change control. In addition, it is responsible for defining the documents to be managed with the specific actors in teamwork. Teamwork contains variety members such developers, software configuration manager, users, and tester.

SCI considers the first step in controlling change in a software product. In addition, it known as software artifact or configuration Items. SCI defined as a combination of software and hardware. These items must be defined to manage and control the software product. Each SCI has uniquely identified by version numbers with specific functional and physical features of the design, code, and elements, which should put under version control. After that written all SCIs in documentation. A SCI it is a basic item (primitive) or aggregation items.

The main purpose of SCI is to select a valid set of configuration items. The selection process of SCI is important to reduce the cost and time moreover, provide an ability to define the software quality. Conversely, if we select the invalid configuration items this can lead to rising cost and the development time with the low quality of software project. Thus, the good SCI is pre-request to other SCM activities. SCI either manages or design model (such as libraries, source code, design model, and documentation). After that, the SCI builds configuration baseline for the configuration control of CIs. Finally, all artifacts that identified and stored in the repository.

The baseline is a reference point (snapshot of the combination of system artifacts) in software life cycle. The main target of this baseline is minimizes the uncontrolled modification by allowing the change just in the formal way. Baseline provides the basement for continued development the configuration items. It was can categorize two labels. One for identifies the baseline itself and the second for identifies the update on specific baseline.

5.4.2 Software Configuration Control (SCC)

The software is easy to change during lifecycle and uncontrolled modification can lead to chaos SCC deals with controlling the change on a software project. The main function for the SCC is to give the administrative mechanism for the creation and preparing project. Analysis, and agree or disagree all change proposals during software development.

SCC ensuring if all CIs put under control to define any change on it. In addition, it maintains the consistency between all parts of a project. Thus, when making any illegal change later, it will be prevented [62]. Moreover, if the developers want to make changing on the controlled item can have a private copy of the item. SCC has many tools to support controlling process such as document version maintenance. Show as Figure 6**.**

1. Identification and documentation change request for embedded software

The change request (CR) is an official request to make the modification on baseline configuration item. The controlling of change starts from change request, where the change request can come from many things such as introducing new feature during development process on existing feature. Anyone can send the change request at any time in the life cycle. All change requests written in change request note [25]. The requested change passed through a higher level of authority to approve this change. This process known as a promotion for CIs. Then change request submitted.

2. Evaluation and analysis of change request to prepare the change proposal for embedded software

The developers assess the change request through perform analysis to define the impact of the proposed change on other work in the development cycle. Such potential affects the reason for changing and defining the cost impact. The result of request evaluation presented as a report. Moreover, the developers test the proposal change to decide if is doable.

3. Approval or Disapproval for change proposal of embedded software

The change report used from the change control board (CCB). CCB area committee to make the decision of change. They have the authority for approving or disapproving the proposed changes to configuration object during software development. Moreover, CCB ensuring the proposed changes done as customer needed.

CCB levels depending on the complexity of software project. In the small software project, the CCB is a leader or is the one CCB. However, in the large and medium software project have more than one CCB. It makes a review to communicate with customers and project team. In order to get the requirement baseline with the proposal change. Finally, CCB decides which is unnecessary and necessary CRs then reject it or approve it.

The engineering change order (ECO) is generated after the CCB approved the CR. ECO is a procedural document to controlling the change in a software project. It contained information about the proposed change such as identification of affected CIs, details of the proposed change, set of constraint and specification of schedule and cost impacts. It used to make updating the component. Thus, the good practice of ECO provides documentation contain the history of all changes and when they do it.

4. Implementation and release change for embedded software

The implementation phase starts with authority to make a change and end with accomplishing the change. The programmer is responsible for making the changes on the objects, which stores in the directory. Therefore, we use some mechanism or tools to facilitate the controlling and updating process on the original object

such Version Control mechanism (VC). VC play the main role for build, release and tracking system.
The first step for making the change on specific object is "Checked out" of the project directory (repository). When you have the private copy of object, we can modify it as the description in VC after that applied the proper software configuration management on it. Then save it. After we finish the modifying process on the object, we back "Checked in" the object into the directory. Based on these changes we can establish the next version of the software project. When the change occurs in system or subsystem, the version number updated to reflect the change. Each change in changes document has identification number. In order to facilitate the tracking change process.

VC classified into to three types of change one: Minor version change reflects the one error being correct or these change no impact on other components in the system. Second: Substantial Version Control this indicate the change is correcting on more than one minor defect simultaneously. In addition, this type effect on another component in a system.

Finally, Severe Version Change is a major change in functionality of the software project. When implementing more than one approved CRs simultaneously. We need a tool for tracking the impact of change on and the entire system. In addition, ensuring the requested change meeting the requirement specification of the project. The test and quality assurance (QA) are the most critical point of the implementation plan. It started after the change approved and implemented. After the formal review, the developers established a new baseline. Once the object becomes baseline, the software project implemented. Then cannot make any change without agreed from a project manager. Finally, at the end of CC phase. The developers prepare a report to support the Configuration Status Accounting (CSA). This report contains a list of all CRs with the status of each CR.

5.4.3 Configuration Status Account (CSA)

CSA also known as status reporting. It is an essential activity of SCM and continues during the lifecycle. The first element in CSA is determined the current approved configuration and version number. It was responsible for recording all information about CIs and displaying the report. This shows the status of each baseline with proposed changes from initiation to the final agreement. This information needed to manage the updating of the software project in an efficient way.

The CSA reports can show the implementation status of authorized change. In addition, it provided traceability from the original baseline of each SCI. Many organization, maintenance team, quality assurance activities and development team can use these reports. In addition, CSA considered the administrative tracking and provided a report about all CIs.

5.4.4 Software Configuration Audit (SCA)

SCA is an evaluation and review of work product or group of work products to determine the compliance and consistency with constraint, specifications, and standers. In addition, SCA has the ability to an assurance of validation the software project implemented as planned in requirement document. CA provides the items, which satisfy with the required physical and functional characteristic.

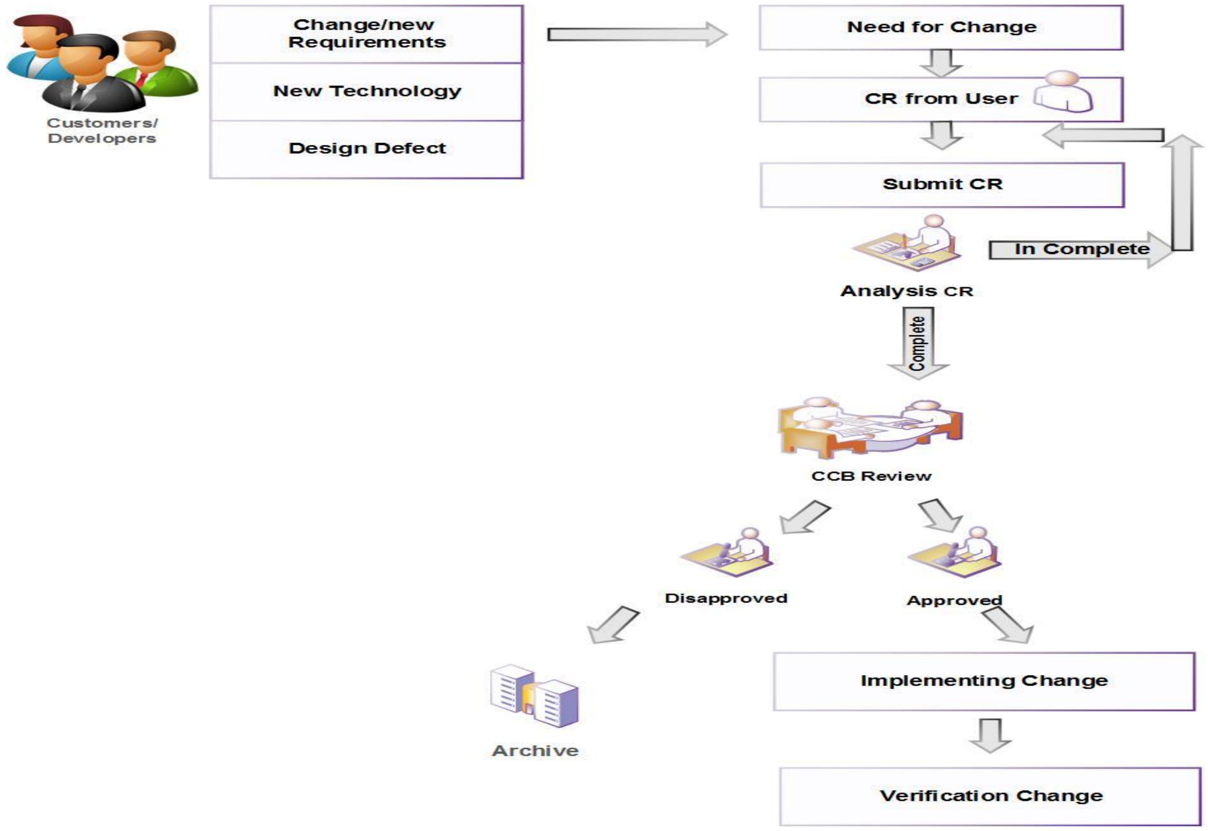

Figure 6: The Configuration Control Process

العنوان:	Software Configuration Management for Embedded and Real Time Software in Software Engineering Domain
المؤلف الرئيسي:	الزواهرة، حنان حمدان أحمد
مؤلفين آخرين:	الصرايرة، خالد تركي، المخادمة، خالد عبدالكريم(مشرف)
التاريخ الميلادي:	2017
موقع:	الزرقاء
الصفحات:	1 - 52
رقم MD:	856632
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	عمادة البحث العلمي والدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات، برمجيات الزمن الحقيقي، إدارة تكوين البرمجيات، برمجيات الأنظمة المضمنة
رابط:	https://search.mandumah.com/Record/856632

CHAPTER SIX

CASE STUDY (RICE COOKER)

This case study shows the simple Rice Cooker. Rice cooker is one of the electric technologies. It designed to cook the Rice but have some challenges as overcooking and burning rice. It contains a kettle for rice and at least one heater used for heating the kettle. In addition, the prototype of the Rice Cooker has two main buttons: start and stop buttons. In addition, there are three cooking modes: Slow, normal and fast. Moreover, Rice Cooker has a lamp to inform the operator if the cooker is cooking or not (see Figure 7).

6.1 Business domain System of Embedded and Real Time Software

The Rice Cooker is an embedded software. When we develop or introduce new modification on it, we may face some problems. To control these modifications we tend to use SCM. This case study aims to clarify our framework. How to control the changes of embedded and real-time software. In order to increase the efficiency, reliability, and traceability of Rice Cooker in the industrial sector (see Figure 5).

6.2 The Critical Customer Requirements and Changing Requirements Process for Embedded and Real-time Software

Rice cooker is real time safety critical. It was focusing on safety and reliability requirements, especially when running it. Where the functional requirement and non-functional requirement is fundamental for an embedded system. In addition, it is considered as succeed factor for it.

6.3 Requirements Change Process and Design using SCRUM

After defined the essential requirements of Rice Cooker send it into the SCRUM process .at the beginning the PO receive the list of requirements. Then put it in Product Backlog that contains the warning button with details. Then prepare the sprint backlog with the development team. After that, enter it into the sprint execution. Finally make meeting to show the rice cooker after added the warning button.

6.4 Software Requirement Design

We use SCMF to manage the changing on simple Rice Cooker to produce new version of Rice Cooker with new design (see figure 9). We add new button called “Warning Button”.

6.5 Software Configuration and Management Process for the Designed and Implemented System

SCM can help the developers to manage the new changing in Rice Cooker. It is the task of tracking and controlling changes in the Rice Cooker .SCM have four elements (see figure 5).

6.6 SOFTWARE CONFIGURATION IDENTIFICATION (SCI)

In embedded system, physical hardware device can be functional user, thus we can define the items of new version of Rice Cooker (see Figure 8) as follows: (Start Button, Heater, Timer, Cooking Lamp, Temperature Sensor, Cooking Mood, Stop Button, warning Button). Each SCI has a unique version number and store it in repository.

6.7 SOFTWARE CONFIGURATION CONTROL (SCC)

When applying the modification on Rice Cooker we must control all SCI and tracking all changes on it .We prosed change on the simple Rice Cooker prototype in order to solve some problem faced the Rice Cooker machine.

6.7.1 Change Request (CR)

The software engineers submit the CR to the SCM leader. CR is introducing new button known as “Warning Button”. This button has two colors (Green and Red) expressed the status of rice. In order to overcome the overcooking or burning rice. That can be define some factors during cooking cycle:

A. The sensor of the temperature T of boiling water in the cooker: The temperature ranging between 0 To 100 C where:

If the T >100. The color is Red with beep sound.

If the T<100 And T> Zero, the color is Green.

B. The amount of water in the kettle.

The SCM leader:

- He/she will analyze the CR by using the SCI documents (current prototype of rice cookers) before add new change and the design document of Rice Cooker.
- He/she will define the impact of Warning Button on the entire rice cooker system and each subsystem on it.

6.7.2 Analysis the proposed change to approve or disapprove

The SCM leader analysis the Warning Button on Rice Cooker and decide if this changes is doable or not. Then make update on SCI to reflect the change. Moreover defined the impact of this change on other works in development cycle of Rice Cooker. All information about changing of rice cooker we recorded in report known as change report. After that the CCB receive the change report to implement this enhancement or discarded it. Thus, SCM Library updated to reflect the request.

If Request is disapproval (by CCB): the SCM Library updated to reflect the disapproval. (See figure 5.2).

If Request is Approval (by CCB): the SCM Library updated to reflect the approval. The SCI amended. (See Figure 6).

6.7.3 Implementation and release change for embedded software

The developers have a private copy (check out) of the items (last version) of Rice Cooker from directory. After we defined the items will be make change on it. Can be making the require modification. Then return it on directory (check in).thus we have new version of Rice Cooker with modifications. After that, the developers built new number (version number) to reflect the change.

6.7.4 Configuration Status Account (CSA)

The change has introduced by the SCM leader. Then the SCI document of Rice Cooker modified. And add the Warning button on it .CSA is essential activity were defined the version number of Rice Cooker with new button.in CSA record the information that will be needed for configuration management.in addition provide the report about the status of all SCIs, proposed changes and the implementation status of approved changes in new version of Rice Cooker.

6.7.5 Software Configuration Audit (SCA)

SCA must validate all changes on Rice Cooker make as written in requirement document. Moreover, it is responsible for ensuring the Warning button consistency with other parts of simple rice cooker prototype and applying standers.

- Audits of SCM Repository that contains the Rice Cooker items. In addition, defined the status of each items.
- Audits of status of approved CRs.
- Audits to ensure that CIs of Rice Cooker are consistent with SCM records.
- Audits to verify that the Rice Cooker CIs have satisfied with their specified requirements.

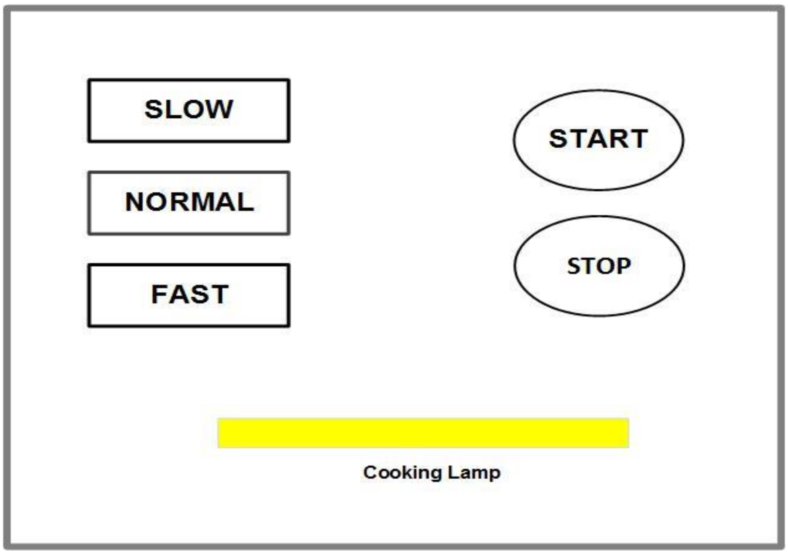

Figure 7: Rice Cooker Operator Control Panel (Firs Prototype)

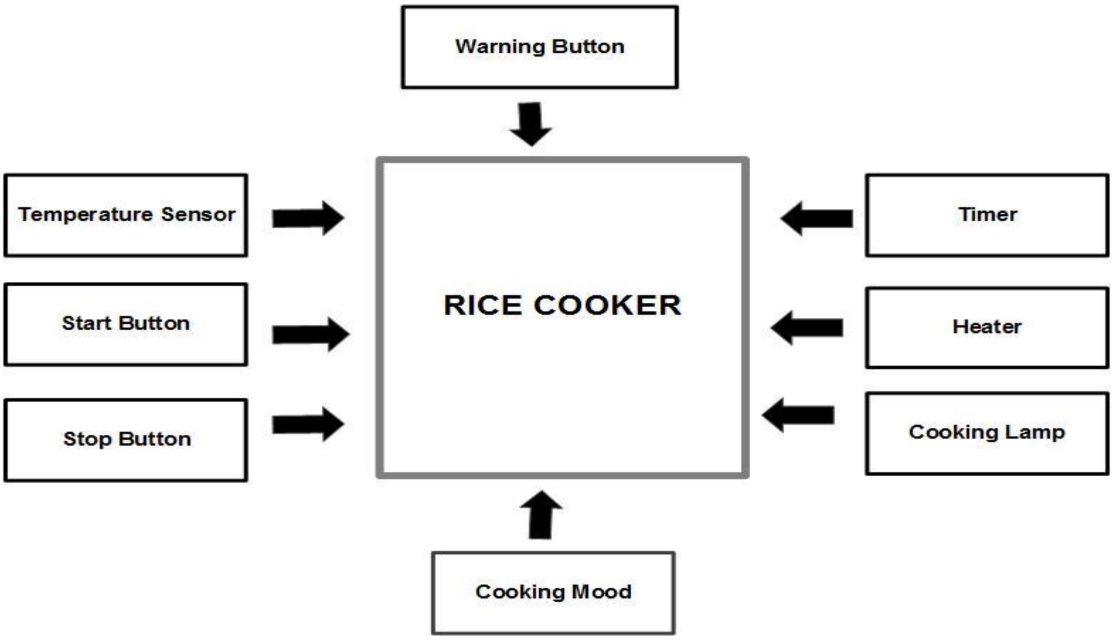

Figure8 : Rice Cooker Software from its FUR

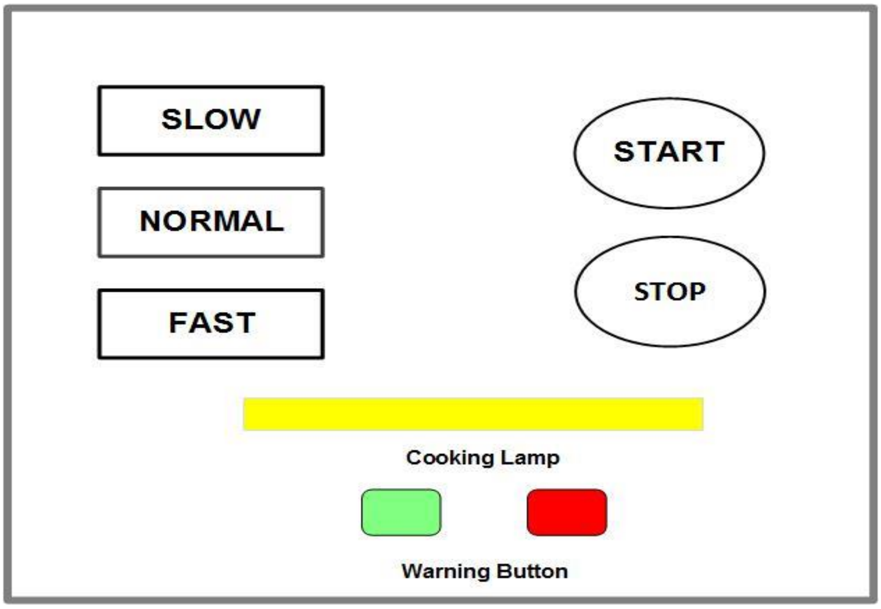

Figure9 : Rice cooker Operator Control Panel (New Prototype)

العنوان:	Software Configuration Management for Embedded and Real Time Software in Software Engineering Domain
المؤلف الرئيسي:	الزواهرة، حنان حمدان أحمد
مؤلفين آخرين:	الصرايرة، خالد تركي، المخادمة، خالد عبدالكريم(مشرف)
التاريخ الميلادي:	2017
موقع:	الزرقاء
الصفحات:	1 - 52
رقم MD:	856632
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	عمادة البحث العلمي والدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات، برمجيات الزمن الحقيقي، إدارة تكوين البرمجيات، برمجيات الأنظمة المضمنة
رابط:	https://search.mandumah.com/Record/856632

ABSTRACT

Software Configuration Management for Embedded and Real Time Software in Software Engineering Domain

By

Hanan Hamdan Ahmad AL-Zawahreh

Supervisor

Dr. Khalid Torki Al-Sarayreh

(Assistant Professor)

Co-Supervisor

Dr. Khalid Al-Makademeh

(Assistant Professor)

Nowadays, embedded and real time software produced as product families. Products in which have an amount in public, nevertheless all the same vary to some extent in one or more features. Developing and controlling these product families is a sophisticated mission. Software configuration management (SCM) is widely ranging sustenance the improvement and development of one single software product and correspondingly supports the concept of alternatives solutions. Furthermore, in software engineering, SCM is charge with adjusting and tracing changes in the embedded and real time software and it is concerned with managing and controlling all the changes during software development cycle.

In this research study, it would be interesting to explore to the previous and proposed software configuration management frameworks (SCMF) and their elements in the academia and used by industry. In the previous works, SCMF formerly has solutions to some of the problems of software product and these SCMF formulate new techniques to support software changeability, controllability and management.

This thesis proposed generic steps for the software configuration management that could be used as a guided framework in the work product environment. These generic steps or what so called SCMF is built on the basis of previous and published configuration frameworks in literature to manage and controlling the software development process of embedded and real time software.

The proposed SCMF in this thesis is to keep the changes and tracking it in software product during software development life cycle (SDLC). In order to develop and maintain quality software products. Moreover, the proposed and generic steps present configuration identification for all items of the previous works. Before it explains how to apply the change management and implement all the changes. The proposed generis steps in this thesis evaluated in automated software system, where we show the simple RICE COOKER prototype, and present the new enhancement, which we introduce to obtain new version of RICE COOKER throughout our proposed generic steps.

ملخص

ادارة تكوين البرمجيات للانظمة المضمنة وبرمجيات الزمن الحقيقي في حقل هندسة البرمجيات

إعداد

حنان حمدان احمد الزواهره

المشرف الرئيسي

د.خالد تركي الصرايره

(أستاذ مساعد)

المشرف الثانوي

د.خالد عبد الكريم المخادمه

(أستاذ مساعد)

في الوقت الحاضر، تنتج برمجيات الانظمة المضمنة وبرمجيات الزمن الحقيقي كالأسر المنتج. هذه المنتجات لها قيمتها في الأماكن العامة، ومع ذلك كل واحده تختلف إلى حد ما في واحد أو أكثر من الميزات. ان تطوير ومراقبة هذه المنتجات مهمة معقدة.إدارة تكوين البرمجيات (SCM) هو على نطاق واسع القوت لتحسين وتطوير منتج برمجي واحد ويدعم في المقابل مفهوم الحلول البديلة. وعلاوة على ذلك، في هندسة البرمجيات، SCM هو المسؤول عن ضبط وتعقب التغييرات في برمجيات الانظمة المضمنة وبرمجيات الزمن الحقيقي وأنها تعنى بإدارة والتحكم جميع التغييرات خلال دورة تطوير البرمجيات.

في هذه الدراسة البحثية، فإنه سيكون من المفيد استكشاف الأطر السابقه والمقترحة في إدارة تكوين البرامج (SCMF) وعناصرها في الأوساط الأكاديمية والتي تم استخدامها في الصناعات. في الأعمال السابقة،SCMF كانت تقدم حلول لبعض المشاكل في البرمجيات المنتجه وهذه SCMFقامت بصياغة تقنيات جديدة لدعم تغيير البرمجيات، السيطرة والإدارة.

اقترحت هذه الدراسه خطوات عامة لإدارة تكوين البرمجيات التي يمكن استخدامها كإطار توجيهي في بيئة منتج العمل. هذه الخطوات العامة أو ما يسمىSCMF مبنية على أساس أطر التكوين السابقة والمنشورة في الأدب لإدارة ومراقبة عملية تطوير برامج الانظمة المضمنة و الزمن الحقيقي .

SCMF المقترح في هذه الدراسه هو للحفاظ على التغييرات التي تحدث على المنتج البرمجي وتتبعها خلال دورة تطوير البرمجيات (SDLC).وذلك من أجل التطوير والحفاظ على جوده المنتج البرمجي. وعلاوة على ذلك، تقدم الخطوات المقترحة والعامه تحديد التكوين لجميع العناصر من الاعمال السابقة. قبل أن يشرح كيفية تطبيق إدارة التغيير وتنفيذ كافة التغييرات.تم تقييم الخطوات العامة المقترحة في هذه الدراسه في نظام البرمجيات الآلي، حيث تم عرض نموذج بسيط عنRICE COOKER . وقمنا بادخال التعديل الجديد على هذا النموذج للحصول على إصدار جديد منRICE COOKER .وذلك من خلال تطبيق الخطوات العامه المقترحه في هذه الدراسه.

Table of Contents

Software Configuration Management for Embedded and Real Time Software in Software Engineering Domain

By

Hanan Hamdan Ahmad Al-Zawahreh

Dr. Khalid Torki Al-Sarayreh

(Assistant Professor)

Co-Supervisor

Dr. Khalid Al-Makademeh

Submitted in Partial Fulfillment of the requirements for the degree of Master in Software Engineering

Deanship of Scientific Research and Graduate Studies at the Hashemite University

Zarqa – Jordan

16/04/2017

اتحاد الجامعات العربية

الجامعة الأردنية / مركز صندوق إيداع الرسائل الجامعية

406870

نموذج تفويض

أنا الطالب حنان حمدان احمد الزواهرة

أفوض الجامعة الأردنية بتزويد نسخ من رسالتي /أطروحتي للمكتبات أو المؤسسات أو الجهات أو الأشخاص عند الطلب .

التوقيع :

التاريخ : ١٣/٧/٢٠١٧م

Association of Arab Universities (AArU)
The University of Jordan

Authorization form

I am Hanan Hamdan Ahmad AL-Zawahreh.

Authorize the University of Jordan to supply copies of my thesis / dissertation to libraries or establishments or individual on request.

Signature :
Date : 13/7/2017

This thesis was defended on April 16, 2017

Examination committee **Signature**

Dr. Khalid Al-Sarayreh, Supervisor ....
Assistant Professor, Software Engineering

Dr.Khaled Al-Makhadmeh, Co Supervisor ....
Assistant Professor, Software Engineering

Dr. Fadi Wedyan, Member ....
Assistant Professor, Software Engineering

Dr. Hani Bani-Salameh, Member ....
Assistant Professor, Software Engineering

Dr.Qutaibah Althebyan, Member ....
Associate Professor, Software Engineering
Jordan University of Science & Technology JUST

Dedication

I dedicate all of my work for those who surround me with love and support:

To the soul of my father, Hamdan Ahmad Al-Zawahreh.

To my grateful mum, Yousra Mohamad Al-Zawahreh.

To my fiancé, Moafaq Nawaf Almuala for his patience, and understanding, and most important of all is love. You are everything to me.

To my lovely sisters, brothers, and my best friends.

Acknowledgment

My conclusive and definitive thanks are due to Allah, the almighty, for his uncountable bounties and blessings. In addition, it is my great pleasure to express my sincere gratitude and appreciation to my supervisor, Dr. Khalid Al-Sarayreh, for his patience,excellent mentorship,and for the time he spent on reading my research and improving it,and for his continous guidance.Also my deep regard to my co-supervisor Dr.Khalid Al-Makadmeh and the members of the examination committee.

My love, gratitude and special thanks go to my wonderful family and my fiancé. They have been a guiding light and source of inspiration during this long process. Finally, I wish to express my appreciation to my friends for their help and support overseas and over phones. Thanks for all.

Table of Contents

List of Figures

List of Symbols and Abbreviations

Abbreviation	**Meaning**
SCM	Software Configuration Management
VC	Version Control System
CM	Configuration Management
Commit	Uploading changed files to the repository
Checkout	Fetching files from the repositories and placing them in folder on the local machine to be able to modify them.
SCI	Software Configuration Identification
Repository	Directory containing files and all revisions of them
CHM	Change Management
ES	Embedded Software
RTS	Real Time Software
CSA	Configuration Status Account
SCA	Software Configuration Audit
CR	Change Request
QA	Quality Assurance
CCB	Change Control Board
ECO	Engineering Change Order
SCC	Software Configuration Control
SCMP	Software Configuration Management Plan
SM	Scrum Master
PO	Product Owner
RTESMC	Real Time Embedded Software and Mission Critical

RTESBC	Real Time Embedded Software and Business Critical
RTSCES	Real time Safety Critical Embedded Software
RTES	Real time and Embedded Software
SCMF	Software Configuration and Management Framework
ASD	Agile Software Development
ECU	Electronic Control
RM	Risk management
DAS	Driver Assistance System
RealSysId	Real-time system identification tool
CPS	Cyber –Physical System
EMS	Energy Management System
SQA	Software Quality Assurance
SPLE	Software Product Line Engineering
DVCS	Distributed Version Control System
CVCS	Centralized Version Control System
SVN	Subversion
VCS	Version Control System
FR	Function Requirement
NFR	Non Function Requirement
SRT	Soft Real Time
HRT	Hard Real Time

ABSTRACT

Software Configuration Management for Embedded and Real Time Software in Software Engineering Domain

By

Hanan Hamdan Ahmad AL-Zawahreh

Supervisor

Dr. Khalid Torki Al-Sarayreh

(Assistant Professor)

Co-Supervisor

Dr. Khalid Al-Makademeh

(Assistant Professor)

Nowadays, embedded and real time software produced as product families. Products in which have an amount in public, nevertheless all the same vary to some extent in one or more features. Developing and controlling these product families is a sophisticated mission. Software configuration management (SCM) is widely ranging sustenance the improvement and development of one single software product and correspondingly supports the concept of alternatives solutions. Furthermore, in software engineering, SCM is charge with adjusting and tracing changes in the embedded and real time software and it is concerned with managing and controlling all the changes during software development cycle.

In this research study, it would be interesting to explore to the previous and proposed software configuration management frameworks (SCMF) and their elements in the academia and used by industry. In the previous works, SCMF formerly has solutions to some of the problems of software product and these SCMF formulate new techniques to support software changeability, controllability and management.

This thesis proposed generic steps for the software configuration management that could be used as a guided framework in the work product environment. These generic steps or what so called SCMF is built on the basis of previous and published configuration frameworks in literature to manage and

controlling the software development process of embedded and real time software.

The proposed SCMF in this thesis is to keep the changes and tracking it in software product during software development life cycle (SDLC). In order to develop and maintain quality software products. Moreover, the proposed and generic steps present configuration identification for all items of the previous works. Before it explains how to apply the change management and implement all the changes. The proposed generis steps in this thesis evaluated in automated software system, where we show the simple RICE COOKER prototype, and present the new enhancement, which we introduce to obtain new version of RICE COOKER throughout our proposed generic steps.

CHAPTER ONE
INTRODUCTION

Currently, software products intended to grow exponentially in market. Thus, the complexity of software has a huge increment. This makes a big challenge for the software development. That leads the strongest companies to force the software engineers to develop their product in minimal time with limited size and high quality while maintaining the same customer's requirements. Therefore, we must impose more control on software product development to avoid failure.

The normal Software Configuration Management (SCM) considered as service function in software development process (SDLC). Where the SDLC is a continuous process of change, due to many reasons, changes in requirements, schedule, budget, design programs and others [1]. SCM is interested in managing the evolution of software systems during the initial stages and during all other phases in SDLC. That means when made modify in embedded software must be efficiently control. In order to ensure that the modifications do not turn the process into chaos. Therefore, the modifications controlled by a software engineering activity called SCM [2].

This introductory chapter presents an overview of the problem statement of this thesis. It discusses the importance of introducing more research effort in order to improve framework to develop the embedded and real time software using SCM .Moreover it also discusses the objectives that achieved throughout this thesis. Finally, this chapter concludes with the chapters' organization throughout this thesis.

1.1 Problem Statement

Software configuration management (SCM) is a special case of configuration management (CM). Where software is faster and easier to change more than hardware. SCM can deal with some problems that related with the evolution of software and lack of control. So when developing the embedded system we face some challenges. Risks from malfunctions of embedded software are much higher than the application software.

There are many frameworks deal with evolution of embedded software. However, these frameworks have some weakness such as their development cycle is not clear enough, other types used in specific domain and they developed based on using other tools and frameworks to support them. Our framework built to overcome this weakness.

Based on the above, this research studies proposed a new and generic steps of SCMF to develop the embedded and real time software's and their role in increasing the efficiency, reliability, and traceability of software product in industrial sector.

1.2 Importance of Thesis

Many organizations deal with complex embedded and real time software. But it face some problems when deciding to improve these types of software, that return to lack of controlling the changes which introduce on the ES through development phase, where changing in ES without controlling and tracking can lead into chaos and losses in software product. Thus, this thesis is to contribute the improvement of SCM .Regarding developing the embedded and real time software to insure the process goes correctly, by the adoption of the SCMF to identify the configuration items and components. Then place them under configuration management in order to control the changes.

1.3 Thesis Objectives

The study aims to achieve the following objectives:

1. Investigate the Software Configuration Management Frameworks of embedded software from previous work proposed in literature.
2. Proposed a generic steps on the basis of the previous and explored frameworks
3. Manage the change and development of embedded and real time software system during software life cycle.

1.4 Thesis Organization

This thesis consists of seven chapters. The first chapter discusses the current problem statement, which is the lack of control changes in embedded software when developing it. **Chapter 2** presents the main motivation and the objectives of this thesis; it also presents a detailed methodology that followed to carry out the objectives of this thesis. Furthermore, **Chapter** 3 shows an overview of the previous work of

SCM in embedded and real time software. **Chapter** 4 shows the requirements of business domain of embedded and real time software. **Chapter 5** relies on Chapters 3 and 4, where this chapter proposes a software configuration management SCMF and its elements to support the development process for the new or enhanced products of the embedded and real time software. **Chapter 6** introduces a case study using the proposed SCMF in order to validate the controlling and tracking changes through developing the embedded and real time software. Finally, a brief conclusion and future work pointed out in **Chapter 7**.

CHAPTER TWO
RESEARCH METHODOLOGY

This chapter aims to discuss the main motivation behind conducting this research work and discusses the detailed research methodology that followed to scope the thesis objectives as well as organizes the research process in order to ensure validity of this thesis. It discusses the research motivation, goals, objectives and a detailed methodology to achieve this research. Moreover, this research methodology contains four phases as follows to extend the research objectives.

2.1 Overview of Research Methodology

This part describes an overview of the research methodology that used to achieve objectives. Moreover, it provides clear steps to demonstrate how the thesis carried out, and it consists of four phases as seen in Figure 1.

Phase1: SCM and SCMF in the literature Review.

This phase includes surveying the literature on the previous and published frameworks of SCM from academia and the industry.as well as it presents an overview of the earlier and recent studies in this domain. For more details. See chapter 3.

Phase2: Analyzing the Requirements of ES in Business Domain and Explain the SCRUM Process

This phase analyzed the identified steps that guided to build the generic SCMF and their elements. It divided into two parts:

1) Defines and analyze the requirements of embedded software in business domain.
2) Shows how the software developed and changed in the organization requirements by using the SCRUM process .In addition this chapter includes the SCRUM principle and activities. See chapter 4.

Phase 3: Software Configuration Management Framework (SCMF)

Is a construction phase. After analyzing the SCM and ES requirements in the previous phases, then propose SCMF based on previous frameworks to support the development process of embedded and real time software through specific steps. See chapter 5.

Phase 4: Case Study

Applying Software Configuration Management Framework SCMF to validate this solution by the case study in RICE COOKER. (See chapter 6).

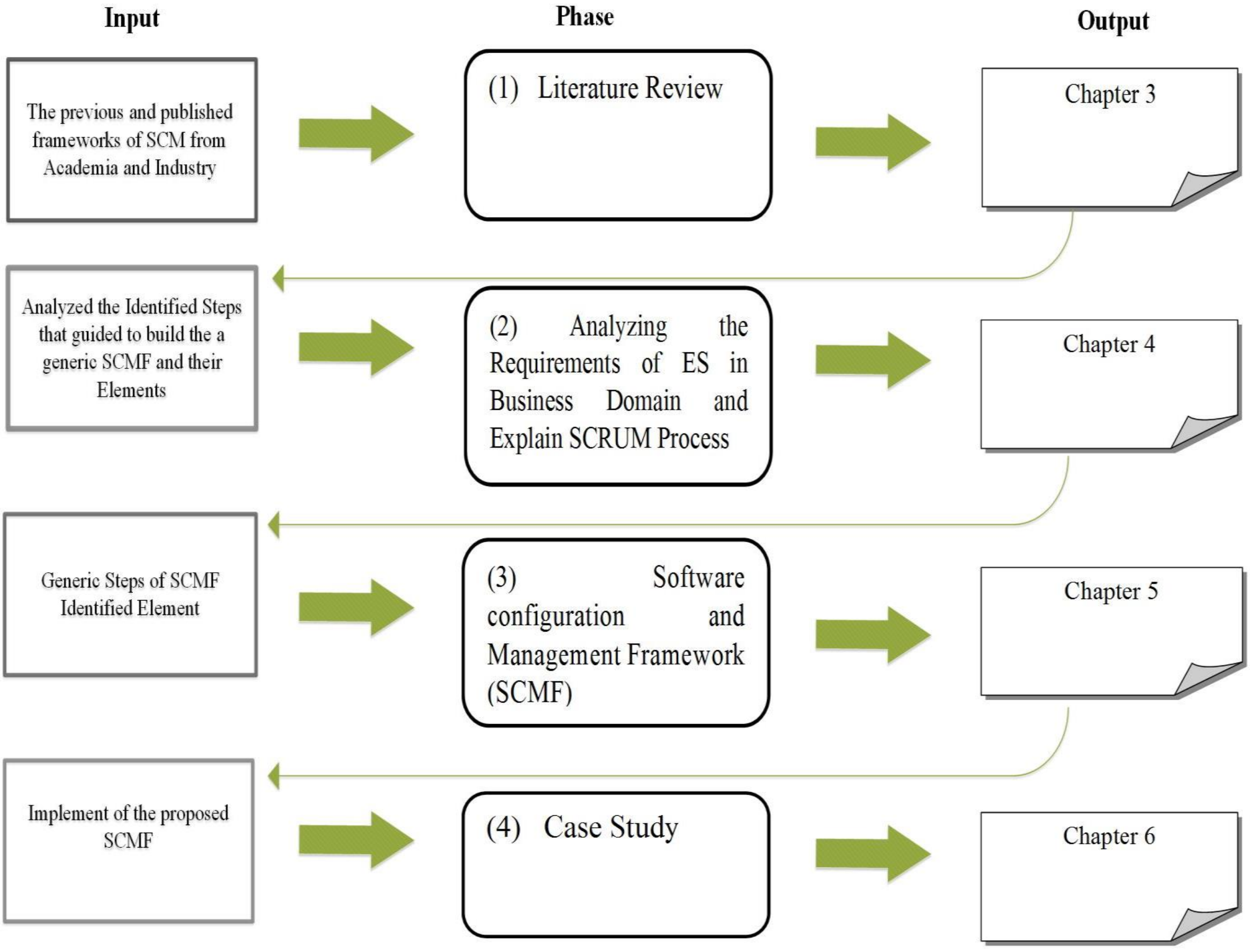
Input
Phase
Output
The previous and published frameworks of SCM from Academia and Industry
(1) Literature Review
Chapter 3
Analyzed the Identified Steps that guided to build the a generic SCMF and their Elements
(2) Analyzing the Requirements of ES in Business Domain and Explain SCRUM Process
Chapter 4
Generic Steps of SCMF Identified Element
(3) Software configuration and Management Framework (SCMF)
Chapter 5
Implement of the proposed SCMF
(4) Case Study
Chapter 6

Figure 1: Research Methodology

CHAPTER THREE

LITERATURE REVIEW

Currently, software products are becoming more complex and larger than before. In the same time during the development phase, we make some changes to develop it. In addition, any change that occurred in the environment causes change in the software requirements. Moreover, customers demand a product with high quality in minimal time. Thus, software changes are unavoidable in software development products. Where each change occurred during the development phase can lead to a better quality, or may lead to failure if the software of product development did not put under good control.

Software development process aims to obtain good embedded software with high level of quality. However, the main issues and challenges that face the developers are how they can control these changes during software lifecycle. Therefore, to ensure these changes do not return to the chaos, we can use configuration management (CM). The essential purpose of software engineering is to obtain the highest level of the improvement in an easy way to introduce new changes to software.

CM is an essential task to develop the complex embedded software. It is a management task. Moreover, it provides administrative and technical guidelines for the lifecycle of embedded software.CM needs some systems to support its process such SCM. SCM is a special case of CM [3]. In addition, SCM provides a way for identifying, tracking, and controlling the version of each item in embedded software.

SCM considered as a method for controlling the changes of embedded and real-time software during the life cycle. The essential activity in SCM is version control and change management. Change management defined as "the process of continually renewing an organization’s direction, structure, and capabilities to serve the ever changing needs of external and internal customers" [4]. Version control is defined as" the assignment of either unique version names or unique version numbers to unique states of software configuration items, usually for a specific purpose, such as a release of the software product to an external group or the identification of a specific baseline"[5].

Embedded systems developers face some problems when dealing with different versions of software. For this reason, the developers tend to use the SCM to manage large files.

Embedded software changed through the development process. Referring to many reasons one of them return to the stakeholder, He/she demands updates of data, service delivered, or functionality. Sometimes the scheduling constraints or budget reasons redefine the system. In addition, the business reorganization causes changes in team formation. Furthermore, the market conditions or new businesses impose changing the requirements or rules of the business [6]. In software, development process has three main classifications applied on the object: controlled, pre-controlled and uncontrolled [7].

Controlled object its configuration control. Software configuration identification (SCI) determines items to be controlled. Because it is impossible to control anything, which is not, known. That means breaking the system down into a number of parts. Then, make relations among the SCI to facilitate the traceability of these relations. After that grouping similar configuration items into the baseline. It, which separated between different versions. Then, understand the status of each item through the development process [7].

The electronic control units (ECU) in automotive industry have integration between hardware and software. ECU with Driver Assistance systems (DAS) can provide control and safety performance. However, the main problem at ECU was in HW/SW. Which cannot make any change at the run time. This contrasts with coming days; in future, the DAS distributed over the all automobile. Thus, the automobile needs to interact the components with each other's .in order to obtain high level of performance, provide ability to change the hardware and software configuration, and allow the incorporation among components at the run time. To get all of these benefits the interface layer for ECU such as radar and video, was designed [8].

Real-time system identification tool (RealSysId). Considered a computational tool. It provides the ability to select and compute through a real time of flight. The main purpose of this tool is developing the selected coefficients online then identifying it and validating all these online. It was the first version created in the aircraft domain.

Moreover, this tool has the ability to select and compute through real time with analytical and visual indications by using some methods. While the most of aircraft systems define all procedures by using pre-planned flight tests offline [9].Recently the cyber-physical system (CPS) is an embedded system. It used to monitor the physical processes .CPS used for multiple tasks in various environments where different constraints and rules need to maintain the system and it often leads to develop several of products variant. It can increase the efficiency and the effectiveness of the development of variability by using the product line engineering (PLE) [10].

PLE tools and methods provide the ability to reuse the existing software. Evolution management is the critical thing for CPS. It achieved by software configuration management (SCM). It focuses on changing control to keep traceability and integrity. So, it was needed the version control. Thus, the combinations between SCM with PLE increase the level of controlling and managing the software evolution. The main point behind this combination is version control to define what and where the change was occurred [10].Ship Power System is a full and independent system. In addition, it demands a high level of quality, especially run time to develop the power quality of ship system using the combination of techniques as configuration software and virtual instrument technology [11].

The electric vehicle spreads in the global market. However, it still faces some challenges. The big challenge is the short of autonomy, where, this vehicle can deal with limited hundred kilometers. The embedded energy software in these vehicles put it under monitoring. Therefore, it used Energy Management System (EMS) [5].EMS has a high level of controlling with a high quality of managing. Therefore, each hybrid and full vehicles use EMS in order to provide the best vehicle efficiency. Other researchers tend to focus on a prediction or the trip knowledge to reach the optimal battery and engine. However, the lasting research interested in embedded software to obtain global Quality of Service through used some Quality Assurance (QA) frameworks [5].

QA framework provides two different ways effectively configure electric vehicle of EMS: on-line and off-line, where, the off-line uses to define the characteristics of the vehicle and then matching it with their abilities. The on-line is the execution of

QA framework. It used to define which driving strategy to choose the best from alternatives. The two ways lead to minimize the computation time that needed to choose the optimal solution from space on-line. QA framework provides Quality of Service as much as possible in addition provide EMS, which has global, features [5].

Industrial embedded systems have long lifetime also demand a high level of reliability and safety requirements, where HW/SW exposed to several changes. So that, it must control these changes over the lifecycle of the system. Embedded software is a critical part; it plays the main role to distinguish a product from another. The dilemma of industrial embedded systems is how to manage and control the changes. Several tools and models used to manage the development process for Programmable Control and Communication Platform (PCCP), which built through Linux. In addition, SCM used in this kind of machine to identify and control software configurations. It provides many benefits to a product such the ability to trace changes, increasing productivity, and more safety [12].

Configuration Management (CM) is a combination of techniques, which coordinate and control with each other to construct the system. Which was developing the CM principles to enable the hardware engineers design and assemble the components [13]? Many authors explain the SCM through many of definitions. The first definition is "SCM, like CM, is defined as the discipline of identifying the configuration of a system at discrete points in time for purposes of systematically controlling changes to this configuration and maintaining the integrity and traceability of this configuration throughout the system life cycle"[14].

According to IEEE standards, the SCM Is " Configuration Management is a discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements"[15].
SCM is a development process with some of standards and procedures to provide a good managing of an evolving the software system. In addition, SCM considers as a Software quality assurance activity that applied during the software process [16]. Thus, SCM is a collection of activities integrated with each other's to manage and

control every change throughout the life cycle of computer software. SCM is an integral part of software quality assurance SQA. It participates to assess the impact of the changes that occurs during SQA activities and defined decisions based on cost and benefit analysis. In addition, SCM is a helping software life cycle process. It used mostly in the development and maintenance of the organization [17] [18].

Customers need to change their requirements at any phases in the life cycle. So these changes addressed, for this reason, organization tend to use SCM to ensure these changes approved or disapproved. The SCM activities divided into two main parts: planning and controlling of the SCM process. Traditionally these activities grouped as series of tasks. These tasks have many functions, software configuration identification (SCI), configuration control (CC), configuration status accounting (CSA), and configuration auditing (CA) [19].

The good software configuration management plan (SCMP) for controlling changes can lead to avoid the chaos and unexpected results. So, to success, SCM must analyze, manage and plan all requirements correctly. Otherwise, the software project fails. SCM is a critical element in software engineering because it must keep the high quality of product and continue to improve the productivity of software through the lifecycle, which achieved through organized and planned change management. Moreover, it can include the change management in SCM .In order to obtain high stability, reliability, better quality of time and provide the way to detect the bug and tracking it. In addition, SCM known as an inseparable part of quality oriented [20] [21].

Risk management (RM) is imperative to success software project. Because it was used to catch the modification and the roles, responsibility, controlling, and tracking. It defines each risk and classify it in risk log. Therefore, we use SCM to manage all potential risks. The first step in SCM Repository is to save all risks in the risk log then stored it in a specific way to determine which can occur before the other. In addition, it provides the ability to track for each risk by new members; this applied in the spiral model. Each risk recorded in a repository. The previous data used as input to the next step while File versioning considered as a major risk under SCM [22].

Software product line engineering (SPLE) is a technique for developing all types of software products in the specific domain. SPLE does not have SCM tool to improve their evolution; SCM tool designed to improve a single product. Therefore, the researcher suggests a new prototype of configuration management called Molhado SPL. It designed to solve the evolution process in SPL [23]. When developing a software it may occur many changes. These changes put into controlling and tracking in order to achieve a high level of quality and improvement of software productivity. Therefore, they use SCM in order to obtain global tractability for configuration management and individual artifact's changes through the combination of individual working environment and configuration management system. Thus, traceability can be control more effectively [24].

The essential feature in SCM is a version control (VC). It is a stand- alone but in most cases, it embedded in different types of software.VC stores versions based on increasing updates rather than entire copies. In addition, it has the ability to retrieve the previous version during debugging and testing phase. In addition, it provides tracking the document to correct the error by editors. All versions stored in a repository. While repository must have the ability to manage, all object types such text, bitmaps, and complicated documents. All of these is to enable effective controlling and management of software product [25].

VC is divided into two types based on the environment are Distributed version control systems (DVCS) and centralized version control systems (CVCS) such as current version control (CVS) and subversion (SVN). However, DVCS have a more popularity than CVCS [25]. Furthermore, VCS an essential role in the tracking system and releases management. VCS also known as revision control. It provided historical views of source code and all changes on it; so VCS named a source control. There are various kinds of VCS such as git, mercurial, bazaar, CVS and SVN, these types aim to develop the merging and branching capabilities and provide more secure to repository [26].
SCM is a discipline of software engineering where SCM consists of procedures and technical. It used to manage the changing of software product. In addition, it considered an umbrella activity because it applied through the software development

lifecycle. In addition, the modification can occur in any phase. Every system in software engineering domain contains many components. These components integrated with each other to success specific goals of the system. Thus, SCM is a system contains many items; it used in some organizations to identify each element then tracking it to extract any changes by change control. In addition, it can verify any modification by using some tools that used in versioning .in order to follow bugs and maintain the right changes.

CHAPTER SEVEN

CONCLUSION AND FUTURE WORK

7.1 Conclusion

This thesis contributes in the field of development of embedded and real time software. Our main goal is to manage and control the changes in ES and RTS during SDLC. Several studies interested in this field.

The development of good quality software is a sensitive element of successful competition for current software market. Especially, when we develop embedded and real-time software because they are becoming more complex and larger. The changes in embedded and real time software are inevitable. Therefore, we must use mechanism for controlling it.

Most of the previous research on this topic have some challenges.so in this research work, proposed generic steps used as guided framework when developing the product.Where the SCM helps the software developers for identifying items, then controlling and ensuring the authorized modification made to an item.

We created proposed generic steps on the basis of the previous and published frameworks of SCM from Academia and industry. In order to support the development process for new or enhanced products of ES and RTS. In addition, it has the ability to define the critical requirement for embedded and real-time software based on the business domain. Moreover, SCMF use for evaluating, coordinating, disapproving or approving and implementing changes in artefacts of ES.

Thus the SCMF consists of specific elements that identify items, change management, version control, track change requests and control changes. These activate connection together to control the modifications of the last version of software product to produce new version of it.

7.2 Threats of Validity

Although the thesis has reached its objective, there were some internal and external threats. This section discusses the validity of the SCMF.

Internal threat: The SCMF presented previous applied and evaluated in electric technology as a RICE COOKER**.** It demanded understand all the SCMF elements.

External threat: Regarding the case study, the proposed recommendations and solution have based on the applied on one case study. In addition, SCMF required quantitative measurements in order to obtain high level of accuracy.

7.3 Thesis Contributions:

The research contributions of this thesis are:

1- Identifying and analyzing the requirements of embedded and real time software from business domain (critical requirements).

2- Proposing a generic steps for SCMF. It is a descriptive model supporting the analysis and understanding the SCM practices.

3- SCMF manage the change and develop of large embedded and real time software during software life cycle.

7.4 Future Work

Our future direction is to improve our framework through add more steps. In order to gain the better level of controlling changes and allow introduce new features in a flexible way. Also, we are planning to combine it with empirical studies. Moreover, we need to apply other types of embedded software to validate our framework. In addition, we aim to produce tool platform that supports our framework.

العنوان:	Software Configuration Management for Embedded and Real Time Software in Software Engineering Domain
المؤلف الرئيسي:	الزواهرة، حنان حمدان أحمد
مؤلفين آخرين:	الصرايرة، خالد تركي، المخادمة، خالد عبدالكريم(مشرف)
التاريخ الميلادي:	2017
موقع:	الزرقاء
الصفحات:	1 - 52
رقم MD:	856632
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	عمادة البحث العلمي والدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	هندسة البرمجيات، برمجيات الزمن الحقيقي، إدارة تكوين البرمجيات، برمجيات الأنظمة المضمنة
رابط:	https://search.mandumah.com/Record/856632

REFERENCE

[1] Ren, Y., Xing, T., Quan, Q., & Zhao, Y." Software configuration management of version control study based on baseline. ", In Information Management, Innovation Management and Industrial Engineering (ICIII), 2010 International Conference on (4) 118-121. IEEE. 2010.

[2] Bartusevics, A., & Novickis, L. "Towards the Model-driven Software Configuration Management Process". Information Technology and Management Science, 17(1), 32-38.2014.

[3] Calhau, R. F., & de Almeida Falbo, R, "A Configuration Management task ontology for semantic integration". In Proceedings of the 27th Annual ACM Symposium on Applied Computing 348-353. ACM. 2012.

[4] Moran, J. W., & Brightman, B. K," Leading organizational change", Journal of Workplace Learning, 12(2), 66-74. 2000.

[5] Tchakaloff, B., Saudrais, S., & Babau, J. P. "Efficient models configuration for an electric vehicle energy management software ". In 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications 48-55. IEEE. 2014

[6] Saeed, S. Embedded and Real Time System Development: A Software Engineering Perspective. 1st edition 2014, Springer-Verlag Berlin Heidelberg.

[7] Mall, R. Fundamentals of software engineering. 4th edition 2014. PHI Learning Pvt. Ltd.

[8] George, K. S., & Anitha, G. S. "Design of an interface layer for automotive video and radar ECU," In Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMD), 2014 Annual International Conference on 1-5. IEEE.2014.

[9] Zanette, J. V., & de Almeida, F. A, "RealSysId: A software tool for real-time aircraft model structure selection and parameter estimation" .Aerospace Science

and Technology (54) 302-311.2016.

[10] Hellebrand, R., Schulze, M., & Becker, M. "A branching model for variability-affected cyber-physical systems". In Emerging Ideas and Trends in Engineering of Cyber-Physical Systems (EITEC), 2016 3rd International Workshop on 47-52. IEEE.2016.

[11] Ouyang, H., & Li, H. "The ship power quality monitoring system based on virtual instrument and configuration software", In Chinese Automation Congress (CAC), 2015, 1646-1649 IEEE. 2015.

[12] Leppäkoski, A., Salminen, E., & Hämäläinen, T. D, "Framework for industrial embedded system product development and management". In System on Chip (SoC), 2013 International Symposium on1-6. IEEE. 2013.

[13] Whitgift, D. Methods and tools for software configuration management. J. Wiley.1st edition 1991.

[14] Bersoff, E., Henderson, V. D., & Siegel, S. G. " Software configuration management ". An investment in product integrity. Englewood Cliffs: Prentice Hall, 1980.1980.

[15] Radatz, J., Geraci, A., & Katki, F." IEEE standard glossary of software engineering terminology ". IEEE Std, 610121990(121990), 3. 1990.

[16] Bach, J. "The highs and lows of change control". Computer, 31(8), 113-115. 1998.

[17] Bersoff, E., Henderson, V. D., & Siegel, S. G,"Software configuration management". An investment in product integrity. Englewood Cliffs: Prentice Hall, 1980. 1980.

[18] Al-Shaikhly, Y. S, "Proposed Model For Software Configuration Management Theoretical Tool ". Doctoral dissertation, Middle East University.2011.

[19] Pressman, R. Software engineering: a practitioner's approach. 8 edition 2014. Palgrave Macmillan.

[20] Bendix, L., Kojo, T., & Magnusson, "Software Configuration Management Issues with Industrial Opensourcing". In 2011 IEEE Sixth International

Conference on Global Software Engineering Workshop 85-89. IEEE. 2011.

[21] Deepali, T. K. S. B, "Configuration Management in Software Development Life Cycle". An International Journal of Engineering Sciences, 4. 2014.

[22] Rao, N. R., & Sekharaiah, K. C," An Incremental Risk Management Framework for Realizing Project Efficiency using Version Control". CCSN and IJCA India, 1-6.2013.

[23] Thao, Cheng, "A Configuration Management System for Software Product Lines". Theses and Dissertations.2012.

[24] Kim, D. Y., & Youn, C, "Traceability Enhancement Technique through the integration of software configuration management and individual working environment", In Secure Software Integration and Reliability Improvement (SSIRI), 2010 Fourth International Conference on 163-172. IEEE. 2010.

[25] Brindescu, C., Codoban, M., Shmarkatiuk, S., & Dig, D, "How do centralized and distributed version control systems impact software changes? ",In Proceedings of the 36th International Conference on Software Engineering 322-333. ACM. 2014.

[26] Rao, N. R., & Sekharaiah, K. C, " Embedding version tag in software file deliverables before build release", In Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2015 4th International Conference on 1-6. IEEE. 2015

[27] Axelsson, J., Papatheocharous, E., & Andersson, J. "Characteristics of software ecosystems for Federated Embedded Systems: A case study". Information and Software Technology, 56(11), 1457-1475. 2014.

[28] Eklund, U., & Bosch, J. "Architecture for embedded open software ecosystems". Journal of Systems and Software, 92, 128-142. 2014.

[29] Pohl, K." Requirements engineering: fundamentals, principles, and techniques". Springer Publishing Company, Incorporated. 2010.

[30] Jäälinoja, J," Requirements implementation in embedded software development" .University of Oulu, Department of Information Processing Science.85.2004.

[31] Carlshamre, P., & Regnell, B," Requirements lifecycle management and release planning in market-driven requirements engineering processes". In Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on 961-965. IEEE. 2000.

[32] Helmy, W., Kamel, A., & Hegazy, O." Requirements engineering methodology in agile environment". International Journal of Computer Science Issues, 9(5). 2012.

[33] Wiegers, K., & Beatty, J. Software requirements. 3rd edition.2013. Pearson Education.

[34] Egyed, A."A scenario-driven approach to trace dependency analysis". IEEE Transactions on Software Engineering, 29(2), 116-132.2003.

[35] Wasserman, A. I. "Software engineering issues for mobile application development". In Proceedings of the FSE/SDP workshop on Future of software engineering research 397-400. ACM.2010.

[36] Vera-Baquero, A., Colomo-Palacios, R., & Molloy, O. Real-time business activity monitoring and analysis of process performance on big-data domains. Telematics and Informatics, 33(3), 793-807.2016.

[37] Al-Zawahreh, H., & Almakadmeh, K. "Procedural Model of Requirements Elicitation Techniques." In Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication 65. ACM.2015.

[38] Shen, M., Yang, W., Rong, G., & Shao, D. "Applying agile methods to embedded software development: A systematic review". In Software Engineering for Embedded Systems (SEES), 2012 2nd International Workshop on 30-36. IEEE.2012.

[39] Srinivasan, J., Dobrin, R., & Lundqvist, K. "State of the Art'in Using Agile Methods for Embedded Systems Development". In 2009 33rd Annual IEEE International Computer Software and Applications Conference on (2), 522-527. IEEE.2009.

[40] zur Muehlen, M., & Shapiro, R. "Business process analytics". In Handbook on Business Process Management 2. 137-157. Springer Berlin Heidelberg.2010.

[41] Schön, E. M., Thomaschewski, J., & Escalona, M. J. "Agile Requirements Engineering: A systematic literature review". Computer Standards &Interfaces, 49, 79-91.2017.

[42] Rising, L., & Janoff, N. S. "The Scrum software development process for small teams". IEEE software, 17(4), 26. 2000.

[43] Lucia, A. D., & Qusef, A. "Requirements engineering in agile software development". Journal of Emerging Technologies in Web Intelligence, 2(3), 212-220. 2010.

[44] Ju, R., Huertas, C., Jim480, S., & Guerra-Garc, C. "Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review". In 2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT) 141-148. IEEE.2016.

[45] Moe, N. B., Dingsøyr, T., & Dybå, T. "A teamwork model for understanding an agile team: A case study of a Scrum project". Information and Software Technology, 52(5), -491.2010.

[46] Eloranta, V. P., Koskimies, K., & Mikkonen, T. "Exploring ScrumBut—An empirical study of Scrum anti-patterns". Information and Software Technology, 74, 194-203.2016.

[47] Koskela, J." Software configuration management in agile methods". VTT publications.2003.

[48] Hossain, E., Babar, M. A., & Paik, H. Y. " Using scrum in global software development: a systematic literature review ". In 2009 Fourth IEEE. 2009.

[49] Guang-yong, H. "Study and practice of import Scrum agile software development". In Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on 217-220. IEEE.2011.

[50] Vlietland, J., van Solingen, R., & van Vliet, H. " Aligning codependent Scrum

teams to enable fast business value delivery: A governance framework and set of intervention actions". Journal of Systems and Software, 113, 418-429.2016.

[51] Tsaramirsis, G., & Basahel, A. "Scrum to manage humans and intelligent systems". In Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on 1353-1356. IEEE.2016.

[52] Paasivaara, M., & Lassenius, C. "Scaling Scrum in a Large Globally Distributed Organization: A Case Study". In Global Software Engineering (ICGSE), 2016 IEEE 11th International Conference on 74-83. IEEE.2016.

[53] Vlaanderen, K., Jansen, S., Brinkkemper, S., & Jaspers, E. "The agile requirements refinery: Applying SCRUM principles to software product management". Information and software technology, 53(1), 58-70.2011.

[54] Werner, L., Arcamone, D., & Ross, B. "Using Scrum in a quarter-length undergraduate software engineering course". Journal of Computing Sciences in Colleges, 27(4), 140-50.2012.

[55] Wallace, C., Mohan, S., Troy, D., & Hoffman, M. E. " Scrum across the CS/SE curricula: a retrospective". In Proceedings of the 43rd ACM technical symposium on Computer Science Education 5-6. ACM.2012.

[56] Jha, M. M., Vilardell, R. M. F., & Narayan, J. "Scaling Agile Scrum Software Development: Providing Agility and Quality to Platform Development by Reducing Time to Market". In Global Software Engineering (ICGSE), 2016 IEEE 11th International Conference on 84-88. IEEE.2016.

[57] Quaglia, E. J., & Tocantins, C. A. "Simulation projects management using Scrum". In Proceedings of the 2011 Winter Simulation Conference (WSC) 3421-3430. IEEE.2011.

[58] Pauly, D., Michalik, B., & Basten, D. "Do Daily Scrums Have to Take Place Each Day? A Case Study of Customized Scrum Principles at an E-Commerce Company". In System Sciences (HICSS), 2015 48th Hawaii International Conference on 5074-5083. IEEE. 2015.

[59] Nejmeh, B., & Weaver, D. S. "Leveraging scrum principles in collaborative, inter-disciplinary service-learning project courses". In 2014 IEEE Frontiers in Education Conference (FIE) Proceedings 1-6. IEEE. 2014.

[60] Schwaber, K. "Scrum development process". In Business Object Design and Implementation 117-134. Springer London.1997.

[61] Pino, F. J., Pedreira, O., García, F., Luaces, M. R., & Piattini, M. "Using Scrum to guide the execution of software process improvement in small organizations". Journal of systems and software, 83(10), 1662-1677.2010.

[62] Bourque, P., & Fairley, R. E. Guide to the software engineering body of knowledge (SWEBOK (R)).3th edition 2014.IEEE Computer Society Press.

Software Configuration Management for Embedded and Real Time Software in Software Engineering Domain

By

Hanan Hamdan Ahmad Al-Zawahreh

Dr. Khalid Torki Al-Sarayreh

(Assistant Professor)

Co-Supervisor

Dr. Khalid Al-Makademeh

Submitted in Partial Fulfillment of the requirements for the degree of Master in Software Engineering

Deanship of Scientific Research and Graduate Studies at the Hashemite University

Zarqa – Jordan

16/04/2017