Software measurement metrics in project scope management

Tek Prasad Gautam

University of Tampere
School of Information Sciences
Computer Science
M.Sc. thesis
Supervisors: Zheying Zhang
Timo Poranen
March 2014

University of Tampere
School of Information Sciences
Computer Science
Tek Prasad Gautam: Software measurement metrics in project scope management
M.Sc. thesis, 62 pages, 12 index and appendix pages
March 2014

Every year software project failure costs huge sums of money, primarily due to the lack of appropriate project scope management (PSM). One of the major failure factors is the result of uncertain expansion or change in project scope, also known as scope creeping (SC). Furthermore, SC directly affects the project's budget, schedule, and finally the project quality. Studies have shown that the main causes of SC are ambiguous project scope, vague and incomplete requirements, and a lack of change control management. In addition to these, there are several other causes resulting in SC. As the causes change and emerge differently in different projects, not all of these could be determined in a single project. The main objectives of this thesis is to make an in-depth study on existing causes; explore several academic projects for any new causes; and establish the degree of impact by these causes in development process. Furthermore, it suggests software measurement metrics that support to minimize SC.

To fulfill the aim of this thesis, a quantitative research methodology was chosen. A group of students who worked as project managers in various student projects, in a course conducted by the University of Tampere were, requested to fill in a set questionnaire. The collected data from this case study were then coherently analyzed and statistically compared with the data from past research to determine additional causes of SC.

This thesis work identified additional causes such as insufficient resource allocation, lack of end-user involvement, ineffective communication, a change in customer needs, platform changes, and the addition of extra features as the major causes of SC. On the other hand, it suggested three metrics: balanced scorecard (BS), earned value management (EVM) and requirement metrics that support the minimization of SC. These metrics can minimize SC by improving the resources allocation, user involvement, and communication. These metrics can also handle the requirements and platform change request from the stakeholders.

Keyword: Scope Creeping, project management, metric, requirement, stakeholder.

Acknowledgements

I would like to express my deepest gratitude to my supervisors Assistant Professor Zheying Zhang and Lecturer Dr. Timo Poranen for their encouragement, guidance and valuable comments. I would like to thank all the participant project managers from the case study without whom it would have been impossible to complete this work.

I would like to thank Professor Jyrki Nummenmaa and Assistant Professor Eleni Berki as well as all the attendee of the seminar course entitled "Master's Thesis Seminar in Software Development" for their suggestions and feedback. I would also like to thank, Sunil Chaudhary, who supported me and give his best suggestions throughout this thesis. It would be an injustice if I failed to thank my dear friends, colleagues, and teachers for their support, suggestions and feedback.

Finally, I would also like to express my heartfelt appreciation to my family and my beloved wife, Nisha Bhatta, for her continued support and motivation throughout this thesis.

Tek Prasad Gautam
20 March 2014, Tampere

Contents

1. Introduction

1.1. Overview

A software project is said to be successful if it is completed on time, on budget and with feature/functionalities as initially specified. The different research conducted in the field of software has shown that it is rare to complete a project successfully within the estimated budget and on schedule. One such research was conducted by the Standish Group which is an American Company established in 1985. An outcome of this study of a collection of project cases, CHAOS manifesto, is published on a bi-yearly basis for software projects mostly from the United States (60%) and Europe (25%). The results from 2012 CHAOS [2013] showed that 39 percent of projects were successful (delivered on time, on budget with required functions and features), 43 percent of projects faced the different challenges during the development, such as late, over budget with less required features and functions and 18 percent were failed which means they were either cancelled prior to completion, or delivered and never used.

As mentioned earlier, time, budget and features/functionalities play a vital role to determine the resolution (success, failure, challenged) of a project. For example, even if a project meets the budget and schedule, there could still be a doubt whether the project delivery meets the final expectation, which can make the project 'challenged'. When any challenged project starts to show symptoms, the project manager needs to deal with these situations and look for solution(s) accordingly. In order to do so, the project manager should identify the actual cause. The most common causes for a challenged project includes ambiguous project scope, lack of user involvement, poor estimation of the schedule, vague and incomplete requirements, change in customer need, and the lack of proper change control management [Anthes, 1994; Kerzner, 2009]. The project manager must then monitor and control the presence of activities that increase the chance of a challenged/failed project.

"Project scope management is the process to ensure that a project includes all of the work required, and only the work required, to complete the project successfully" [PMI, 2009]. A proper practice of project scope management (PSM) can play an important role in minimizing scope creeping (SC). SC is an extra expansion in the initially defined project scope due to the change and/or addition in the requirements [Gurlen, 2003]. It is often found in projects with incomplete requirement sets and also in those practicing counter-effective change control management [Bronstein, 2010]. Software projects are

initiated with fixed schedule, cost and resources. If a proper PSM process is missing in a project, business objectives and requirements may change without a proper control. These changes can affect project schedule, cost and resources. Because of this, it is hard to ensure that the project can be conducted successfully.

On the other hand, project requirements do and can change and thus cannot be completely avoided during the software development lifecycle. This is due to several reasons such as change in customer needs, arrival of new technology, and end-user expectations, among others. However, the change implementation can be controlled and minimized by using software measurement metrics. In addition, a management plan is equally necessary during the software development lifecycle. It includes the collection of rules, methods and principles regarding how to control and handle changes such as requirement changes, environment changes and technology changes. There are different management plans prepared for a software development project. Some of the plans, such as scope baseline, change control management plan, requirement management plan, and configuration management plan, are suggested in this thesis for the purpose of supporting a good scope management process. These plans are useful in handling SC, but the question on how to identify the possibility of SC in a running project still need to be researched further.

Software measurement metrics are used for monitoring the progress of a project and understanding the potential risk of SC. Among a number of software measurement metrics the following metrics are recommended because they have different functionalities of monitoring and minimizing SC, they are, earned value management (EVM), balanced scorecard (BS), and requirement metrics. EVM focuses on cost and schedule change [Dwivedi, 2009]. Whilst, BS improves the development of system by determining the different parameters related to financial, customer, internal process development, and learning and growth perspective [Kaplan and Norton, 2007]. Requirement metrics, on the other hand, focus more on tracking and documenting the changes of requirement and relates those changes with the actual needs of the project. In addition, requirement metrics track and document the changes along with their causes and the entity responsible for the changes [Carlos, 2010]. With these facts, it can be said that with the proper preparation of management plans at the beginning of the project, along with an implementation of a suitable software measurement metrics, are supporting factors for minimizing SC in any software projects.

1.2. Research question

As shown in the overview, SC can be minimized by identifying and handling its proper causes. Furthermore, software measurement metrics supports PSM in handling and reducing the occurrence of SC. These factors lead towards two relevant research uncertainties that need to be explored and studied in more detail. Therefore, this thesis will investigate and answer the following two research questions:

(i) What are the causes of scope creeping and how do they influence the project plan and development?

(ii) How does software measurement metric support project scope management by handling and reducing the risk of scope creeping?

1.3. Thesis outline

The thesis is structured as follows. Chapter Two describes the PSM in detail. It includes various sections with overviews, definition of the PSM process and its importance. Chapter Three describes the SC and then it further introduces various causes of SC, their impact and measurement. Chapter Four describes the software measurement metrics with definitions and their individual role in SC. Chapter Five consists of the case study, including the environment of the case study, motivation of the case study, followed by its methodological orientations. Furthermore, it includes the data and its analysis. In addition, it includes the main challenging causes of the software project identified from case study, whilst bringing in the main causes of SC and relates them with existing ones. Chapter Six describes different management plans and metrics that support the minimization of SC. Chapter Seven presents the conclusions, limitations of this thesis work, and some possibilities for further research.

2. Project scope management

A project scope is a valuable piece of information that will guide the project in the right direction [Woods, 2012]. The scope helps to structure the time, resources, and budget associated with the project. This is helpful in keeping strict management around the project scope which will have a positive impact on outcome of the overall project. A project scope development process includes the following steps [PMI, 2009; Woods, 2012].

- *Brainstorming.* Collecting the project objectives by conducting the brainstorming session amongst the project's stakeholders.
- *Requirements.* After the brainstorming session, list all the requirements that have to be fulfilled during the project. In the majority of cases, all the ideas collected from the stakeholders cannot be included in the project, and hence clearly filter those ideas to get a proper set of requirements.
- *Deliverables.* Identify the deliverables for each project development phase and if possible link them to a specific milestone. Add new project personnel only after approval from stakeholders.
- *Costs.* Estimate the project cost accurately. If the estimation is too low, then the project will be in danger of going over budget, and if high, then it affects other running projects by having all the resources to itself.
- *Sign, seal and deliver.* Assemble all the necessary elements into one document. All the project stakeholders and owners need to agree with it and sign it.
- *Scope change management.* Determine and handle unexpected changed in cost and schedule. It managed the changes by the well-defined process by making the right decision at the right time.

An essential documentation produced in the initial phase of software project planning is known as scope statement [Woods, 2012; PMI, 2009]. It outlines expected project results, and terms and conditions. It also helps to manage the stakeholder's expectations from the very beginning of the project. Therefore, a scope statement is considered 'proper' only when it is acceptable for stakeholders. In fact, the quality of scope statement is one of the components which determine the fate of the project.

PSM comprises of the processes required to ensure that the project includes the entire work required, and only the work required, to complete the project successfully. It

is primarily concerned with defining and controlling what is included in the scope statement [PMI, 2009]. Typically, many ideas arise during the project development phases. However, stakeholders drop some of them due to their high resource expectations. In many situations, some of these ideas help to achieve a better result in the project. Therefore, scope planning and management is an essential factor for successful project management. In addition, scope management always remains valuable among other project management knowledge areas such as time management, cost management and risk management. The PSM triangle (Figure 1) shows that the expansion in the scope directly affects the time, quality and resources related to the project.

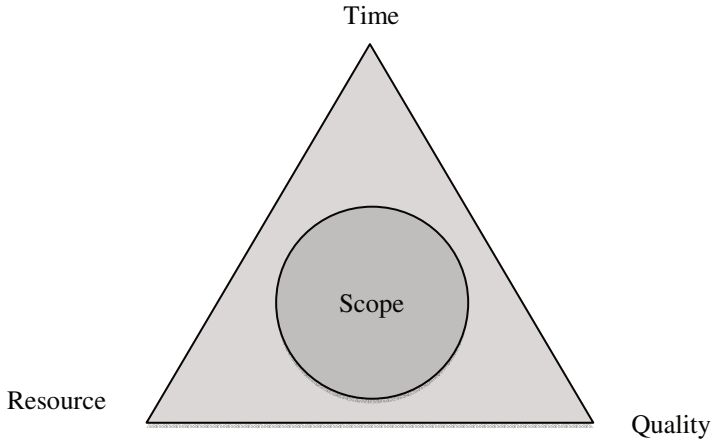

Figure 1: Project scope management [PMI, 2009]

2.1. A project scope management process

PSM includes five different processes: initiation, scope planning, scope definition, scope verification, and scope change control [PMI, 2009]. Different tools and techniques are used in these processes to produce desired outputs from the given inputs.

2.1.1. Initiation

Initiation is the process of formally recognizing the necessity of new product or continuing with the existing product into its next phases [PMI, 2009]. The initiation links the project and the performing organization. There are some factors such as market demands, business needs, customer requests, and technology changes, which lead in the initiation of the project. Organizations have their own strategies for a formal project initiation and many organizations formally initiate the project when the preliminary plan is prepared by conducting different feasibility studies [PMI, 2009].

The inputs of the initiation process are product description, performing organization strategic plan, project selection criteria and historical information. Product description is the detailed description of the characteristics of the product that has to be developed. Project selection criteria is also considered as input for initiation which are defined for the product by considering different factors such as financial return, customer perceptions and satisfaction, and market demand. [Conchúir, 2012] Besides, the past project decision selection result and performance are also considered as an input to the initiation process.

The tools and techniques implemented in the initiation process are project selection methods and expert judgments. Different methods such as comparative approaches, scoring models, benefit contribution or economic models fall, and mathematical models are considered as decision models for project selection [PMI, 2009]. Any group or individual who possesses good knowledge or training in the related field is considered an expert for the project initiation process. An expert's judgment must consider assessing the inputs to different process.

The outputs of the project initiation process are project charter, project manager, constraints, and assumptions. A project charter is a document that includes the business needs that have to address and the product description [Conchúir, 2012]. It provides authority to the project manager to utilize the organizational resources according to the necessity of the project activities. The project manager should be a dedicated and responsible person who will direct the project into the right direction by linking all the activities happening in the project. The constraints such as budget, technology, and many more, are identified at the beginning of the initiation process and are also considered as the output in the initiation process. [Sliger and Broderick, 2008] Assumptions are true, real or certain factors that are considered for planning purpose. More assumptions introduce more risk in the project.

2.1.2. Scope planning

Scope planning is the process of developing the scope statement in the written form that can be used as the basis for making a decision regarding the project in the future. This is important because the statement helps to determine the completion of various project phases as well as the project itself [PMI, 2009]. Scope planning is always considered as a basis for an agreement between the client and the project team because it determines the main objectives and goals of the project.

Product description, project charter, constraints, and assumption from the initiation process are considered as inputs to the scope planning process.

Product analysis, benefit and cost analysis, alternative identification and expert judgment are the tools and techniques to the scope planning. Product analysis techniques such as system engineering, value engineering, value analysis, function analysis, and quality function deployment are used for the better understanding of both product and the project [PMI, 2009]. On the other hand, benefits and cost analysis are used to determine the cost and benefits of the different project alternatives. Project alternatives are the different approaches to the project determined from the brainstorming and lateral thinking. [Sliger and Broderick, 2008] The expert ideas and experiences in the related field are also implemented in the scope planning process.

The outputs from the scope planning process are scope statement, supporting detail, and scope management plan. The scope statement is the collection of the documents, such as project justification, project product, project deliverables, and project objectives [Woods, 2012]. They are the basis for the project decision and developing the common understanding of project scope among the stakeholders. A supporting detail includes the well-organized documentation of all the identified assumptions and constraints, which facilitates other project management processes. A scope management plan describes how the scope will be managed and how the scope changes will be integrated into the project. [Schwalbe, 2007] It also clearly defines how scope changes will be identified, classified and integrated during the different project phases.

2.1.3. Scope definition

Scope definition involves dividing the main deliverables identified in the scope statement into a number of smaller manageable components. When the main deliverables are divided into small components, it helps to improve the accuracy of cost, time, and resources. In addition to that, it will help to define the baseline for performance measurement and control of the project [PMI, 2009]. A poor scope definition often forces stakeholders to increase the final expected cost of the project by increasing project time, disrupting project flow, causing rework, and ultimately lowering productivity.

Scope statement from the scope planning process, and constraints and assumptions from the initiation process are considered the inputs to the scope definition. Apart from that, historical information such as errors and omissions from past projects should also be included as an input to the scope definition [Sliger and Broderick, 2008].

The Work breakdown structure (WBS) templates and decomposition are the tools and techniques used in scope definition. WBS is a deliverable-oriented grouping of the project elements, which organizes and defines the total scope of the project [Kerzner, 2009]. Each project is unique and has its own product description. Even if the project is unique, the past project WBS could be used as a template for a new project. This reuse of the WBS helps in the project development by saving time and cost. Decomposition means dividing the whole project deliverables into smaller and more manageable components [Schwalbe, 2007]. Decomposition includes different steps, for example identifying the major elements of the project, estimating the adequate cost and duration for each element, identifying the constituent elements of deliverables, and verifying the correctness of the decomposition [Conchúir, 2012; Sliger and Broderick, 2008].

The WBS is the output from scope definition. It supports the development of common understanding of the project scope. It is generally assigned a unique identifier that is often called code of accounts. The items mentioned at the lowest level of the WBS can be considered as work packages, which can be further decomposed.

2.1.4. Scope verification

Scope verification is the process of accepting the project scope by the stakeholders. It is concerned with the acceptance of the work results. Scope verification reviews all the work results and ensure that all the products were accomplished perfectly and adequately.

The work results and product documentations are the two inputs for the scope verification process. The work results are the output of the project plan execution, which are fully or partially completed by the deliverables. Documents that were produced to describe the project's product for review such as plans, specifications, technical documentation, and drawing is logically named as product documentations [PMI, 2009].

Inspection is the technique used in scope verification process. Inspection means conducting the activities such as measuring, examining, and testing the product. From the inspection stakeholders ensure that all the products fulfill their requirements.

The acceptance documents are considered as the output(s) from the scope verification process, these documents are important as they ensure that the client finally accepts the product.

2.1.5. Scope change control

Scope change control is concerned with the influencing factors that create the change in scope and control the impact of those changes [Sliger and Broderick, 2008]. It also helps to ensure that the changes are beneficial. Furthermore, scope change control is also integrated with the other control processes such as time control, cost control, quality control and others [Conchúir, 2012].

The WBS is one of the most valuable inputs to the scope change control as it defines the project's scope baseline. In addition, the performance report is another valuable input to the scope change control. Performance report is the collection of the information that shows which parts of the product are completed and which are not. It also warns the project team by pointing towards the issues that can create a problem in the future [PMI, 2009]. The different internal and external factors that force to change the project scope are also considered as input to the scope change control. The changes request can expand or contract the project scope. The common reasons for the change request, are changes in government policies, error in defining scope, and change in technologies [Conchúir, 2012]. Besides, early-defined outputs of scope planning are also considered as an input to the scope change control process.

The tools and techniques used in scope change control are: scope change control system, performance measurement, and additional planning. Scope change control system defines complete procedures by which project scope can be changed [PMI, 2009]. It includes the paperwork, tracking system, approval levels necessary for authorizing changes. Scope change control system is always integrated with the overall change control system. Performance measurement techniques help to determine the causes of variance and its magnitude with which team can take corrective action in precise time [Conchúir, 2012; Sliger and Broderick, 2008]. Unfortunately, only few projects run exactly according to plan which is why these measures are needed.

Scope change, corrective action, and lesson learned are the outputs from the scope change control. Scope changes are the modifications on the early defined and agreed project scope. These changes are likely to affect to time, cost and quality. Also, these changes require adjustment in planning and technical documents. Corrective action is the work that is done to bring expected future project performance inline with the project plan [Sliger and Broderick, 2008]. The reason and procedure involve in choosing the action; along with causes of variance and other learned lessons are documented into database. This information can be used as a reference in future by other projects of the performing organization.

2.2. Importance of project scope management

Project scope defined at the beginning of a project is likely to be changed during different development phases. These changes are also known as SC. Scope changes can also occur due to the changes in the government regulations that drag the scope out from the early-defined margins. Scope changes always have serious affects on the overall performance of the project. The changes in the scope directly affect factors such as cost, time and quality related to the project. To overcome these effects, besides the properly defined scope definition, proper change management is also required. Project manager always needs to be aware of the changes in the previously defined scope. Furthermore, they should maintain a good relationship between customers and developers. There are four strong reasons [Inder and Rivera, 2007], due to which proper PSM is required for the successful projects.

- *Cost.* Scope changes can affect the work that is performed and those that will be done in the future. The cost of the project will increase due to the reworks and changes.
- *Time.* Scope change always has a serious impact on project completion time [Collegiate-project, 2009]. It is because most of the allocated time and resources are not increased enough compared to the amount of work increased by scope change.
- *Quality.* Scope changes increase the unplanned overheads i.e. affect the overall schedule of the project and create an extra pressure to carry out the extra works. Due to this, the responsible project personnel have to make a quick decision and fix these changes, resulting in affect of the project quality.
- *Moral.* Scope changes can cause the loss of control of the work planned by team. Changing the focus or direction to meet the change requests has a negative impact on team morale. SC forces the project manager and team members to spend more time on the job, with less time for family and recreational activities. Spending more time at work reduces the overall morale of the team because they could feel that they live only to work [Collegiate-project, 2009].

Scope changes can affect the project during different phases, but the impact differs on the timing of its occurrence. The changes occurred in the later stages of software development process highly increase cost, risk, and project duration [Collegiate-project, 2009].

3. Scope creeping

Every project passes through its development life cycle by following a certain development paradigm. The project scope states the objectives and work to fulfill in the project. It is defined in the initial phase of the project life cycle and mainly describes project goals, deliverables, tasks and deadlines [Bronstein, 2010]. Project scope becomes more refined as the project progresses. It, however, always remains within the initially defined scope. When the customer's expectations change, the feature and functionality of product can also increase. This causes the scope to go outside of the initial parameters affecting the time and cost of the project. This change in the project scope is called SC [Babu, 2005; Gurlen, 2003].

In other words, SC is defined as the extra expansion in the scope of a project due to the changes and addition of the requirements that are not included in the initial planning phase of the project. The additions of requirements from customers affect the developers work performance by increasing the workload [Gupta, 2011]. SC is also known as focus creeping, or feature creeping, or function creeping, or requirement creeping [Gube, 2008; Elliott, 2007]. SC occurs more frequently during the later stages of the project such as programming and testing, than the earlier design stage [Gurlen, 2003]. During the later stage, teams are able to understand and develop the clear vision toward the project goal. This situation mainly happens when the initially defined requirements and objectives are unclear. The clear understanding of project scope might cause small or enormous changes in the project scope. When SC occurs, it directly affects the budget and schedule of the project and can lead to the failure of the project. In order to make a project successful, it is important to manage the scope when any changes are proposed. The goal in managing SC is to try to minimize the impact of any changes on the project [Gurlen, 2003]. Many company methodologies have change control processes for managing SC. These change control processes often include filling out forms describing the requested change in scope and an approval process. Some view this form as a good method to raise awareness of the project stakeholders to what the change is and what the implications of change are such as an increase in the timeline and cost. Others view a change control process as a way to deter potential changes in scope [Veryard, 2001].

3.1. Causes of scope creeping

The global survey 2010 conducted in Qatar showed that SC is one of the leading causes of project failure [Hussain, 2012]. Furthermore, it discussed the direct cost of SC. Another piece of research conducted by Anthes [1994] described how SC is the main cause of project failure and listed the reasons behind its occurrence. He conducted the survey among 160 information technology (IT) professionals. The result of the survey showed that 80 percent of the respondents reported that SC occurs frequently or always. According to the articles published in Money Marketing [2010], it is very important to include the SC clauses in the project contract. Generally, two types of the project development contracts are in practice in software development. Firstly, the variable price contract where hourly fee is mentioned that means customer will have to pay for the total working hours. Secondly, the fixed price contract where the total amount for the project development is fixed in the beginning that can't be changed during the development. SC can occur in both fixed price as well as variable price contracts. Projects with a fixed price contract suffer more from SC than a variable price contract. The customer may argue that the fixed price includes what they want more in the project; whereas the developer thinks it is outside of the range of the fixed price. The customer and the developer argument about the extended scope can create a bad relationship that could lead to the failure of the project. Apart from the above mentioned causes, SC can also occur in the case where the project scope is clearly defined due to the addition of already defined features to the product design without providing the proper resources, time and budget.

Besides Anthes survey, a number of other studies on PSM have been conducted. Those studies listed a number of factors that can cause SC to occur. For example, Larson and Larson [2009] listed and verified the top five causes of SC. They conducted a case study in New Energy Inc., a fast-growing supplier of green energy solution, and verified those causes. In addition, many renowned project management scholars and practitioners have also identified and listed the causes of SC. These causes can be grouped under two perspectives: business SC and technology SC.

Business SC. The business needs of a company changes constantly, which force a change in the business requirements. To support new requirements, new systems and technologies are designed and implemented in the projects. These changes in the business requirements make the occurrence of SC in the project known as business SC. The common causes of business SC occurrence are listed as follows.

(i) *Poor requirements analysis* [Larson and Larson, 2009; Anthes, 1994]. Customer cannot always provide their specific requirements, but they only deliver their ambiguous ideas. Due to this, customer always wait to see the product design with willingness to give their feedback, which can change the project scope. Forty-four percent of the respondents of Anthes' survey reported poor requirement analysis as a leading cause of SC.

(ii) *Misinterpretation* [Kerzner, 2009]. Scope creep is caused by a misinterpretation of what is contained in the project scope, contract, or narrative description of the work required for the project named as statement of work (SOW). A misinterpretation among the stakeholders occurs mainly due to the mixing tasks, no proper structure and order, variation in task size and work description and failing to get the review.

(iii) *Not involving the users early enough* [Larson and Larson, 2009; Anthes, 1994]. It is important to involve the user in the requirement analysis and design phases to know what they want. If stakeholders think that they know precisely what the user wants, then it is a big mistake that can lead to the occurrence of SC. Nineteen percent of the respondents from the survey conducted by Anthes reported that not involving the users early enough made the occurrence of SC in their projects.

(iv) *Underestimating the complexity of the project* [Brenner, 2002]. A project passes through different problems when it is new for developers and the developing organization. Project development team cannot expect what the actual needs of the project are and which is the best way to achieve those needs due to a lack of previous experience and support from the experts.

(v) *Lack of change control management* [Kerzner, 2009]. There is always a possibility for SC to occur at any stage of the project. Thus, it is important to handle any deviation of the project from the initial scope by designing a proper change control management from the early phase of the project. Without proper documentation of change management, no one involved in the project development receives the proper knowledge of features and scope change. The proper change control management minimizes the affect of SC by protecting the project from huge cost and schedule deviation.

(vi) *Lack of risk planning and management* [Kerzner, 2009]. Risk management is considered to be the most important part of the project management. Failure to

identify the risks and their impact upon the project directly affects the cost and schedule of the project. So, lack of risk analysis and planning can cause SC.

Technology SC. Technology SC can be further classified into two categories. The first type is the result of trying to please the customer. The project manager tries hard to make the customer happy by implementing all of their demands in the project without saying 'No' which causes SC. This SC can be managed by conducting visual walkthrough sessions between the customer and developer to identify the key features that the project must have. It will help developers and designers to deliver a final product closer to the client's needs, which will more likely result in project success [Babu, 2005]. The second type is known as "technical gold plating" which may occur when the developers decide to add features and functionality that have not been specified in the approved requirements definition. In most of the cases, this is caused by the implementation of unnecessary and ambiguous technology to please the customer, which significantly increases the risks to the project's successful implementation. The main causes that occur technology SC are as follows.

(i) *Gold plating* [Larson and Larson, 2009]. This term refers to the practice of exceeding the initial scope of the project by adding some attractive features to project, with belief that it will increase the customer satisfaction. Gold plating consumes more time and cost, however, it is unable to guarantee the increase in customer satisfaction.

(ii) *Lack of formal communication* [Kerzner, 2009]. Proper communication between the stakeholders and the development team helps to determine the key information which support the decision making process. Any decision made without getting the proper information can increase the chance of SC leading to project failure.

(iii) *Customer requirement changes* [Abramovici, 2000]. Customer always willing to show their existence in the market with a successful product. So, they try to cover market demand and make their product more flexible and reliable to the users. Due to this, the customer can add and change the previously defined project scope during the development stage which can cause SC.

(iv) *Environment changes* [Abramovici, 2000]. SC occurs due to the organizational structure or environment, which is functional or projectized [Greiman, 2013]. In a projectized environment, the authority and power are retained in project manager. In this environment, the project manager has the flexibility to evaluate the change impact to the project and suitably to say

'No', or to perform the changes in the next phases. However, in functional environment the users have more power and authority then project manager. The project manager could have to accept some additional requirements without differing the original schedule and budget. In some situations, SC occurs due to unrealized schedule and resources.

(v) *Platform changes* [Abramovici, 2000; Anthes, 1994]. SC can occur due to the change in the working model or technical platform in between the project. Stakeholders try to implement new technology and the latest development model in the project, to make it more competitive and reliable in the market. The changes in the development platform in between the project affect the schedule and cost of the project. Besides, it is also difficult and time consuming to make the user familiar with a new application and technology. Thirty-six percent of the respondents of Anthes' survey reported that a new application unfamiliar to the users is a leading cause of SC.

The causes mentioned above are not only the factors responsible for SC. There are many different seen and unseen factors that too cause SC. For example, when a team proceeds through the various phases, frequently one or more of the team members will strive to improve or perfect the situation or product. This may result in a change of the project scope [Veryard, 2001]. SC can also occur when the proper business requirements have not been defined upfront and involve the wrong user in the definition of the requirement.

Some changes in scope are caused by external entities to the organization. Changes in things such as legislation, regulatory changes, market conditions, or in the technologies being utilized can cause SC. All these items are out of the control of the project team and their company [Gurlen, 2003].

When multiple projects are consolidated into one, such as to combine resources, this can increase the scope of all the projects. Thus additional effort and time will be needed to determine how the various projects fit together [Gurlen, 2003].

Even different researchers decipher different causes of SC. If all the stakeholders concentrate and are aware about the possibility for SC, and attempt to develop and follow a well-managed scope change strategy from the beginning of the project, then SC can be controlled and managed.

3.2. Scope creeping impact and its measurement

SC is considered as one of the leading causes of project failure because it affects the project by increasing the cost, delaying the schedule, and decreasing the quality of the products, which is described briefly in Section 2.2.

In every running project, scope is changed due to new requirements or a modification of existing requirements. The team members always have to perform the necessary analysis before implementing any scope change request. The team members might be able to recognize some of the impact of the scope change during the development. However, it is difficult to communicate those impacts of scope change to the stakeholders, who made a request for change. If the impacts are not communicated to the stakeholders, then they will not have any idea about how their request to change the requirements can affect the project. To communicate those impacts, a quantitative measure of the impacts is necessary. Software measurement metrics are considered as the best solution to determine the quantitative measure of the impacts. The impact measurement means the measurement of the scope performance against the scope baseline. The process of measuring scope performance against the scope baseline is called variance analysis [Piscope, 2013]. The process of scope performance measurement is used iteratively throughout the project lifecycle to monitor scope.

The impacts of SC can be managed by measuring scope-variance from baseline using software measurement metrics such as EVM, BS, and requirement metrics.

4. Software measurement metrics in project scope management

4.1. Software metrics and measurement

Software metrics measure software complexity on different aspects such as software process, quality, resources and products [Singh et al., 2011]. They help the development team to understand and monitor the progress of the project, which can lead to improved project performance and product quality. The descriptive data are collected for the software metrics, which is also easier and understandable for the developers to predict, manage and control the product. According to Goodman [1993] metric is a continuous application of measurement-based techniques to the software development process and its product to supply meaningful and timely management information, together with the use of those techniques to improve the process and its product. In general, metric can be defined as a measurement derived from a software product, process, or resources. The main purpose of metric is to provide a quantitative assessment of the extent to which the product, process, or resource processes certain attributes [Costello-Liu, 1995].

According to Singh et al. [2011], software metrics can be classified into process metrics and product metrics. Metrics that are used to measure the properties of the software development process are known as process metrics. Process metrics always support in determining flow of the project and predicting the size of the final of the developing system. Process metrics provide overall information of software development process focusing on cost, time and phases related to the product development. Process metrics include cost metrics, effort metrics, reuse metrics, and advancement metrics. On the other hand, the metrics that are used to measure the properties of the software are known as products or quality metrics. Product metrics help in improving the quality of the system by comparing with the existing systems. Product metrics include product non-reliability metrics, functionality metrics, performance metrics, usability metrics, cost metrics, size metrics, complexity metrics, and style metrics [Singh et al., 2011].

Measurement calculates something, therefore, if we consider the real project scenario, then measurement is defined as the measurements of the different attributes and entities related to the project's product. Entity is a real object exists in the world and attributes are properties or behavior of object which is defined as an entity. According to Fenton [1997], measurement is a process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to

clearly define rules. In general, software measurement is defined as the measurement and collection of the different metrics related to the project management activities like planning, organizing, controlling and improving for guiding the product development in the proper direction.

4.2. Roles of software measurement metrics in project scope management

Software measurement metrics are used to measure the past and give guidance for the future directions. The observation of metrics on ongoing projects enables us to collect the factors that are used to meet the project goal. Some of the software metrics enable continuous comparison of planned to actual values, which helps to locate weaknesses in the software development process and can enable process improvement. Metrics enable early risk recognition and support the mitigation of those risks. Therefore, metrics improve communication between project team members and the other stakeholders. A properly planned and implemented software measurement can improve the development process and the quality of the product.

Software measurement metrics are considered as an instrumental factor to control and save the project from SC. Software metrics can help the project manager to effectively communicate with stakeholders regarding the impact of adding features. SC can be minimized employing metrics at every stage of project lifecycle in the following ways [Zuber, 2013]:

- *Project repository.* Software measurement metrics can be used as a project repository to store the historical data and calculate productivity. The historical data related to the organization and team capacity support the calculation of the effort and time required to take on additional work. SC can be determined by comparing recorded time, cost, effort, size and requirements with planned or estimated cost, time, effort, size and requirements.
- *Size, schedule, cost and quality estimating.* Software measurement metrics are used to estimate the size, schedule, cost and quality quickly and easily to analyze the project scenarios early based on the available data. The calculations of uncertainty and risk on time helps the project manager to set project strategy.
- *Variance analysis and adaptive forecasting.* Software measurement metrics are used to track actual scope completion, milestones, effort, cost, and defects against plan. The multiple metrics are implemented to determine the realistic project position. Forecasting the current trajectory beside alternative scenarios

that incorporate additional scope supports effective communication to decision-makers. The ability to log multiple plans (baseline estimates, revised estimates) shows the delta between them, documenting when SC occurred and how that impacted the project.

- *Industry benchmarking and process improvement*. Software metrics are used to observe patterns and identify improvement opportunities using historical data. The causes and effects of size growth on project performance support both initial estimates and forecast. The data from the metrics can be used to construct the trends such as time/effort/cost overruns vs. size, size/requirements overruns vs. size, which support in the decision making and improving the development process.

4.3. Metrics supporting software project scope management

There are a number of software measurement metrics that are used in software development to improve process by measuring and tracking the project from scope baseline. The measuring and tracking helps to increase quality of the product by reducing a number of risks in the project. Among them some of the metrics are described below, which support in minimizing and preventing the project from SC.

4.3.1. Earned value management

EVM is a project management technique that is used to measure the project performance. It does so by using work in progress to determine what will happen in the project in the future. Thus EVM supports the project manager by determining the variances in projects by comparing worked performed and work planned. Furthermore, EVM is very useful in schedule and cost control due to its capability of providing quantitative data for decision making [Dwivedi, 2009]. EVM is developed with adding more feature and functionalities that was not covered by traditional accounting progress measure [Nagrecha, 2002].

Traditional accounting progress measure focuses on planned expenditure and actual costs where as EVM focuses on the actual expenditure that helps the project manager to figure out the actual potential risk areas. Project manager can create effective risk mitigation plans based on the clear picture of the actual cost, schedule and progress determined. EVM is not a specific system or tool set, but rather, a set of guidelines that guide a company's management control system [Nagrecha, 2002].

All the EVM related activities are based on the project baseline or scope baseline. So, it helps in minimizing and controlling the SC by preventing the project from cost

and schedule overrun.

The EVM consists of primary and derived data elements. Primary data elements values are based on time and date when an EVM is performed on the project. The primary data elements are as follows [Attarzadeh and Hock, 2009]:

- *Budget at completion (BAC).* BAC is the sum of all budgeted cost for all the schedule work package and management reserve.
- *Budgeted cost of work scheduled (BCWS) or planned value (PV).* BCWS is the total cost allocated to accomplish the work that has been completed.
- *Budgeted cost of work performed (BCWP) or Earned value (EV).* It is the sum of the budgets for completed work packages and completed portions of open work packages.
- *Actual cost of work performed (ACWP) or Actual cost (AC).* ACWP is the actual cost to accomplish all the work that was performed within a specific date or schedule.

The derived elements are derived from primary data elements that show the project performance. The derived data elements are as follows [Attarzadeh and Hock, 2009]:

- *Estimate at completion (EAC).* It is the total expected cost required to finish the project. At the beginning of the project BAC and EAC will be equal. When ACWP vary from BCWP during the development stage than EAC will vary from BAC. Some of the common approaches of calculating EAC are:

  EAC= AC+ Estimate to Complete (ETC)

  EAC=BAC / Cost performance index (CPI)

  EAC =AC + ((BAC - EV) / CPI)

  EAC= AC + (BAC - EV)

- *Estimate to complete (ETC).* ETC is the total cost required to complete the remaining part of the project. It is always calculated from where performance measurement is carried out in the project. ETC can be determined by following some of these approaches:

  ETC =EAC-AC

  ETC= (BAC / CPI) - (EV/CPI)

  ETC= (BAC - EV) / CPI

- *Schedule variance (SV).* SV is the difference between EV and PV. It is calculated in terms of difference in cost between the amount of work has to be completed in the given time period and work actually completed. A negative value shows that the project is behind schedule, whereas a positive value

represents project is ahead of schedule.

SV= EV-PV

- *Cost variance (CV).* CV is the difference between EV and AC. This is the actual cost value by which the project stage is determined. A negative value shows that the project is going over budget and positive value represents project is under budget.

  CV= EV-AC

- *Variance at completion (VAC).* VAC is the difference between BAC and EAC. This is the monetary value by which the project will be over or under budget [Nagrecha, 2002].

  VAC = BAC – EAC

- *Cost performance index (CPI).* The CPI is the ratio of EV to AC. A CPI of one implies that the actual cost matches to the estimated cost. A CPI greater than one indicates that the work is accomplished for less cost than what was planned or budgeted. A CPI less than one indicates the project is facing a cost overrun [Nagrecha, 2002].

  CPI = EV / AC

- *Schedule performance index (SPI).* The SPI is the ratio of EV to PV. A SPI of one implies that the project is on schedule, and a SPI greater than one indicate that the project is ahead of the planned schedule. Whereas SPI less than one indicate that the project is behind schedule.

  SPI = EV / PV

4.3.2. Balanced scorecard

BS is a strategic planning and management tool that is used by profit and non-profit organization to monitor their performance against their strategic goals. It enables any organization to clarify their vision and strategy, and translate them into action through the process, customer, and learning and growth perspectives [Kaplan and Norton, 2007]. Besides, it also supports in controlling the consequences that arise from the execution of different staff activities. Any organization can get continuous feedback from the internal and external business processes and implement BS, which enables them to keep their performance and result precisely. The term “scorecard” signifies quantified performance measures and “balanced” signifies the system is balanced between short-term and long term objectives, financial and non-financial measures, lagging and leading indicators and internal and external performance perspectives.

Robert Kaplan and David Norton originated BS in 1992 as a performance measurement framework by adding non-financial performance measures to traditional financial metrics. Kaplan and Norton [1996] describe the innovation of the BS as follows:

> "The BS retains traditional financial measures. But financial measures tell the story of past events, an adequate story for industrial age companies for which investments in long-term capabilities and customer relationships were not critical for success. These financial measures are inadequate, however, for guiding and evaluating the journey that information age companies must make to create future value through investment in customers, suppliers, employees, processes, technology, and innovation".

BS has a simple performance measurement framework in the early stage. Some of the weaknesses and vagueness of the Kaplan and Norton BS were recognized and modified. The resulted BS became one of the best strategic planning and management systems. This new BS transforms an organization's strategic plan from an attractive, but passive, document into the daily marching orders for the organization. It provides a framework that not only provides performance measurements, but also helps planners to identify what should be done and measured. It enables executives to truly execute their strategies [Kaplan and Norton, 2007].

BS maps the organization's strategic objectives into performance metrics in four perspectives: financial, internal processes, customer, and learning and growth. These perspectives provide relevant feedback as to how well the strategic plan is being executed so that adjustments can be made as necessary. The four perspectives are described below: [Werner and Xu, 2011]

(i) *Financial perspective.* Financial performance measures indicate whether the company's strategy implementation and execution are contributing to bottom-line improvement. The traditional financial measures do not improve customer satisfaction, quality, cycle-time, and employee motivation. According to Kaplan and Norton [2007], financial objectives alone are an insufficient measure for private and government organizations. The ultimate goal of any organization is to make profit which can only be achieved by selecting the measurement index that is well integrated with the strategy. The financial indicators of any organization are set according to their own strategic goals.

The financial objectives of any organizations are different according to the development stage. Among the different development stages, growth, sustain

and harvest stages are considered as the key stages. [Kaplan, 2009] Organization growth is considered as the early stages of development, this means that organization possesses services and products with a certain potential for growth. The organization has to develop infrastructure, distribution networks and production system to shows that they have a potential to growth. The main financial objectives for growth are revenue growth, growth of sales in targeted markets, and customer groups. In general most of the units in the organization have to attract the investments and apply reinvestment techniques. That means they are in a sustaining stage. In this stage, the organizations main goal is to keep the current market share and, in some cases, gradually increase it. The financial objectives in this stage are operating income and gross margin. Harvest stage organization management tries to collect the investments that have been made in previous two stages by focusing on the maintaining of equipment or performing the similar tasks as before, rather than new investment. The main goal at this stage is to maximize cash flow and reduce the working capital requirement.

The main financial objectives are revenue growth and mix, productivity improvement/cost reduction and investment strategy/ assets utilization [MacLellan, 2007]. Revenue growth and mix is mainly about reaching new customers and markets, expanding products and services and changing them, introducing new pricing policies. Productivity improvement / cost reduction include reduction of indirect costs, sharing common resources with other departments, and lowering direct costs. Investment strategy/assets utilization means a greater utilization of fixed assets base and improving return on investment. Financial perspectives of BS focus more on the measurement of these financial objectives during the development lifecycle.

(ii) *Customer perspective.* Customer perspectives mainly focus on the measurement of customer needs and how to fulfill their expectations to succeed in business. In modern business ethics, customer satisfaction is considered the main part of business. Kaplan and Norton [1992] argue that success of any organization is measured by how effectively and efficiently they meet the needs of their customers. Customer satisfaction metrics are defined by selecting the objectives and measures by dividing the market in different segment by analyzing the different kinds of customers. Customer measures identified for all types of organizations are market share, customer

retention, customer acquisition, customer satisfaction, and customer profitability [Kaplan and Norton, 2007]. Market share is the proportion of business in a particular market segment that is measured when it is identified. The sources of information for market share are public, industry groups, trade associations, and government organizations. The measurement of customer retention determines the change trends which helps the organization to continue customers with them. The measurement of customer acquisition determines the rate of attraction of new customer towards the company. The success of any organizations depends on the customer satisfaction because satisfied customer stays loyal to them [Niven, 2006]. Customer satisfaction can be determined by conducting surveys among the customers. The customer profitability measure shows the efficiency of actions taken within marketing campaigns. However, a highly satisfied customer business does not necessarily guarantee high revenue, because sometimes companies have to compromise with the profitability to satisfy the customer by fulfilling their demands.

(iii) *Internal business process perspectives.* The internal business perspectives based metrics begin with the managers' knowledge about the running business. Moreover, it determines whether their products and services fulfill customer's requirements or not. The conventional performance measurement systems focus on monitoring and improving existing business processes, where as scorecard approach usually identifies entirely new processes at which an organization must excel to meet customer and financial objectives. There are different business processes such as innovation process, operations process, and post-sale service that help in choosing right measures for internal process perspectives. Innovation process includes identification of customers' needs and development of new solutions to satisfy those needs [Werner and Xu, 2011; MacLellan, 2007]. It is important for organizations to have long development and design plans. The operation process begins when an organization receives an order for a product from the customer and ends when the product is delivered to the customer. Metrics such as cost, quality, and time are applicable in the operations process. Post-sale service relates to all activities regarding the satisfaction of the customer's needs after the purchase of product such as processing of payments, training, and support.

(iv) *Learning and growth perspective.* The learning and growth perspective

includes employee training and corporate cultural attitudes related to both individual and corporate self-improvement. In the current climate of rapid technological change, it is necessary for knowledge workers to be in a continuous learning mode [Capatina and Crista, 2011]. Learning and growth metrics guide managers in focusing training funds where they can help, furthermore, they also provide the infrastructure for organizations to achieve their stretch goals identified by the previous perspectives [Kaplan and Norton, 1996]. The gaps between the financial, customer and internal business process objectives and the organization's existing capabilities to achieve these objectives, lead to the need to invest in the three categories of learning and growth scorecard such as employee capabilities; information systems capabilities; and motivation, empowerment, and alignment. Employee capabilities mainly focus on collecting and implementing the creative idea and mind of employee in the further development of the organization. Whereas, information system capabilities focus on creating the environment such that the employee can get accurate, complete and up-to-date information about customers, their goal, internal processes, and possible consequences of employee decisions [Kaplan and Norton, 2007; Rohm, 2008]. The organization should also have a proper online information system so that employee easily access customer feedback and implement changes based upon feedback to satisfy them. The motivation and empowerment, and alignment categories focus on the employee motivation. If the employees are poorly motivated then the organization is unable to reap benefit from them, even though they could be well educated and have access to information on products and services [Niven, 2008]. Senior management is responsible for motivating the employees by collecting and considering their suggestion.

4.3.3. Requirement metrics

The changes in the project requirements are considered as one of the leading causes of SC. Requirement measurement metrics can be used to minimize SC. The metrics that are useful in identifying the risks of a project by identifying errors and changes in the requirements document are known as requirement metrics [Costello and Liu, 1995; Carlos, 2010]. These metrics validate the written requirements against actual requirements. They evaluate whether the requirements are complete or not. There are many metrics used for measuring the requirements such as requirement traceability

metric, and volatility metric. A single metric cannot ensure overall quality; therefore, multiple metrics should be used for measurement.

(i) *Requirements traceability metric (RTM).* This metric is a tool which helps to ensure that project's scope, requirements and deliverables remain same when compared to the baseline. It traces the requirements from the project initiation to the final implementation. RTM can be implemented during all phases of a project to track the requirements, assist in the creation of request for proposal (RFP), project plan task, deliverables documents and test scripts [Guo et al., 2009]. Besides that, it is also used to ensure that all system requirements have been met during the verification process. The metric forms the basis of the project scope, because of this, it is developed in concurrence with the initial list of requirements during the beginning of the project. It is updated after the development of the specifications and test protocols.

The metric is considered as a bilateral metric. It tracks the requirement forwards by examining the output of the deliverables and backwards by looking at the business requirement that was specified for the feature of the product.

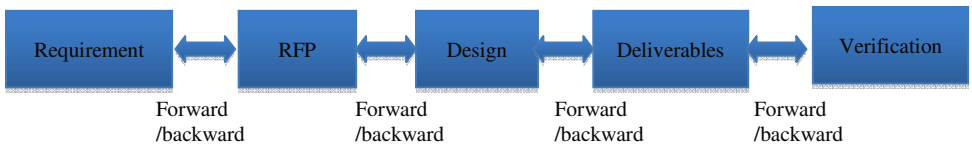

Figure 2: RTM [Carlos, 2010]

The RTM is used by the validation team to verify that all the requirements are met and to identify changes to the scope when they occur. The use of the RTM enhances the scope management process. It also assists with the process control and quality management. Therefore, RTM can also be thought as a process of documenting the connection and relationships between the initial requirements of the project and the final product or service produced [Carlos, 2010].

(ii) *Requirements volatility metric.* The degree of measuring the requirement changes over a time period is called requirement volatility. Apart from this, it also determines the reasons of requirement changes. These factors are measured to know whether the changes are consistent with current development activities or not. It helps in tracing future requirements, design, and code volatility by indicating the requirements changes such as addition,

deletion and modifications. As requirement volatility can be high in the initial phase of software development, it should be reduced as the project progresses so that further development should not be affected.

5. A Case study: Challenging factors in students' software projects

5.1. Motivation and purpose of case study

A software development project always starts with the initial project scope that includes objectives, cost and schedule estimation, and quality of the product. Even though project scope set a clear boundary for project, there are different factors that take the project out from the initially defined boundary.

Different research has shown that there are different factors responsible for the occurrence of SC. The main aim of this case study is to determine the factors which make the project more challenging, however every challenging factor cannot be considered as the cause of SC. In this case study, the main challenging factors pointed out by several project managers in their projects will be determined at the beginning. Then the causes of SC are determined by comparing and analyzing those challenging factors with the causes of SC pointed by different past research. The case study fulfills the first objectives of this thesis work "the causes of SC and how it influences the project plan and development".

In the software development lifecycle, a number of problems occur in different areas such as planning phase, communication amongst team members, requirement elicitation, risk management and planning, requirement change management, team members' expertise, and work division. In this case study, numbers of question are prepared related to those areas based on their strength of effect on the cost, schedule, quality, scope, and team morale. All the questions are properly organized under the area which they are related to.

5.2. Case study environment

The case study was conducted on a group of students from the School of Information Sciences, University of Tampere, who took the course entitled "TKOPS117: Software Project Management". The School of Information Sciences offers this course as a compulsory course for the fourth year students of computer science. In this course, students work in project groups and construct a relatively large software project. The main goal of this course is to familiarize students with the principles and practices necessary for the initiation, management, and supervision of a software project [ProjectWiki, 2013]. During the course, real software projects are defined and numbers of students are assigned to the projects according to the project size and necessity. The

main responsibility of the project group is to produce the final product under the supervision of the course supervisors and clients within the allocated time frame.

In this case study, the software projects that were conducted during the academic year 2011-12 and 2012-13 are considered. According to the course statistics report [Mäkiaho and Poranen, 2012], in the academic year 2011-12, 30 students were assigned as project managers and 67 students were assigned as developers in 14 different software projects. Likewise, in the academic year 2012-13 [Mäkiaho and Poranen, 2013], 40 students were assigned as project managers and 54 students as developers in 13 different projects. Out of 70 project managers from the two academic years, only 18 project managers responded for this case study. Moreover, it was found that the respondents were not only involved as project managers in their projects, but also had different other roles such as designer and developer. Forty and 30 percent of the respondents agreed that they were involved in their projects as developer and designer respectively. Further, more than 70 percent of project managers replied that they used scrum methodology in their project with some modification, such as no face-to-face daily meetings or no fixed length iterations.

5.3. Data collection methodology

A questionnaire was designed to investigate the factors that contributed to project success, challenged and also factors that contributed in project failure. The questionnaire consists of eight different sections: general information, project objectives and planning, team member's expertise and work division, communication, risk planning and management, requirement elicitation, requirement change management, and SC. The first section gathered the information regarding the project title, implemented project development method, and the roles of the respondent, apart from project manager. The project objectives and planning section consisted of 11 questions with multiple options to collect the information such as clarity of scope defined, stakeholders supports, expectations and involvement, resource allocation, and used of project management tools. The team members' expertise and work division section consisted of four different questions to collect the information about team competence, participation, motivation, and work division with multiple options. The fourth and fifth sections collected the information related to the communications among the stakeholders, risk identification, and its management. The requirement elicitation and change management section consists of different questions to gather information such as process of collecting requirement, finalizing the set of requirements, causes of

requirement change, and the process of handling the changed requirement. The final section consisted of different questions related to the causes of SC and its proper management. Hence, the questions were formulated closely related to the project success, challenges and failure factors.

5.4. Data collection and its analysis

In the case study, 18 project managers from 16 university projects answered the questionnaire. The project managers were asked to answer all the eight sections of the questionnaire. The data from the respondents were collected and analyzed under the following sub sections:

(i) Initially defined project scope. The initially defined project scope, which included project goals, deliverables, tasks, costs, and deadlines, is considered as fundamental for the software development process. The success, challenges and failure of the projects are highly determined by the project scope defined in the beginning of the project. If the defined project scope is ambiguous, then there is a high chance of a potential misunderstanding arising among the stakeholders which may cause SC. Project manager's opinion about their project scope definition at the beginning was collected. More than 50 percent of the respondents replied that the scope was well defined in their projects. The results are shown in Table 1.

Table 1: Project scope definition in the case study projects

Initially defined project scope	Percentage of responses
Well defined	56
Ambiguous (unclear, having multiple meaning)	39
Unrealistic and unachievable (objective that they cannot fulfill)	0
Other	6

(ii) *Client expectation, executives' support and resource allocation.* Software projects which have their objectives well-defined according to the client expectations and available resources have high chances of success. It does not mean each and every client request has to be covered in the project objectives. The objectives are set according to the resources allocated to the project, in this instance resources mean the hardware and software required to develop the project. Allocating insufficient resources has impact on the initially

defined scope that eventually causes SC, besides which, the support of executives has a great role in project success. The majority of the project managers replied that they received clear and achievable project objectives from their clients. Fifty percent of the project managers stated that they received executive support only in the important stage of development and they had to share resources with two or three members. The respondents' replies regarding their client expectation, executives' support and resources allocations are illustrated in Figures 3 and 4 and in Table 2 respectively.

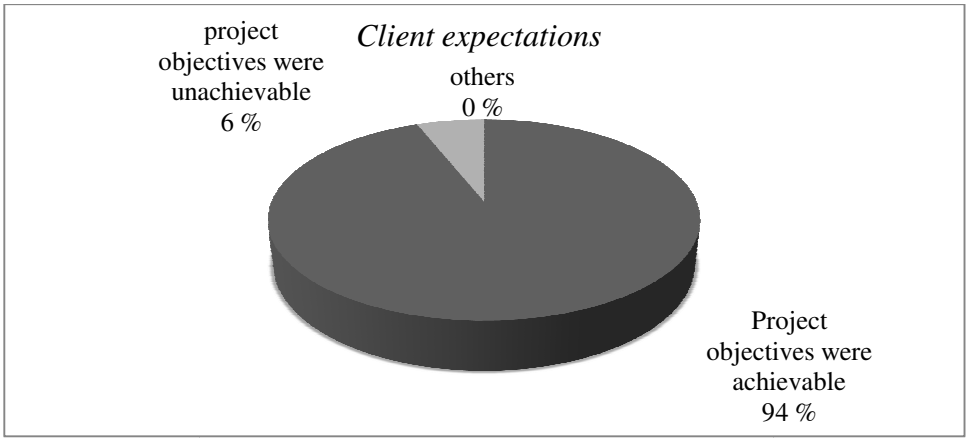

Figure 3: Client expectations in the case study projects

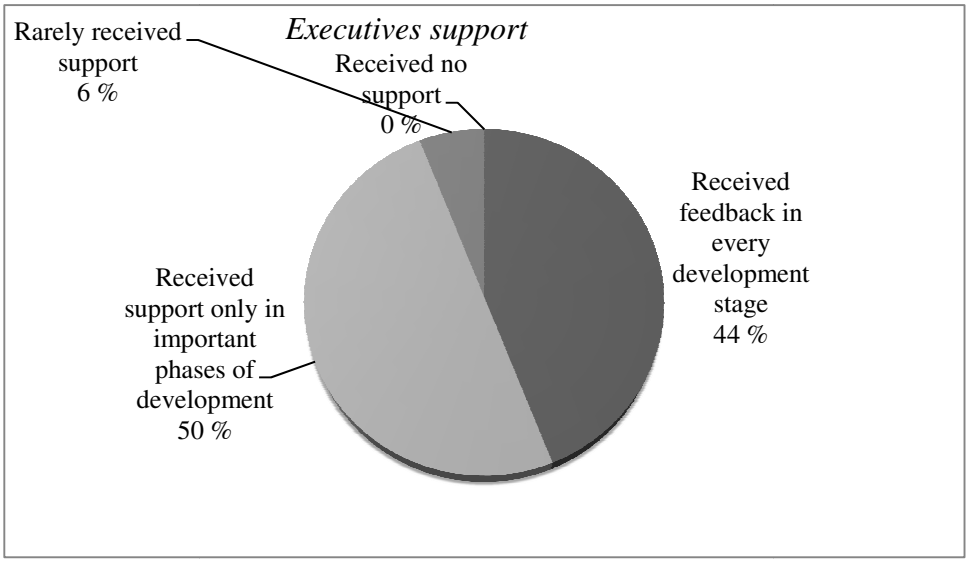

Figure 4: Executives support in the case study projects

Table 2: Resource allocation in the case study projects

Allocation of resources	Percentage of responses
Provided individual set of resources	39
Shared resources among two members	22
Shared resources among more than two	28
Some necessary resources were never provided	11
None of the necessary resources were provided	0

(iii) *User involvement.* The involvement of the user in the project supports the development team by providing the actual needs and other constructive feedback. Projects that involve users only at the end of the project might affect the project by changing or adding the numbers of requirements, which were not included at the beginning of the project. While analyzing the responses from the respondents, it was clear that the clients were highly involved in the project, but the end users were much less involved. Table 3 provides the statistical data of the case study regarding user involvement.

Table 3: User involvement in the case study projects

User involvement	Client involvement	End user involvement
	Percentage of responses	
From beginning	83	12
In testing phase	0	35
Only in some pieces of work	3	18
Not at all	0	35

(iv) *Project planning and management.* The schedule and cost estimation is considered as an important part of the software development. The case study have excluded the cost part and focused more on the schedule estimation because the case study projects focused more on schedule. Seventeen percent of the respondents replied that their projects were unable to meet the estimated schedule in every task, which affected the overall project schedule. The highest percentage of the respondents (66 percent) replied that some of their tasks did not meet the estimated schedule but at the end about 33 percent of

projects met the overall schedule. From the project managers responses it can be concluded that only 11 percent of projects in the case study met the entire estimated schedule. The poor estimation of the tasks can force the development team to leave or change some initially defined requirements that cause SC.

A proper use of the project management tools leads in the success of project by tracking it in the right way. After the analysis of the responses, only fifty-nine percent of project managers implemented project management tools such as JIRA, Redmine, SVN, Wiki, and The Bug Genie in their project. They used the project management tools for different purposes such as checking progress (27 percent), managing budget (27 percent), share resources (12 percent), discussion (12 percent) and change management (18 percent).

It is important to track the project through every development stage, which can be performed only by using the project management tools. In the same way, a poor estimation of the schedule can force the development team to leave or change the initially defined requirements. A poor practice of the project management tools and poor estimation of the schedule can also cause SC.

(v) *Team competence and work division.* Allocating the team and the division of their work have a significant influence on the success of software projects. If the team members are assigned to the project where they have less knowledge of the working platform, then it can significantly affect the project. The responses received from the project managers showed that more than 39 percent of project teams were composed of medium and learner level members. The project managers' response regarding team competence is shown in Figure 5.

Apart from team competence, proper work division among the team members, team member's participation in discussion and planning, and implementation of the correct strategies to motivate the team members also support in the success of project. The questionnaire had also included some questions to determine how properly project managers covered and implemented those things in their projects. After analyzing the responses, it was found that work division among the team members based on their interest was employed as a motivational factor for the team members. There were only few projects that implement other motivational factors such as training. Some

of the project managers also stated that a few members of their team were highly inactive in planning and discussion. Their detailed responses are represented in Figures 6, 7 and 8.

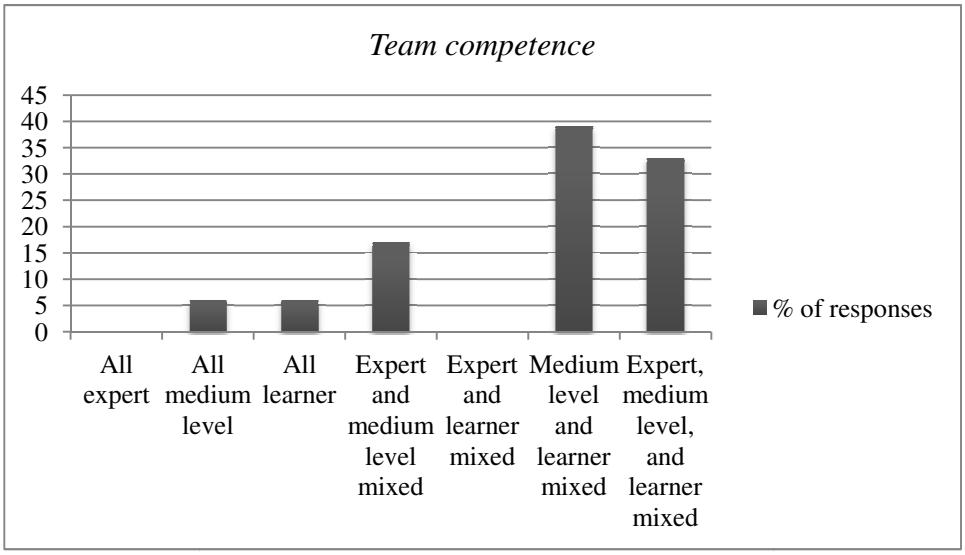

Figure 5: Team competence in case study projects

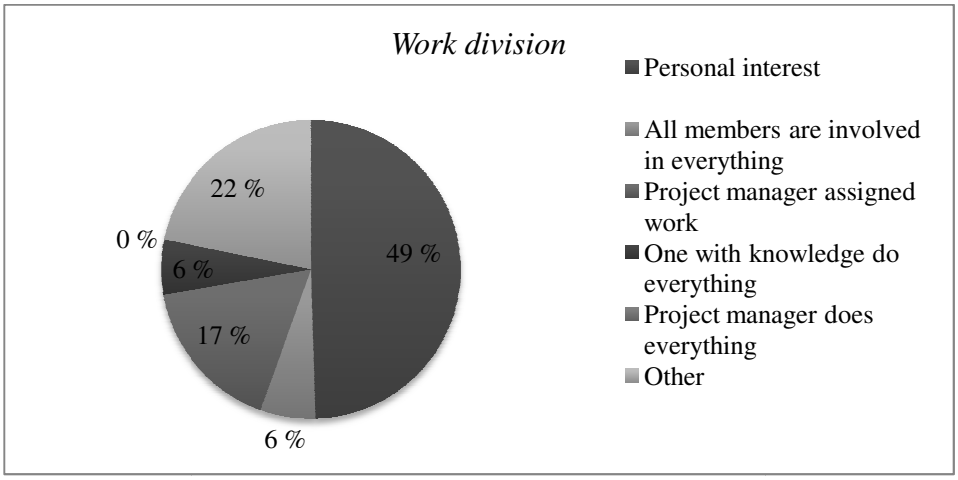

Figure 6:Work division in the case study projects

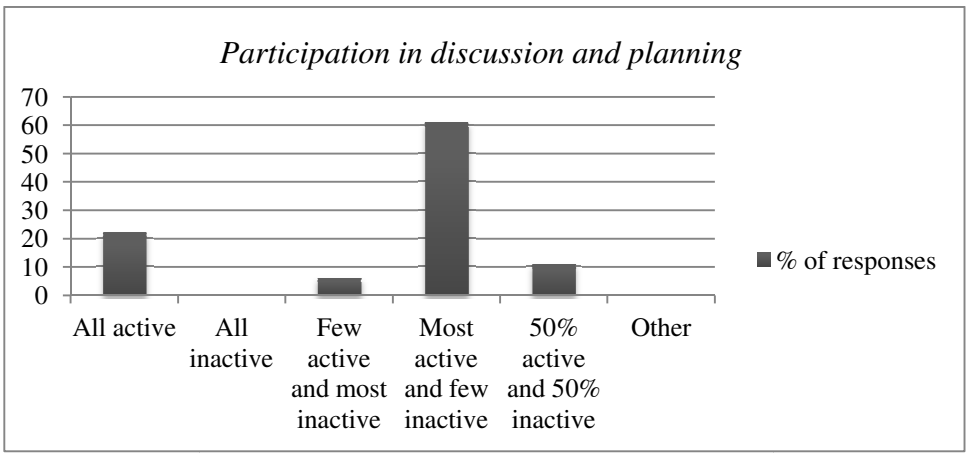

Figure 7: Team members' participation in discussion and planning in the case study projects

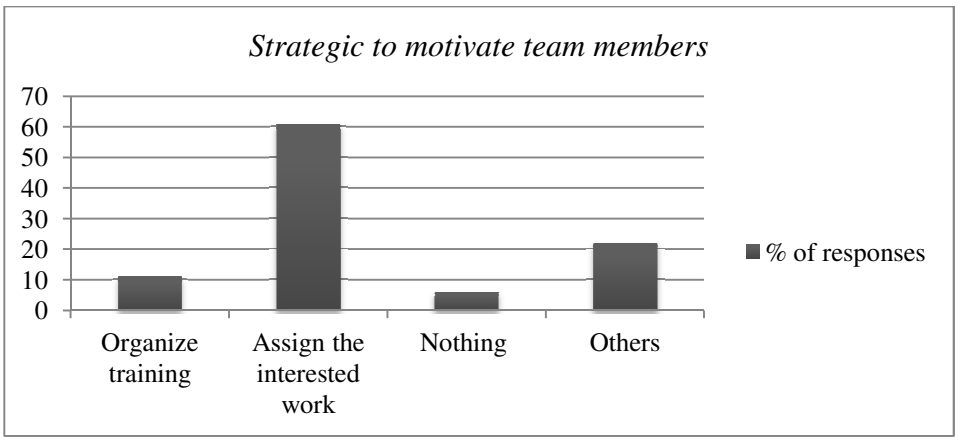

Figure 8: Strategic to motivate team members in the case study projects

(vi) *Communication.* Efficient and effective communication between the clients and team members is practiced in successful projects. Efficient communication supports in discussing problems, analyzing results, and validation of the requirements. The lack of communication between the stakeholders can lead to the development of misunderstandings between them regarding the requirements, and thus could cause requirement changes in the later stage. There are different mediums of communication such as face-to-face meeting, social media, telephones, and so forth. From the case study, it was found that 25 and 30 percent of the respondents used instant messaging (e.g. IRC, Skype, messengers) and face-to-face meeting as the main medium for communication respectively. Furthermore, 21 percent used mobile and phone, 17 percent used social media (e.g. Facebook, Google+), and four percent used discussion boards in software project management tools.

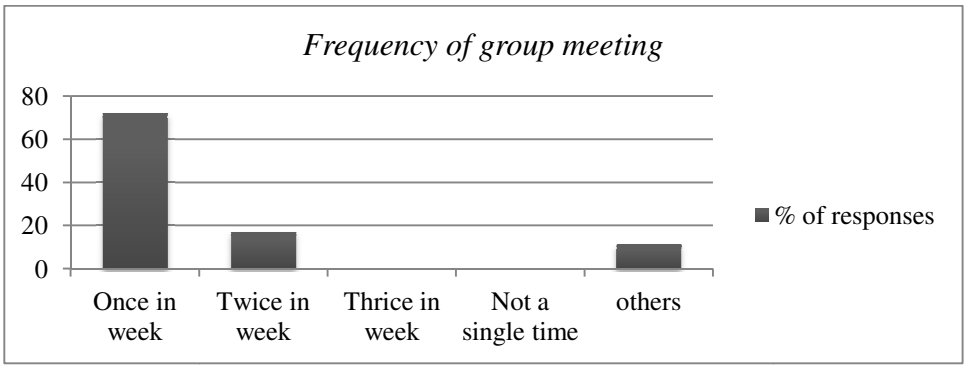

Figure 9: Frequency of group meeting in the case study projects

According to the respondents, 67 percent of project managers reached their team members using above mention communication medium. From the data, it was found that only 28 percent of the project managers used face-to-face meeting to communicate with their team members. The frequency of

group meeting among team members is shown in Figure 9 and the frequency of review meetings with their client is shown in Figure 10.

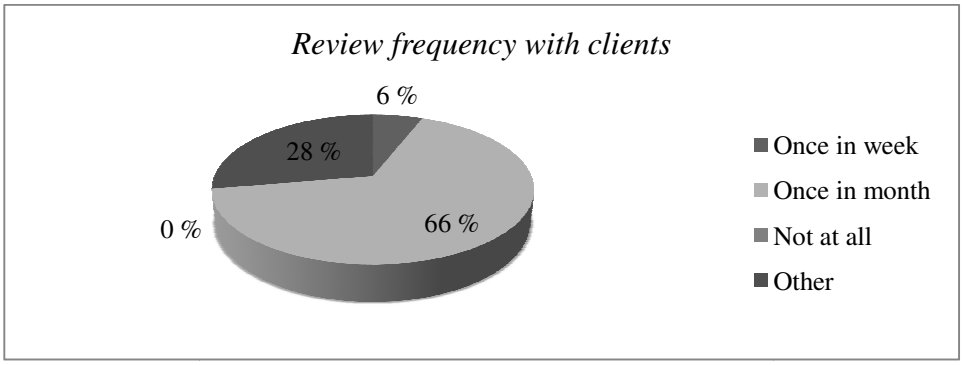

Figure 10: Review meeting with client in the case study projects

(vii) *Requirement elicitation.* Requirements are collected from the different stakeholders such as clients, end-users, developers, professionals, and executive bodies. This section covered the requirement collection methodology, documentation, requirement change, and requirement change management process implemented by the responded project managers in their project. The collected data are analyzed by splitting requirement elicitation techniques in different groups.

a. *Requirements elicitation responsibility.* The most important aspect of the requirement elicitation is identifying members involved in requirement gathering. Fifty-six percent of the project managers replied that both team members along with project managers were equally responsible for requirement elicitation. Seventeen percent of the managers replied that they received a complete set of requirements from their client. Similarly, 17 percent of the managers replied that only the project managers were responsible for requirement elicitation.

b. *Requirement collection.* In practice, there are different methods to collect the requirements such as interview, brainstorming etc. In most of the projects, clients provide the complete set of requirements, whereas in some of the projects, clients provide only their unclear and vague objectives. In that case, the developing team has complete responsibility to finalize the requirements based on the client objectives. The result how requirements were collected in the projects considered for the case study is shown in Figure 11.

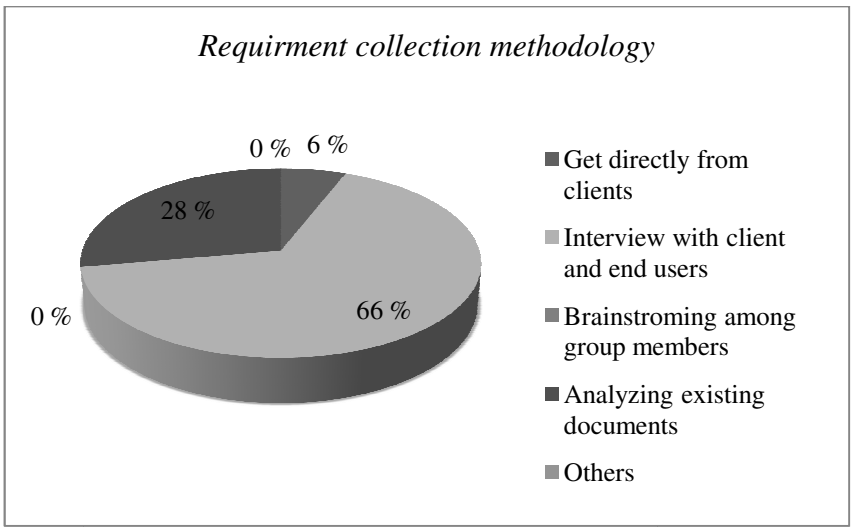

Figure 11: Requirement collection methodology implemented in the case study projects

c. *Requirement documentation and priorities.* All the collected requirements are documented using suitable documentation tools such as MS-Office, Redmine, JIRA etc. After the documentation, requirements which fulfill the project goal are selected and then prioritizing them. Forty-four percent of the project managers replied that they used MS-Office to document requirements in free form. Likewise, 33 percent of the project managers replied that they used tools such as JIRA and Redmine for documentation. Apart from this, 22 percent of the project managers used other media such as project wiki for requirement documentation.

The project managers implemented different methodologies to select and finalize the requirements. Sixty-one percent of the respondents answered that they selected the right requirements by discussing with their clients. Seventeen percent of the respondents selected the requirements discussing only with the team members and the remaining 17 percent of the respondents selected the right requirements by discussing with team members as well as their clients. The opinion of the project managers regarding the final set of requirements is illustrated in Figure 12. Thirty-three of the project managers said that they had covered only a part of the clients' need in their requirements due to the high expectations from client and submitted the end-product with bugs in it.

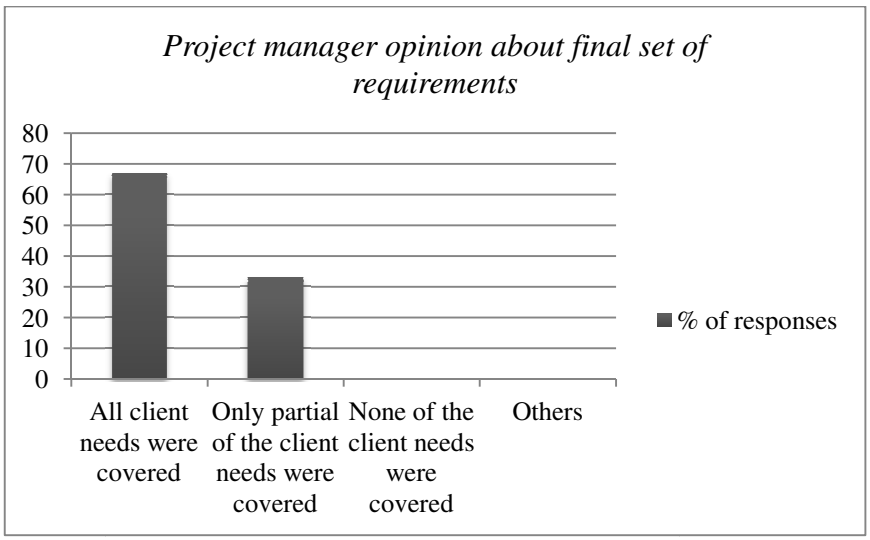

Figure 12: Client needs coverage in final set of requirement in the case study project

(viii) *Requirement changes.* The changes in the requirement highly affect the software project development. Projects whose requirements change frequently have high chances of failure. Eighty-three percent of the respondents faced few times requirement changes in their projects where as only six percent of the respondents' project initially defined requirements did not change. Apart from this, 11 percent of the respondents' projects were highly affected by the frequently changing requirements. The percentages of requirement changed according to the project managers participated in case study are listed in Table 4.

Table 4: Percentage of requirement changes in the case study projects

Percentages of requirement changes	**Percentage of responses**
10-20	56
20-30	28
30-40	0
More than 40	17

Besides the changes in the overall requirements, 88 percent of the respondents replied that some of their requirements changed repeatedly during the development phase. The project managers cannot remember the exact numbers of repeatedly changed requirement but some of them mentioned that there were around 5 to 10 requirements.

(ix) *Requirement changes causes and its proper handling.* There are different factor that cause the requirement changes in software projects. The project

managers who participated in the case study figured out some causes which had a role in changing requirements. These causes are listed in Table 5.

Even though there are a number of causes for the requirement changes these can be handled properly with good project management practices such as risk analysis, cost and availability schedule etc. Fifty percent of the project managers replied that they implemented the request without considering any of the factors. However, the remaining 50 percent replied that they considered and followed through with the different aspects by discussing with the client and amongst the group members about the necessity of the changes, whilst further analyzing the risk impact, and checking technical and time availability.

Table 5: Causes of requirement change in the case study projects

Causes of requirement changes	**Percentage of responses**
Client needs change	38
Changes in development environment and organization structure	19
Group members necessity and opinion	25
Hardware and software availability	13
Lack of communication with client	6

(x) *Risk planning and management.* Proper project management practices also include the identification of risk and its minimization. A successful project always identifies possible risk in the planning phase and prepares management plans to handle the identified risk. From the case study, it can be concluded that in more than 50 percent of projects risks were identified, assessed and prioritized in the planning phase. In these projects, project managers immediately took the suitable action to minimize, monitor, and control the identified as well as from unidentified risks. According to the results from the case study, 28 percent of projects faced numbers of unidentified risks but handled them efficiently on runtime. In only 17 percent of the projects all the risks were identified and handled properly where as in six percent of the projects were seriously affected from the unidentified risks. The risks along with the percentages of projects in which they occurred are listed in Table 6.

Table 6: Risk occurred in the case study projects

Risks	**Percentage of responses**
Participant dropping out of the course	12
Member unable to give promised time (illness, other courses, exam)	28
Member not serious about project	26
Resources unavailable on time (e.g. hardware, software, working space)	8
Project larger to finish during course time	4
Bias working environment (differentiate team member's working environment)	0
Technology problem (Unknown about technology, server crash, design error)	14
Client unavailable for project	4
Supportive guidance unavailable from supervisor	0
Other	4

(xi) *Gold plating*. This term refers to the addition of hardware and software, the modification of design, and the addition of requirements, which were not mentioned in the project plan to satisfy the clients or with the intention to make the project more attractive. The additions of extra requirements drag the project out from the initial scope and can be the cause of SC. The case study results showed that project managers and their group also practiced gold plating in their projects. The percentages of projects, which added features not included in their initial plan (i.e., gold plating) are shown in Figure 13.

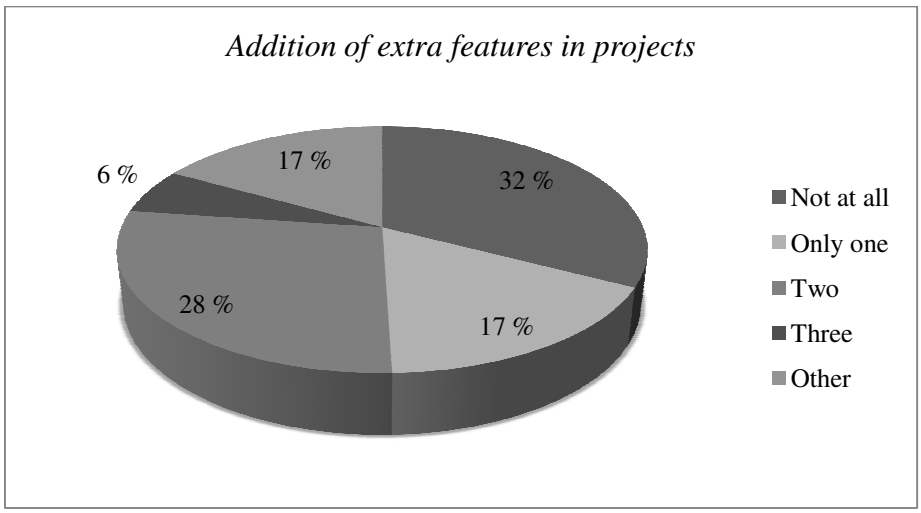

Figure 13: Addition of extra features in the case study projects

5.5. Summary

5.5.1. Problems occurred in the development of the case study projects

Software projects face different challenges during the development process. Proper handling of those challenging factors contributes to the success of a project. From the case study, the challenging factors identified are listed in Table 7.

Table 7: Challenging factors in software projects

S.no	Challenging factors	Case study statistics
1.	Ambiguous project scope	39 percent of the responded project managers were faced with an ambiguous project scope.
2	Less executive support	50 percent of the respondents' received executive support in important phases, whereas six percent rarely got support in their projects.
3	Insufficient resource allocation	In more than 50 percent of the projects two or more than two team members had to share resources.
4	Lack of end-user involvement	In 35 percent of the projects, the end-users were involved in the testing phase and in other 35 percent end-users were not involved during any stage of development lifecycle.
5	Poor estimation of the schedule	50 percent of the projects' overall schedule was affected.
6	Poor practice of project management tools	41 percent of respondent projects did not implement any kind of project management tool.
7	Less experienced team	More than 40 percent of the projects had team members who were either with medium expertise or were beginners.
8	Inactive team members	55 percent of the projects had team members who were inactive in group meeting.
9	Unfair work distribution	In 50 percent of the projects, work was assigned according to the interest of team members.
10	Ineffective communication	In 28 percent of the projects, the project managers communicated with their team members only through face-to-face meeting.
11	Vague and incomplete requirements	Only 17 percent of the projects got a complete set of requirements from the client.
12	Change in customer needs	In 38 percent of the projects, the requirement was changed due to a change in the needs of the client.
13	Change in Environment, technology and organization structure	19 percent of the projects suffered from the changes in environment and organizational structure.
14	Platform changes	13 percent of the projects suffered from the changes in hardware and software platform.
15	Lack of change control management	67 percent of the projects implemented new scope without following any change control management.
16	Addition of new features (Gold plating)	68 percent of project added new features in the project to please the customers.

5.5.2. Factors that cause scope creeping

Changes and additions of requirements highly affect the projects by expanding the initially defined project scope, that expansion in the project scope is known as Scope Creeping, there are a number of reasons behind the occurrence of SC. Apart from project managers opinion, the addition of new requirements and the change in requirements in 13 different projects from the University of Tampere developed during the academic year 2012-2013 were also studied in this thesis work [Mäkiaho and Poranen, 2013]. Table 8 shows the requirements statistics for every month starting from September to February for the corresponding projects. The requirements number mentioned in each month is in the following format: New / in progress / Resolved / Feedback / Closed / Rejected.

Table 8: Requirements elicitation statistics in software projects in 2012-2013 [Project Wiki, 2013]

Project (P)	Sep	Oct	Nov	Dec	Jan	Feb
P1	0/0/0/0/0/0	1/0/0/1/0/0	22/4/2/0/2/0	10/10/4/2/8/0	11/10/3/2/15/0	10/9/1/0/20/0
P2	0/0/0/0/0/0	47/5/0/0/0	47/11/7/0/0/0	47/11/7/0/0/0	48/47/21/0/1/1	48/43/29/0/1/5
P3	0/0/0/0/0/0	14/9/7/0/3/0	6/15/19/0/11/3	12/16/21/0/11/3	12/8/39/0/12/3	3/9/49/0/16/3
P4	0/0/0/0/0/0	7/2/0/0/0/0	7/2/2/0/0/0	0/0/7/0/0/0	0/0/0/0/7/0	0/0/0/0/7/0
P5	0/0/0/0/0/0	25/8/6/0/7/0	22/6/5/0/31/4	0/3/5/0/0/0	18/9/5/0/41/4	8/2/28/0/43/5
P6	0/0/0/0/0/0	17/7/1/0/9/0	22/6/0/0/15/1	28/8/2/1/17/3	33/4/1/0/26/3	38/8/5/0/26/3
P7	/0/0/0/0/0	14/2/0/0/0/0	10/6/0/0/0/0	10/6/0/0/0/0	7/9/1/0/0/0	2/8/5/0/0/2
P8	0/0/0/0/0/0	13/0/0/0/0/0	7/5/1/0/0/1	2/4/4/0/16/8	1/1/3/0/21/8	1/1/3/0/21/8
P9	0/0/0/0/0/0	16/1/0/0/0/0	9/4/4/0/0/0	0/3/14/0/0/0	0/6/16/0/0/0	0/1/18/4/0/0
P10	0/0/0/0/0/0	4/1/0/0/0/0	4/1/0/0/0/0	0/3/0/0/0/0	0/3/0/0/0/0	0/0/3/0/0/0
P11	0/0/0/0/0/0	not-reported	13/2/1/0/0/0	13/2/1/0/0/0	7/9/1/0/0/0	2/4/3/7/0/1
P12	0/0/0/0/0/0	0/0/0/0/0/0	27/4/2/1/0/0	26/5/3/2/0/0	27/5/5/2/5/0	0/3/2/2/5/0
P13	0/0/0/0/0/0	25/13/0/0/0/0	23/18/4/0/0/0	22/20/1/3/0/0	22/20/1/3/0/0	19/13/5/10/10/0

From the data in Table 8, we can easily see that a number of requirements were added during the development phases. For example, Project 6 started without any requirements elicitation. Also, around 13 new requirements were added in the last month of development, February. In another example, Project 3 also started without requirements elicitation and 15 new requirements were added in the last month. Referring to the case study, the changes and additions of these new requirements deviate these projects from their initial project scope causing SC.

The opinion about requirement changes and SC were also collected from the project managers who participated in the case study. Ninety-four percent of the project managers agreed that some of the requirements were changed in their projects. Among them 17 percent replied that more than 40 percent of their requirements were changed where as 28 percent replied that only 20 to 30 percent of their requirements were changed. From the case study, it was seen that 39 percent of the project managers experienced the SC in their project. This data supports the fact that changes in requirements increase the chances of SC. The project managers also replied that SC affected their project by altering the project schedule (29 percent), cost (10 percent), quality (33 percent) and team morale (33 percent). The project managers who experienced SC in their project also pointed out different factors that can cause SC such as misinterpretation of in the scope of the project, lack of change control, change in the customer requirements, poor analysis of requirement, environment and technology changes, and poor communication. When customers change their needs during the development stages then it can affect the initially defined project scope and thus plays a role in the occurrence of SC.

The different challenging factors that were determined from the survey study were already mentioned in Section 5.5.1. When comparing and analyzing challenging factors identified from the case study with SC causes mentioned in Section 3.1, it was found that most of them were responsible for SC. The causes that somehow link with the identified challenging factors are listed in Table 9.

The ambiguous scope defined in the beginning of the project arises misunderstanding among the stakeholders. This misunderstanding can change the project scope, which may lead to SC. Therefore an ambiguous project scope can be one of the causes of misinterpretation. In the same way, a lack of resources forces the sharing of resources with other projects. This sharing of resources can bring a change in project scope, also causing SC. The late or noninvolvement of the user can cause SC. This is due to the fact that the users' expectations can change the project scope.

Table 9: SC causes in software projects

S. no	SC perspective	Causes of SC	Challenging factors identified in the case study
1	Business SC	Misinterpretation [Kerzner, 2009]	Ambiguous project scope
2		Lack of resources [Gurlen, 2003]	Insufficient resource allocation
3		Not involving the users early enough [Larson and Larson, 2009; Anthes, 1994]	Lack of end-user involvement
4		Underestimating the complexity of problem [Brenner, 2002]	Poor estimation of the schedule
			Poor practice of project management tools
			Less executive support
			Less experience and inactive team members.
			Unfair work distribution
5		Lack of change control management [Kerzner, 2009]	Lack of change control management
6		Poor requirement analysis [Larson and Larson, 2009; Anthes, 1994]	Vague and incomplete requirements
7	Technology SC	Customer requirement changes [Abramovici, 2000]	Change in customer needs
8		Environment and Technology changes [Abramovici, 2000; Gurlen, 2003]	Change in environment, technology and organization structure.
9		Platform changes [Abramovici, 2000; Anthes, 1994]	Platform changes
10		Gold plating [Larson and Larson, 2009]	Addition of extra feature (Gold plating)
11		Lack of formal communication [Kerzner, 2009]	Ineffective communication

Under-estimating the complexity of the problem can be another reason of leading to SC. The case studies' projects faced different challenges such as poor estimation of schedule, poor practice of project management tools, less experienced and inactive team members, and unfair work distribution that lead to SC. These challenging factors can be grouped under the heading of the 'under-estimating the complexity of problem' because these causes can be the result of lack in the studies and understanding of the complexity of the project at the beginning. The identified challenging factors such as ineffective communication, vague and incomplete requirement, change in customer needs, and addition of extra features are also responsible in the occurrence of SC. Besides that, changes in technology can be a cause of SC. Technology change occurs mainly due to the unavailability of a previously planned technology or availability of new technology. In addition, the clients and development organization structure and their power sharing can also be the cause of SC. This is due to the interest and power of different managerial level to change or add the requirements that are not included in the project scope. A proper change control management can handle SC in software project whereas leaving it out can be a cause of SC.

6. Scope creeping minimization

6.1. Best practices for minimizing scope creeping

Project scope is an essential part of software project that includes all the requirements and deliverables, which have to be accomplished to determine if the project is successful or not. Furthermore, the entire projects planning activities are based upon this. The change in the project scope is inevitable in almost all projects' lifecycle. If those changes are not properly managed, controlled and documented then it will result in SC. Therefore, the project manager should perform all the PSM and preventing SC tasks throughout the project lifecycle in order to minimize the risk of SC developing. The stakeholders may have varying interests in the project, which affect their work, organizations, budgets, schedules or resources. Therefore, the project manager must consider all of those interests to maintain the support of stakeholders in the project. Due to this, SC planning and managing is one of the most challenging tasks for a project manager. The preparation of specific management plans, and to strictly follow those plans, can prevent or minimize SC. The preparation and proper implementation of following plans are supportive in minimizing SC.

Scope baseline. The project scope baseline is developed at the beginning of the project. The development team should develop the scope baseline when they have a clear understanding of the project vision. The comment and review from the stakeholders should be considered and include while developing the scope baseline. When the entire needs of stakeholders are collected then it is necessary to prioritize them and receive approval from the stakeholders. The actual requirements are elicited by dividing the approved stakeholders' needs. The project scope baseline contains the approved project scope statement, work breakdown structure, and WBS dictionary. The team establishes the baseline depending upon those components against which the actual project scope will be measured. The scope baseline should be created carefully and deliberately planned to ensure that all the project works are captured. As the project progress, the team should measure and compare the completed work with the previously defined work requirements. If the work is running outside the defined requirements then the development team should immediately take action to bring the project back within the defined baseline. In this way, the proper development of scope baseline protects the project from SC. The measurement of project work and requirements, and comparison with the scope baseline determine project success.

Scope management plan. The scope management plan is developed as a part of the project plan, which defines the process to control and manage the scope during the project lifecycle. This plan clearly states who has the authority and responsibility for managing and controlling the scope. It also describes how the scope will be controlled, measured and verified. A clearly defined scope management plan and its proper implementation support the development team to avoid any kind of SC.

Change management plan. The change management plan is considered as the most important tool in preventing SC. Most of the projects have to face a number of change requests from the stakeholders. Some stakeholders may want to add functionality that is not included in the scope baseline or they may want more work on the project, which will be beneficial for the organization. When the change request arises, then the project manager should ensure that it is well documented and, furthermore, should also conduct an analysis to determine the impact of the changes in scope, time, and cost of the project. The change management plan should clearly state the responsibility and authority to the team, designated change control board; to determine whether the change is necessary or not, and whether it supports the project's goals or not. Sometimes, the purposed changes modify the project's scope baseline, schedule or budget, which needs to be included in all the other documents of the project, by modifying and updating the latest information. The necessary information of the changes should be communicated amongst the stakeholders and the project team. If the necessary changes can be carefully managed and communicated and added to the project, then we can help prevent the project from falling victim to SC.

Configuration management plan. This plan contains the necessary information to manage the changes in the project documentation and tools throughout the project. It is necessary to ensure that all the project documentation and tools are managed based on the original project scope and any approved changes to the scope. This proactive approach to managing project documentation ensures that there is consistency between the scope baseline and any changes in the project scope, to prevent incidents of SC by finding their way into the documentation.

Requirements management plan. This plan defines the process to identify, analyze, document, prioritize, and manage the requirements in the project. The plan supports capturing all the requirements during the planning phase and also prevents the project from changing the requirements in future. This plan shows the way to coincide and fulfill all the collected requirements to make the project successful. Even just this plan can prevent the project from changing requirements but it differs from the change

management plan. It does not include the information about the change control board and their responsibility, and also other change such as environment, technology. It only consists of the plans related to requirement collection, prioritize and documentation.

Variance analysis. Variance analysis is a process of measuring the scope performance against the scope baseline. Some acceptable variance is defined in the scope management plan of every project. When the scope changes then those changes are first compared with the acceptable variance. No action is taken when the changes are within an acceptable variance, otherwise corrective action is taken. If corrective actions are needed then it may also require updating the scope baseline, project plan, or other project documentation, which should be done through the change control process. Variance analysis is an effective tool, which can be used iteratively throughout the project lifecycle to monitor requests regarding scope changes.

6.2. Implementation of software measurement metrics in scope creeping

As I have already discussed, we can utilize several components of the formal project management plan as tools for preventing SC. This can be achieved if we carefully follow the project plan. When a project is in execution phase we have to monitor the project scope to ensure that the project is running within the approved scope baseline. There are some software measurements metrics in practice, which can be utilized to monitor the project scope and implement corrective action if some deviation occurs in the project scope. This thesis work discussed three different metrics, which can be used to monitor and minimize the changes in project scope. These three metrics and their capabilities of minimizing SC are described separately in following sections.

6.2.1. Earned value management

EVM is used to track and measure the performance of project, it has its own procedure to track the project. In this metric the project's actual status are determined at different development stages and then compared with the estimated value. When described in a simple way, a project can be divided into numbers of parts or tasks. The cost and time is then estimated for each task during the project-planning phase. When the project is running, then the actual cost and schedule of every task is collected and then compared with the estimated cost and schedule. The cost and performance indices are determined using different equations (described in Section 4.3.1). The value of those performance indexes answer the two basic questions:

(i) Is the work completed at the time of measurement is more than or equal to the work planned for that interval of time?

(ii) Is the money spent at the time of measurement is less than or equal to the money budgeted for the project to get to the point of delivery?

The answers of these two questions can help to track the project. If both the questions can be answered 'positively' then we can say that the project is on track, whereas, If the answers are 'no' then the project is facing the problem of being over budget and behind schedule. In the same way, if the answer of the first question is 'no' and second question is 'yes' then the productivity of the project is suffering. This is caused by different problems or risks happening in the project. In this case, EVM cannot figure out the actual reasons, which ultimately will decrease the productivity of the project, however, it can show that the running project is facing the problem of lower productivity. The project managers and other stakeholders have the responsibility to determine the causes of lower productivity. A quick response from the project managers and other stakeholders can prevent the projects from over budget and delayed.

If the answer of the first question is 'yes' and second question is 'no' then project is facing the problem of being over budget, the causes of being over budget can be due to poor planning or motivating the team members by providing money as an incentives. In this way, EVM can be implemented to track the project and protect it from SC by determining the schedule and cost status of the running project at any time. If the result shows that project is facing a cost overrun and is behind schedule then the project manager should immediately take corrective action to bring back the project on track. This corrective action prevents the project from SC.

6.2.2. Balanced scorecard

BS is used to track the performance of project. It measures the project from four different perspectives such as customer, financial, internal process, and learning and growth. The measurement of the project through these four perspectives helps to identify the risk that occurred or may occur in the project that could cause SC. The customer perspectives focus on the measurement of the customer's demands and expectations. The customer's expectations and demands can be understood only when they are involved in the project. Therefore, the proper implementation of BS metrics increases the user involvement in the project.

The internal process perspective focuses on the improvement of the organizational internal process. The internal process includes end users identification and their expectation, innovation and use of new technology, proper management of the risk, good communication among the team members, and more. The implementation of the BS metrics helps to track and improve the performance of internal process parameters.

The learning and growth perspective focuses on the appointment of team members in the field where they have expertise. The BS metrics determine the performance of the team members on their assigned work and suggest any training or incentive needed for them. This will motivate the team to complete the work within time and budget. The fourth financial perspective focuses on the financial part of the project. In this perspective, BS metrics compare the estimated cost with the actual cost to determine the financial performance of project. In this way, financial, customer, internal process and learning and growth perspectives cover most of the risk factors that may cause SC in the project.

In the current scenario, any developing system should have the capabilities to identify, describe, and fully harness the intangible assets driving organizational success. The original intent of the scorecard system was to balance historical financial numbers with the drivers of future value for the firm; more and more organizations experimented with the concept and found it to be a critical tool in aligning short-term actions with their strategy [Niven, 2008]. BS improves many issues related to effective strategy implementation. Furthermore, it removes some barriers related to effective strategy implementation and supports the project to be more successful. Some of those barriers, and how the BS system works, and removes them, are described below [Niven, 2006]:

- *Vision barrier.* The vision barrier means there is misunderstanding or complexity in the project strategy. BS is developed by sharing understanding and the translation of the organization's strategy into objectives, measures, targets and initiatives in each of its four scorecard perspectives. During the translation of the vision and strategy process, the executive team has to make it clear that there are not any misleading or vague terms remaining in the project strategy. The use of BS supports the success of project by providing a clear direction to all team members in achieving the required goal.
- *People barrier.* A plan is considered a successful plan only when it is understood and implemented at every level of development. BS provides equal opportunities to each team member to demonstrate how their daily activities support the project plan. From this, the project manager can determine the expertise and the interest area of the team members. A proper implementation of BS removes the barrier between the stakeholders and supports in the development of a successful project.
- *Resources barrier.* The development of BS provides an excellent opportunity to link the project objectives and resources. The development and

implementation of BS forces the project manager, not only to think about objectives, measures and targets but also to consider the allocation of all human or non human resources to achieve the objectives. BS also supports in determining the scarcity of resources to achieve the goal by reviewing all implemented resources.

- *Management barrier.* In the current scenario, most of the strategic decisions depend upon the analysis of actual achievement. This analysis is actually done by determining variance between planned and actual value. Unfortunately, many management teams spend their time together discussing variances and finding ways to correct the defects. BS provides the necessary elements to move away from this paradigm to a new model in which scorecard results become the starting point for reviewing, questioning, and learning about the strategy.

The successful handling of these barriers also prevents the project from developing SC. From the above explanation, we can conclude that BS can be considered as one suitable metric to handle SC.

6.2.3. Requirement metrics

A proper requirements definition plays a great role in the success of the project. An incorrect or a poorly defined requirement decreases the quality of product even when it has well defined code. The requirement metrics play a vital role in analyzing the quality of requirements, collecting valuable requirements, and determining the causes of software failure. The practice of requirements metric increases the quality of the product, and decreases the project failure by minimizing the chance of SC. There are a number of requirement metrics in practice. Among them, RTM as well as requirement volatility metrics are fruitful in minimizing SC in software projects. How these two metrics play role in minimizing SC are described below:

(i) *Requirement traceability metric.* All the project requirements are documented in the beginning of the project in order to understand the goals of the project, communicate them with the team members, and to ensure that the project is completed in the end. However, only the well documentation of the requirements is not sufficient to monitor project scope and prevent the project from SC. In every project there must be a clear understanding of each requirement and accountability or ownership of each requirement. The project manager uses the requirements documentation and traceability metrics to establish understanding and ownership of each requirement and to track the

completion of each requirement before the new system is implemented. Generally, the project-analysis team or project manager tracks the requirements through the various stages, beginning with tracing the requirement back to the initial justification. The proper implementation of RTM has following advantages:

- *Understanding the need.* The requirements for the software design can be understood clearly only when we can get a clear knowledge of the user and the underlying principles. Either the project manager, or the requirement management team of the project, must be able to track the requirement back to the need, which is an essential component of the project. With the proper examination of the needs, the project manager or the team can identify the missed requirements early in the development process. The metric also helps in recognizing the extra requirements that are not really needed.
- *Anticipating changes.* RTM also supports keeping track of what happened and when changes are implemented without redesigning the system. This helps the project manager to anticipate what should take place in the company to successfully adjust to the change. The effective traceability measures give a better understanding of what kind of changes are needed. The tracing of the requirements at the different stages shows that whether the requirement has been successfully addressed or not.
- *Streamlining the testing phase.* RTM has a capability to assist the project team in what areas the requirements must be tested. The testing of each requirement approach is impractical because it is time consuming and an expensive process. The testing is based on the risk that may cause a problem, as well as the impact on the organization if a particular problem occurs. Generally, high-priority requirements are traced for testing purpose.
- *Project success assurance*. RTM supports in avoiding unnecessary requirements, aids in guaranteeing the project completion, controlling the cost, and preventing the project from delays. It assures that there will be adequate resources of time, manpower, and money available to code, test and verify project requirement throughout the project development lifecycle.

Overall, RTM gives the indication of the occurrence of SC in the project if any work performed or requested outside the requirements that are documented. This metric provides an organized method for monitoring scope and ensuring all project work supports an approved and documented requirements.

(ii) *Requirement volatility metrics.* The requirement volatility metric supports in the measuring of the numbers of requirements added, deleted and modified. They also determine the causes of adding, deleting and modifying the requirements and classified those requirements with the reason of change. A proper implementation of the requirement volatility metric has following advantages:

- *Determine the number of initially allocated requirements.* The requirement volatility metric measures the number of initially allocated requirements. It includes all the technical and non-technical requirements provided by the customer. This metric describes the level of requirements volatility along with the number of final allocated requirements as well as the number of changes allocated per requirement.
- *Determine the final requirements.* This metric measures the numbers of finally allocated requirements. It also includes all technical and non-technical requirements to build the final software products.
- *Track the number of changes per requirement.* This metric tracks the number of changes made to each requirement. Along with describing the level of volatility of the requirements, this metric also describes the impact of changing the requirements in the software process.
- *Number of changes in specific time period.* This metric contains a number of changes of requirements for specific time period, such as week and month. It describes the degree of the volatility of the requirements. Its value should decrease towards the end of the software lifecycle (indicating convergence of requirements). They are measured during the project lifecycle.
- *Causes of change.* This metric collects all the causes of requirement changes and categorizes them. It helps in identifying the most common causes of change in the software process and can be used to improve the software process.
- *Who requested the change.* This metric helps in identifying the source of the change, the reason for implementing a specific functionality, and

anticipates the source of changes in the future.

Due to these specific functionalities of the requirement volatility metric, it is suitable to determine the causes of scope changes and prevent the project from SC.

In a conclusion, the causes of SC with the recommended metrics and management plans are listed in Table 10.

Table 10: Recommendation to minimize SC

S.no	SC causes	Recommended metrics and plan
1	Ambiguous project scope	BS, Scope baseline
2	Insufficient resource allocation	BS, Requirement metrics
3	Lack of end-user involvement	BS, Requirement metrics
4	Underestimating the complexity of problem	EVM, Scope baseline
5	Ineffective communication	BS, Requirement metrics
6	Vague and incomplete requirements	Requirement metrics, Requirement management plan
7	Change in customer needs	Requirement metrics, Requirement management plan
8.	Change in environment, Technology and organization structure	BS, Configuration management plan
9	Platform changes	BS, Configuration management plan
10	Lack of change control management	Change control management plan, Requirement metrics and Requirement management plan
11	Addition of extra feature	Change control management, BS and Requirement metrics

7. Conclusion

Project scope is a basic foundation upon which an entire project process is developed. A proper PSM can determine the success of project. An incomplete and poorly defined project scope faces different problems during the project lifecycle. The changes of the project scope during the development cycle increases the chances of the failure of the project by affecting project schedule, cost and quality. There are a number of reasons behind the deviation of the scope. In this study, the different factors that hinder the project scope in the development cycle have been determined. A case study as a part of this thesis work was been conducted among the students, who worked as project managers during the software project management course at the University of Tampere. All the projects considered were real projects and related to the company and university research unit. All the factors that cause SC collected from the case study are found to be mostly the same as the causes determined in past research and studies. It must be remember that the change of the project scope is inevitable in the software development process, but it can be controlled or minimized to protect the project from SC. In this thesis work, management plans and software measurement metrics are identified to minimize SC. There are a number of management plans that can be developed at the beginning of a project such as scope baseline, change management plan, requirement management plan and more. A proper and clear development of management plans beforehand can help to protect the software project from SC.

This thesis recommends three different software measurement metrics to minimize SC. Among them, EVM metrics is one, which focuses on the cost and schedule of the projects. EVM determines the cost overrun and delay in the project by comparing actual and estimated cost and schedule respectively of the completed and remaining tasks. The cost and schedule data obtained from the metrics provide the warning about the possibility of SC. The next suitable metric proposed in this thesis work is BS. This metric measures the software project from four different perspectives: customer, financial, internal process, and. learning and growth. A proper implementation of this metric improves users' involvement, distribution of the resources, communication, executives' support, team motivation, training and many more. In addition, a proper improvement of these factors might decrease the possibility of SC. The final metrics that play an important role in protecting project from SC are requirement metrics. The proper requirement metrics to control SC are requirement traceability and requirement volatility metrics. These metrics properly track the requirements and document any

change that occurs in the requirements along with the factors that cause the changes. Apart from these, the metrics have the capabilities to relate the changes with the needs of the project, which support the project management team to determine the necessity of those changes.

Finally, it is concluded that even though there are a number of factors causing SC, a proper implementation of software measurement metrics and preparation of the different management plans beforehand can prevent or minimize the chances of SC in a software project.

7.1. Limitations and future enhancement

This thesis work came up with conclusion about how SC can be minimized. However, there are some limitations, which are described below:

- SC can highly affect the budget which can be illustrated by comparing estimated budget with the actual budget of the project. But, the case study completely excluded the cost estimation part of the software project development because case study project focus more on schedule than cost.
- The case study was conducted with a small group of project managers. If it had been conducted in a large group of project managers, the possibility of finding more unidentified causes of SC would have increased.
- Highly experienced project managers would have a better understanding of SC and potential solutions. However, most of the project managers who participated in the case study possess very little experience in project management.
- The thesis work recommended some of the metrics for controlling SC but did not focus on the available requirement management tools and their effectiveness in controlling SC.

There is some room for more work, which could be carried out in the future to build upon the foundations of the work started in this thesis. The affect of SC on budget can be analyzed by considering the cost estimation part of the project. Besides, more work can be done to identify the proper requirement management tools available on the market which support in controlling SC. This can be done by in-depth study of the available requirement measurement tool, and analyzing its pros and cons. In addition, further work can be carried out to develop the different management plan templates, which can support in minimizing SC.

References

[Abramovici, 2000] Adrian Abramovici, Controlling scope creeping. *PM Network* **14** (1), 2000, 44-48.

[Anthes, 1994] Gary H. Anthes, No More Creeps. *Computerworld Framingham* **28** (18), 1994,107-110.

[Attarzadeh and Hock, 2009] Iman Attarzadeh and Ow Siew Hock, Implementation and evaluation of earned value index to achieve an accurate project time and cost estimation and improve earned value system. In: *Proc. of International Conference on Information Management and Engineering, IEEE*, 2009, 312-316.

[Babu, 2005] Suresh Babu, Scope creep management. Project Perfect white paper collection, 2005. Available as *http://www.projectperfect.com.au/info_scope_creep_mgmt.php*

[Bashir and Thomson, 1999] Hamdi A. Bashir and Vince Thomson, Metrics for design projects: A review. *Design studies* **20**,1999, 263-277.

[Belassi and Tukel, 1996] Walid Belassi and Oya I. Tikel, A new framework for determining critical success/failure factors in projects. *International Journal of Project Management* **14** (3), 1996, 141-151.

[Brenner, 2002] Rick Brenner, Some causes of scope creep. *Point Outlook* **2**(36), 2002.

[Bronstein, 2010] Nadya Bronstein, Scope creep. November 15, 2010. Available as *http://www.umsl.edu/~sauterv/analysis/Fall2010Papers/Bronstein/scope%20creep.html*

[Bundschuh and Dekkers, 2008] Manfred Bundschuh and Carol Dekker, *The IT Measurement Compendium*. Springer Berlin Heidelberg, 2008, 179-206.

[Carlos, 2010] Tom Carlos, Requirement traceability matrix. Thomas Carlos Consulting, 2010. Available as *http://www.carlosconsulting.com/downloads/RTM.pdf*

[Capatina and Crista, 2011] Alexandru Capatina and Dragos Cristea, Comparative analysis of two software companies performance using balancec scorecard as a competitive intelligence tool. In: *Proc. of 6th European Conference on Innovation and Entrepreneurship*. Academic Publishing Limited, 2011, 183-189.

[CHAOS, 2013] The Standish group, CHAOS Manifesto 2013. *The Standish Group International*, 2013.

[Collegiate-project, 2009] Collegiate-project, Managing scope creeping. Available as *http://www.collegiateproject.com/articles/Managing%20Scope%20Creep.pdf*

[Conchúir, 2012] Dr. Deasún Ó Conchúir, *Overview of the PMBOK® Guide*. Springer Berlin Heidelberg, 2012, 59-74.

[Costello and Liu, 1995] Rita J. Costello and Dar-Biau Liu, Metric for Requirements Engineering. *Journal of Software system* **29**, 1995, 39-63.

[Dwivedi, 2009] Umesh Dwivedi, Earned value management explained. *Project Smart*, 2009. Available as: *http://www.projectsmart.co.uk/earned-value-management-explained.html*

[Elliott, 2007] Bill Elliott, Anything is possible: Managing feature creep in an innovation rich environment. In: *Proc. of Engineering Management Conference, IEEE*, 2007, 304-307.

[Fenton and Pfleeger, 1997] Norman E. Fenton and Shari L. Pfleeger, *Software metrics: A Rigorous & Practical Approach*. PWS Publications, 1998.

[Goodman, 1993] Paul Goodman, *Practical Implementation of Software Metrics*. McGraw Hill, 1993.

[Greiman, 2013] Virginia A. Greiman, *Megaproject Management: Lessons on Risk and Project Management from the Big Dig*. John Wiley and Sons, 2013, 152-184.

[Grossman, 2003] Ira Grossman, Why so many IT project fail, and how to find success. *Financial executive* **19** (3), 28.

[Gube, 2008] Jacob Gube, Eight tips on how to manage feature creep. February 6, 2008. Available as *http://sixrevisions.com/project-management/eight-tips-on-how-to-manage-feature-creep/*

[Guo et al., 2009] Ying Guo, Meihong Yang, Jun Wang, Ping Yang and Feng Li, An Ontology based Improved Software Requirement Traceability Matrix. In: *Proc. of Second International Symposium on Knowledge Acquisition and Modeling, IEEE*, 2009.

[Gupta, 2011] Shilpa Gupta, Scope creep- causes and solution. Available as *http://www.shilpagupta4.com/2011/08/09/scope-creep-causes-and-solution/*

[Gurlen, 2003] Stephanie Gurlen, scope creep. December 2, 2003. Available as *http://www.umsl.edu/~sauterv/analysis/6840_f03_papers/gurlen/*

[Humphery, 2005] Watts S. Humphery, Why Big Software Project Fail: The 12 Key Questions. *The Journal of Defense Software Engineering*, March 2005.

[Hussain, 2012] Osama Hussain, Direct cost of scope creep in governmental construction projects in Qatar. *Global Journal of Management and Business Research* **12** (12), 2012.

[Inder and Rivera, 2007] Dale Inder and Mark Rivera, Scope creep-A lethal project disease thoughts on prevention and cure. *Integrated Management Systems*, 2007.

[Kaplan and Norton, 1996] Robert S. Kaplan and David P. Norton. *The balanced Scorecard: Translating Strategy into Action*. Harvard Business School Press, 1996

[Kaplan and Norton, 1992] Robert S. Kaplan and David P. Norton, The balanced scorecard-measures that drive performance. *Harvard Business Review*, 1992, 71-79.

[Kaplan and Norton, 2007] Robert S. Kaplan and David P. Norton, Using the Balanced Scorecard as a Strategic Management System. *Harvard Business Review*, 2007, 75-85.

[Kaplan, 2009] Robert S. Kaplan, Conceptual foundation of the balanced scorecard. In: Christopher S. Chapman, Anthony G. Hopwood, and Michael D. Shields (eds.), *Handbook of Management Accounting Research* **3**. Elsevier, 2009, 1257-1269.

[Kerzner, 2009] Harold Kerzner, Project Management: *A Systems Approach to Planning, Scheduling, and Controlling*. John Wiley and Sons, 2009.

[Knight et al., 2012] Joe Knight, Roger Thomas, Brad Angus and John Case, Project Management for Profit: *A Failsafe Guide to Keeping Projects On Track and On Budget*. Harvard Business Press Books, 2012.

[Kuprenas and Nasr, 2003] John A. Kuprenas and Elhami B. Nasr, Controlling design-phase scope creep. In: *AACE International Transactions*, 2003.

[Larson and Larson, 2009] Richard Larson and Elizabeth Larson, Top Five Causes of Scope Creep...and What to Do About Them. In: *PMI Global Congress Proceedings*, 2009.

[MacLellan, 2007] Sister B. MacLellan, The Balanced Scorecard□Developing, Measuring, and Communicating Organizational Strategy. In: *CHAO Continued Strategic Planning Process*, 2007.

[Mäkiaho and Poranen, 2012] Pekka Mäkiaho and Timo Poranen, Software Projects 2011□2012. University of Tampere, School of Information Sciences, Report R9-2012, April 2012.

[Mäkiaho and Poranen, 2013] Pekka Mäkiaho and Timo Poranen, Software Projects 2012□2013. University of Tampere, School of Information Sciences, Report R23-2013, October 2013.

[Money marketing, 2010] Money marketing, Business strategy: Seconds out for hourly fees. *Money Marketing* **35**, 2010 (ID: 2128191191).

[Nagrecha, 2002] Suketu Nagrecha, An introduction to Earned Value Analysis. March 16,2002. Available as *http://www.pmiglc.org/COMM/Articles/0410_nagrecha_eva-3.pdf*

[Niven, 2006] Paul R. Niven, *Balanced Scorecard Step-by-Step: Maximizing Performance and Maintaining Result*. John Wiley and Sons, 2006.

[Niven, 2008] Paul R. Niven, *Balanced Scorecard Step by Step for Government and Non-profit Agencies*. John Wiley and Sons, 2008.

[Osmundsona et al., 2003] John S. Osmundsona, James B. Michaelb, Martin J. Machniakc and Mary A. Grossmand, Quality management metrics for software development. *Information & Management* **40**, 2003, 799–812.

[Pinto and Mantel, 1990] Jeferry K. Pinto and Samuel J. Mantel, The causes of project failure. *IEEE Transactions on Engineering Management* **37** (4), 1990, 269-276.

[Piscope, 2013] Mark Piscope, How to manage scope creep. Available as *http://www.projectmanagementdocs.com/blog/how-to-manage-scope-creep.html*

[PMI, 2009] Project management institute, *A Guide to the Project Management Body of the Knowledge (PMBOK Guide)*, Project Management Institute, 2009.

[ProjectWiki, 2013] ProjectWiki. University of Tampere, School of Information sciences, 2013. Available as *https://projectwiki.sis.uta.fi/wiki/Statistics_2012-2013 (Retrieved on April 2013).*

[Rehman et al., 2010] Israr U. Rehman, Sajid ullah, Abul Rauf and Arshad Ali Shahid, Scope Management in Agile Versus Traditional Software Development Methods. In: *Proc. of National Software Engineering Conference, Proceedings of the 2010 National Software Engineering Conference*, *J. ACM Article* **10**, 2010.

[Rohm, 2008] Howard Rohm, Using balanced scorecard to align your organization. *Balanced Scorecard Institute*, 2008.

[Schwalbe, 2007] Kathy Schwalbe, *Information Technology Project Management*. Thomson Learning, 2007.

[Singh et al., 2011] Gurdev Singh, Dilbag Singh and Vikram Singh, A Study of Software Metrics. *IJCEM International Journal of Computational Engineering & Management* **11**, 2011, 22-27.

[Sliger and Broderick, 2008] Michele Sliger and Stacia Broderick, *The Software Manager's Bridge to Agility*. Pearson Education, 2008, 67-82.

[Thommazo et al., 2012] Andre D. Thommazo, Gabriel Malimpensa , Thiago R. de Oliveira and Sandra C. P. F. Fabbri, Requirements Traceability Matrix: Automatic

Generation and Visualization. In: *2012 Brazilian Symposium on Software Engineering, IEEE*, 2012.

[Vandevoorde and Vanhoucke, 2006] Stephan Vandevoorde and Mario Vanhoucke, A comparison of different project duration forecasting methods using earned value metrics, *International Journal of Project Management* **24**, 2006, 289-302.

[Veryard, 2001] Richard Veryard, *The Component-Based Business: Plug and Play*, Springer London, 2001.

[Werner and Xu, 2011] Michael L. Werner and Fuyuan Xu, Successfully executing strategy by implementing the balanced scorecard. In: *Proc. of Computer and Management (CAMAN) International Conference*, 2011, 1-5.

[Woods, 2012] Marian Woods, Project scope management. December 11,2012. Available as *http://corasystems.com/blog/2012/12/11/project-scope-management/*

[Yang, 2009] Jian-hua Yang, A balanced performance measurement scorecard approach for product service systems. In: *Proc. of International Conference on Business Intelligence and Financial Engineering, IEEE*, 2009, 548-551.

[Zuber, 2013] Laura Zuber, Project metrics are the best defense in the battle against scope creep, 2013. Available as *http://www.qsm.com/blog/2013/project-metrics-are-best-defense-battle-against-scope-creep*

[Zuse and Bollmann, 1989] Host Zuse and Peter Bollmann. Software metrics: using measurement theory to describe the properties and scales of static software complexity metrics. *IEEE Transactions on Software Engineering* **24**, 1989, 23-33.

Appendix I

Definition of the terms used in the following question

Scope creep: Scope creep is defined as the extra expansion in the scope of a project due to the changes and addition of the requirements that are not included in the initial planning phase of the project (PMBOK).

Project scope: The work that needs to be accomplished to deliver a product, service, or result with the specified features and functions (PMBOK).

Requirement elicitation: Requirements elicitation is the practice of collecting the requirements of a system from users, customers and other stakeholders (Requirements Engineering: A good practice guide, Ramos Rowel and Kurts Alfeche).

General Information

1. Name of project _____
2. Your role in the project are project manager plus (can choose multiple)
   - ☐ Developer
   - ☐ Designer
   - ☐ Tester
   - ☐ Customer
   - ☐ User
   - ☐ Others (please specify)_____
3. Project development method used in project
   - ☐ Scrum
   - ☐ Extreme programming
   - ☐ Waterfall model
   - ☐ Spiral Model
   - ☐ Others (please specify)_____

Project objectives and planning

4. The project scope (list of project goals, deliverables, tasks, costs and deadlines) defined in the beginning of the project is
   - ☐ Scope was well defined
   - ☐ Defined scope was ambiguous
   - ☐ Scope defined was unrealistic and unachievable
   - ☐ Others (please specify)_____
5. How you defined the client expectations regarding the project

☐ Project objectives were achievable with provided time and resources (human and non-human)

☐ Project objectives were not achievable with provided time and resources

☐ Others (please specify)_____

6. How you defined the stakeholders (executives) support during the project development.

   ☐ Received constructive feedback in every development stage.

   ☐ Received support only during important phase of the development stage

   ☐ Rarely received support

   ☐ Received no support

7. Rate the allocation of the resources during the project development

   ☐ Team members had their individual set of resources (all necessary resources provided)

   ☐ Two team members had to share one set of resources (all necessary resources provided)

   ☐ More than two team members had to share one set of resources (all necessary resources provided)

   ☐ Some necessary resources were never provided

   ☐ None of the necessary resources were provided

8. The involvement of the clients in the project development.

   ☐ From the beginning

   ☐ In testing phase

   ☐ Only in some piece of work

   ☐ Not at all

9. The involvement of the end users in the project development.

   ☐ From the beginning

   ☐ In testing phase

   ☐ Only in some piece of work

   ☐ Not at all

10. Scale the schedule estimation of the project

    ☐ All tasks completed exactly on estimated schedule

    ☐ Some tasks did not meet estimated schedule, but in a whole it did not affect the project schedule

    ☐ Some tasks did not meet estimated schedule, but in a whole it slightly affected the project schedule

☐None of the tasks met the estimated schedule, but in a whole it did not affect the project schedule

☐None of the tasks met the estimated schedule, but in a whole it slightly affected the project schedule

☐None of the tasks met the estimated schedule, and highly affected the project schedule

11. Have to practice any project management tool to carried out management activities

☐No (move to question 13)

☐Yes (Mention the name of the tool and answer question 12)_____

12. What was the role of the project management tool in your project (select multiple)

☐Checking progress

☐Managing time budget

☐Share resources

☐Discussion

☐Change management

☐Others (please specify)_____

13. Do you add any specification such as hardware and software, modify design and add more requirements which was not mention in the project plan to satisfy the client and made project more attractive?

☐Only one

☐Two

☐Three

☐Not at all

☐Others (how many, please specify)_____

14. Have you feel any of these causes occurred scope deviation in your project (can choose multiple)

☐Misunderstanding of project scope among the stakeholders

☐Change in working model

☐Change in technical platform

☐ Arrival of new technology

☐Others (please specify)_____

Team member's expertise and work division

15. What is your opinion about team competence (knowledge about programming, testing, other activities)
    - ☐ All expert
    - ☐ All medium level
    - ☐ All learners
    - ☐ Expert and medium level mixed
    - ☐ Expert and learner mixed
    - ☐ Medium level and learner mixed
    - ☐ Expert, medium level, and learner mixed
16. Work division in team was based on
    - ☐ Personal interest
    - ☐ All has to be involved in everything
    - ☐ Project manager decide who should do and what
    - ☐ One with knowledge does everything
    - ☐ Nobody does anything, and project manager does everything
    - ☐ Others (please specify)_____
17. Participation of team members in discussion and planning
    - ☐ All active
    - ☐ All inactive
    - ☐ Few active and most inactive
    - ☐ Most active and few inactive
    - ☐ 50% active and 50%inactive
    - ☐ Others (please specify)_____
18. What strategic you followed to motivate team members
    - ☐ Organize training for team member
    - ☐ Assign the work on which they have interest
    - ☐ Nothing
    - ☐ Others (please specify)_____

Communication

19. Medium used for communication in team and with client (can choose multiple)
    - ☐ Social media (e.g., Facebook, Google+)
    - ☐ Mobile and phone
    - ☐ Face to face meeting
    - ☐ Instant messaging (e.g., IRC, Messenger, Skype)

☐ Discussion board in SPM tool

☐ Others (please specify)_____

20. How often do you in team (using any medium)

☐ Everyday

☐ Only during the face to face meeting

☐ Whenever necessary

☐ Not at all

21. Group meeting per week

☐ Once

☐ Twice

☐ Thrice

☐ More than thrice (please specify)_____

☐ Not a single time

22. Frequency of project progress review with the client

☐ Once in a week

☐ Once in a month

☐ Not at all

☐ Others (please specify)_____

Risk planning and management

23. Kind of risk that occurred in project (can choose multiple)

☐ Member dropping the course

☐ Member unable to give promised time (illness, other courses, exam)

☐ Member not serious about project

☐ Resources unavailable on time (e.g. hardware, software, working space)

☐ Project larger to finish during course time

☐ Bias working environment

☐ Technology problem (Unknown about technology, server crash, design error)

☐ Client unavailable for project

☐ Supportive guidance unavailable from supervisor

☐ Others (please specify) _____

24. Scale risk planning and management in team

☐ All risks were identified, assessed and prioritized beforehand and suitable actions were taken to minimize, monitor, and control them

☐ Most of the risks were identified, assessed and prioritized beforehand and suitable actions were taken to minimize, monitor, and control them. Even unidentified risks were handled efficiently on runtime

☐ Many unidentified risks arose, but they were handled efficiently on runtime

☐ Many unidentified risks arose, and seriously affected the project

☐ None of the risks were identified and seriously affected the project

Requirement Elicitation

25. Who is responsible for the requirement elicitation

☐ Project manager only

☐ Team only

☐ Manager and team together

☐ Received complete detail of requirements from client

☐ Others (please specify)_____

26. How the requirement were collected

☐ Get directly from the client

☐ By interview with client and end user

☐ By brainstorming among the group members

☐ Analyzing the existing documents

☐ Others (please specify) _____

27. How were requirements documented?

☐ Using a requirements management tool (if yes, please specify)_____

☐ Using a word processing tool (such as MS office) to document the requirement in a free form

☐ Using a software requirement specification template to document requirement in a word process tool (if yes, please specify the template)_____

☐ Others (please specify)_____

28. How did you analyze the elicited requirements?

☐ Discussing with the client

☐ Discussing among the group members

☐ Analyzing and comparing with the project scope definition

☐ According to the demand of the end user

☐ Others (please specify)_____

29. How many requirements were documented?

_____

30. What is your opinion about final set of requirements of your project?

- ☐ It covered all client needs
- ☐ It partially covered the client needs
- ☐ None of the clients needs were covered (please specify reason s)_____
- ☐ Others (please specify)_____

Requirement change management

31. What were the causes of requirement changes (can choose multiple)

- ☐ Client needs changes
- ☐ Changes in development environment and organization structure.
- ☐ Group member's necessity and opinion
- ☐ Hardware and software availability
- ☐ Others (please specify)_____

32. Requirements change frequency

- ☐ No change, the first time was the last time
- ☐ Changed few times
- ☐ Changed quite frequently

33. How many percentage of the requirements were changed during the development

- ☐ 10-20
- ☐ 20-30
- ☐ 30-40
- ☐ more than 40

34. Was there any requirement which changes repeatedly during the development

- ☐ No
- ☐ Yes
    i. Please mention the number of requirement_____

35. How did you deal with the requirement change request

- ☐ Implement the request without considering anything
- ☐ Follow the change control process (please specify)_____

36. How you handle the situation when project deviate from the initial project scope.

- ☐ With proper change control management
- ☐ Nothing done

☐ Just follow the new scope

☐ Others (please specify)_____

Scope Creep

37. Do you have a clear understanding of the term "Scope creep"?

☐ Yes

☐ No (definition is at the beginning)

38. Did you experience SC in your current or previous projects

☐ Yes (Answer question 39)

☐ No

39. If your answer was 'yes' for question no. 38,

a. Project type is/was

☐ Commercial

☐ Open source

☐ Others (please specify)_____

b. Project size is/was (you can select the size considering lines of code (LOC) and numbers of requirements)

☐ Small

☐ Medium

☐ Large

c. Who is responsible for controlling the project scope?

☐ Project manager

☐ Sponsor

☐ Project team

☐ Collective responsibility

d. From the project records, what were the major problem that causes scope creeping in your project (can choose multiple)

☐ Misinterpretation of what is contained in project scope

☐ Lack of change control

☐ Customer requirement changes

☐ Environment and technology changes

☐ Poor requirement analysis

☐ Others (please specify)

_____

e. From your experiences what more factors can cause SC?

_____

f. SC mostly effect (can choose multiple)

- ☐ Schedule
- ☐ Cost
- ☐ Team moral
- ☐ Results quality
- ☐ Others (please specify)_____

Thank you very much your participation!