# The Role of Software Configuration Management and Capability Maturity Model in System Quality

**Zaffar Ahmed Shaikh[1*], Asif Ali Wagan[2], Asif Ali Laghari[2], Karar Ali[1], Mashooq Ahmed Memon[1], Anwar Ali Sathio[1]**

[1]Benazir Bhutto Shaheed University, Lyari, Karachi, Sindh, Pakistan
[2]Sindh Madressatul Islam University, Karachi, Sindh, Pakistan

**Abstract**
The need to improve system quality and how an organization can achieve a higher level in terms of system quality are the major issues discussed in this work. The system quality has been regarded as the most important element of a computer system. This paper discusses how Software Configuration Management (SCM) and Capability Maturity Model (CMM) contribute towards achieving better system quality. Though both concepts have different domains in which they perform towards their primary goal, however, both the concepts have a tremendous and direct impact on the quality of the system. The paper finds out answers to how do we measure system quality? Is there any way we can clearly state about a system or an organization to be comparable in terms of quality? What parameters need to be checked to claim that the system is of a high-quality? It has become evident from previous research that change is important if we want system quality to improve. Besides, the change in the process cannot take place without proper management. Thus, this paper also discusses how the improvement within the organization structure contributes to system quality. It finds out answers to how does an organization works throughout the levels to improve its processes? Does it make any difference for an organization to improve its level? What is the importance of a level of improvement for an organization? How does the organization's maturity impact system quality in a positive way? In the end, the paper informs policy, procedures, and practices to improve system quality.

***Key words:***
*Change, change management, capability maturity model, CMM, SCM, Software configuration management, system quality, organization quality.*

## 1. Introduction

Before we start the discussion on how SCM and CMM work towards the desired system quality, it is also important to know why system quality is important. The quality of the system does not mean a system that is not malfunctioning in terms of its hardware or minor errors, but it is the continuous improvement of its process in a way that is more refine and accurate. The processes must also have several other properties like security and efficiency [1].

It is also true that the end-user is naïve without having knowledge of the underlying structure. This is also a fact that for a naïve user, the meaning of the system quality is different when compared to a developer who takes the system quality differently [2]. Here comes the most important question when we discuss the above topic. How do we measure the system quality? Is there any way we can clearly state about a system or an organization to be better when compared to another? And can we improve a system's quality? The answer to all the questions is yes. There are some parameters, and on the basis of those, we can state about a system and the organization to have more quality. While discussing the change within the system and organization to make it better, there must be a controlled changeable scenario rather than enforcing a change without knowing the impact. Is there any controlled scenario? The answer is yes. We shall discuss that in the upcoming topic abut a proper change flow that takes in place, including the need for change and change management [3].

## 2. Software Quality

The one questions come in a person's mind on how we can measure the system quality. What are the parameters that we can claim a system to be a higher quality system? There are certain quantitative and qualitative measures to know about a system's quality.

Once we are well-aware of how a system quality is distinguished, we can learn on how to improve that further. The quality here does not only describe the system quality, but it also talks about a broader organization level measure [4].

System quality, according to a naïve user's point of view, is quite different than it is. A system with higher quality will be more reliable and efficient. According to a naïve user, the system with good quality is which fulfills certain commands and outputs some results. A naïve user does not bother about the underlying structure or how things and processes are being implemented. Though a naïve user may not feel much difference, there is a massive difference in how a system with lower quality deals and comes out

Manuscript received November 5, 2019
Manuscript revised November 20, 2019

with the final, conclusive result when compared to the system with the higher quality [5].
When the system is configured, and all the models are enforced with an intention to improve the system quality, it certainly does so. As shown in Fig. 1, as a system with higher quality will be more efficient, and the processes will be more refined. That is the only thing we want to have from an efficient system.
When we mention efficiency, we are referring to the speed of the system, the memory it consumes, and throughput. The efficiency is the only measure that a naïve user can talk about, and still, he may ignore the sub-parameters. The configuration management and the changes that we make in the system helps in increasing the quality measures like usability. They help in making the system usable, and it is an important measure of software quality.

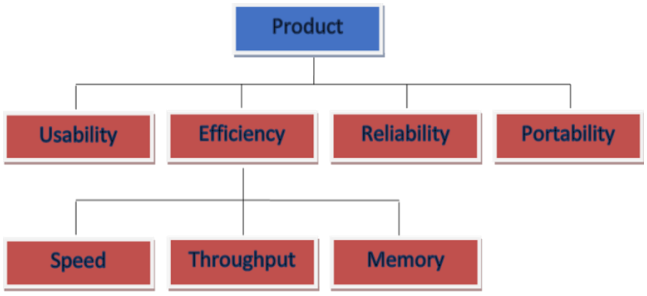


Fig. 1 The Different Measures of Software Quality.

## 3. The Change

What if we have designed a system fulfilling our requirements? Do we need to improve it? Can we make it much better? Can the improvement impact one element of the system negatively?
The system and all associated elements must synchronize in a way that the system adopts the changing without crashing. It must work under heavy load without crashing. The change must be easily implemental. The quality of the system is not the improvement of only one aspect of the measures which are necessary and ignoring any other at all. Yes, partial improvement is a choice, but the overall system has to be an improved system. It handles all complex calculations and works under all situations. The same goes for the organization. The organization keeps on improving in terms of the processes involved. The improvement of the processes within the organization results in an improved organization, which is said to be a higher-level organization in terms of quality. The quality is not something that stops improving. The organizations keep on improving their processes resulting in improved overall system quality [6].
The change must be easily adoptable within the organization. There is always a process for implementation of the processes to make it more refined.
Suppose the system has been developed and further a change is required in order to work better, the change will be implemented or not? Yes, the change will be implemented, but the topic does not end here. The change will be implemented, but there is proper management of change. The discussion is pointing to one change management system.
The quality provision does not just end by just making a system providing the solution to our problem. There is a little more than that. It must do it efficiently, fulfilling all other quality measures too. There are several other measures that it works with which are related to the organization. The in-depth view of such measures will be discussed later.

## 4. Software Configuration Management (SCM)

Suppose, we are working with an efficient system which is working fine and fulfill all our needs. We, as a user, do not find any need to improve it. It has been working fine for many years without any improvement in system quality. Do we need to change it in terms of development? Do we need to improve the processes or the workflow? The answer is obviously yes.
There may be multiple reasons for a change and improvement in system quality other than making it just fast [7].
As shown in Fig. 2, SCM or Software Configuration Management is the process of change in the system for improvement of overall system quality. The difference can take place in the code as well as in the processes. The goal of SCM is to increase productivity with the implementation of systematically enforcing the entities throughout the SDLC (Software Development Life Cycle).

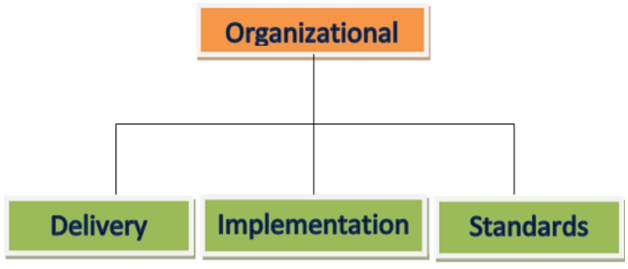


Fig. 2 The Process of SCM.

When we mention the systematic change, it means the change is not implemented right away directly on the

elements or the modules. Once we have decided to make a change in the system, can we go and make the change? The answer is simply no. We are discussing the system, not the individual entity. There are several issues and the interconnectivity of various elements involved. There must be a refined process of approval, processing, and implementation [8].

Once decided, the baseline is established along with the track to implement that change finally. Though change is made in one module or process, there are some modules that will directly be affected by a change in any other module of the system. The best example that can be given here is documentation.

While implementing the change, there are several scenarios that are taken into consideration, and those require to be looked carefully before and after the implementation (Fig. 3).


Fig. 3 The Steps of Change Management.

It is clear that there are several issues when changes are made without recording a change. Here recording means the proper documentation of the change. The documentation states that the change has taken place, which will reduce the chance of any problem later on.

SCM is an umbrella activity, and this activity is not activated or applied to a specific phase. This is not an activity-specific for one stage. It is applied to all the stages involved in the process.

SCM is the configuration management, which includes the change of process and improvement in all modules in which the documentation recording is the one. Here one point comes in mind before discussing SCM in detail. The point is the need for SCM. The fact is the several various or modules work for the one system. We have to accommodate user requirements, policy, budget, or schedule. SCM, no doubt, contributes towards the control of the cost involved. But mainly it contributes towards the improved system quality [9].

## 5. SCM Tasks & Process

Now when we have established a ground, let us mention tasks of the SCM process (Fig. 4).

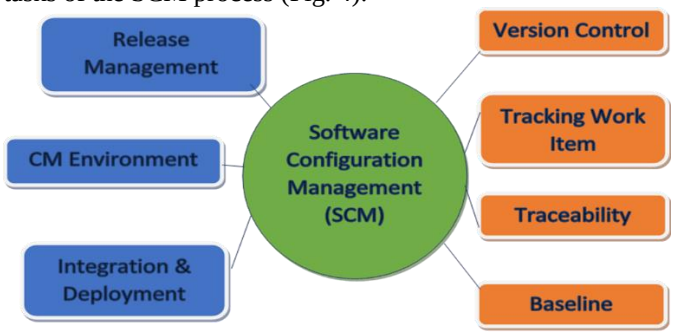


Fig. 4 The Tasks of the SCM Process.

Hence, the SCM is a crucial activity available within all the domains of the SE. And it provides and avoids confusion that may be created with the unavailability of SCM. Primary purposes of SCM are;

Configuration Identification is important to have a clear idea about the overall system we are working with the scope of the system. Only then we can make a change after once we identify it. Here we use the object-oriented approach to identify the SCM repository. Each module has its features, name, and functionalities. Here, the name has its own parameters for identification.

The baseline is to facilitate the flow of the process. Here the flow means the configuration process to improve the quality of the system. The baseline is associated with functional and development. This is the important task of SCM [10].

There are several products and parts:

- Specification – Systems
- Requirements
- Specification – Design
- Code
- Procedures & Data
- Operational System

Control of Change is an important task of SCM, which ensures quality and consistent behavior during the whole process. One change has been decided to improve the system quality, cannot be implemented as desired without the proper control because there are more than one module and entities involved during the change. The repository is where changes are committed. The ad-hoc environment for the change to take place is provided where the impact of the change is tested. This makes this task more important. The problem is the change, if any, is found and revert on the technical grounds.

In this process, configuration modules provision is made sure throughout the process, and the management of changes is ensured throughout the process.
Configuration Status is important when there must be some task of SCM, which should be related to keeping track of all the changes made during the process. It basically compares the tasks performed with the previously achieved goals that simply meant the comparison of baselines.
The complete record of all the changes since the system was in the previous state is arranged (Fig. 5).

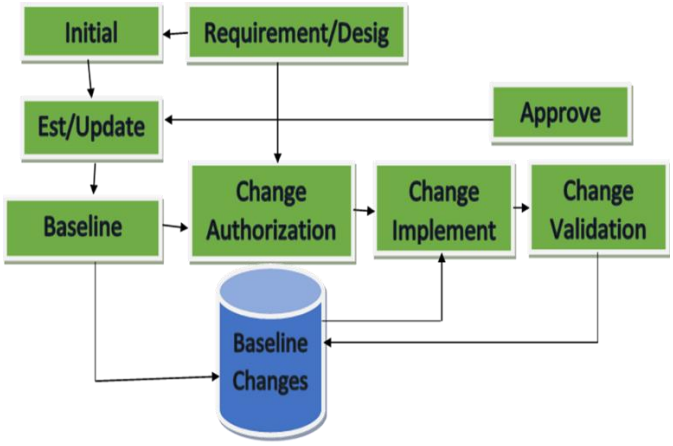


Fig. 5 The Complete Record of the Changes.

Audit & Reviews task where it is ensured that everything is going as required. It is the task where it is made sure that everything is according to the needs of the baseline. Here it is ensured that the goals are being achieved along with the traceability which is intended.
SCM is the activity which is directly related to managing the control rather than just emphasizing of already built ideas. Here it must be understood that change is something which is necessary within the system built and there may be several reasons for that. The main reason may be that we wish to increase the quality of the system by just implementing a few rules for the system. But it must be understood that the change is a must and the main problem occurs when the changes have to be managed in an efficient way. Also, we have discussed what confusion the changes may create if not implemented properly [11].
We are discussing the role of SCM towards system quality. Why do we need to improve the system quality?

- Market & Business demands
- Customer Demands
- Business Change & Growth
- Budget constraints
- Schedule constraints

There may be market requirements. We, as an organization, need to compete with the market demand and fulfill the demand of the business. If the competitor or let's say the market has started using new technologies for refined workflow and the system quality has been improved, it becomes necessary for us to do the same. Another reason may be the demands of the customer. Though the confusion may be reduced during the documentation gathering phase and providing a rapid prototyping model to the customer still, the knowledgeable user still gets new requirements of something to be changed.
Business change and growth bring the attachments with new market measures in which the technology with HW & SW may be included. Also, the budget constraints may force us the change to implement, along with several schedule constraints. No matter what the reason is, the change control and management are necessary. While discussing all the change sources, it must also be understood that not all the changes that are required are justified. But it can be said that most of those are justified. A baseline is there to enforce a change within a system, which is justified.

## 6. CMM: Another Approach

Is SCM the only approach for refining the process to improve system quality? The answer is simply no. There are several methodologies used to enhance the system quality and a lot of ways an organization can improve its own quality, impacting directly towards the overall system.
CMM or Capability Maturity Model was developed in 1987. This is a model by which the overall improvement of an organization is achieved. There must be some mechanism to identify the quality and process refinement of the organization. The system quality is achieved by an organization which keeps on improving itself by the refinement of the processes. CMM is actually a process to refine the processes of an organization, and this is quite clear that a continuous refine process leads towards better system quality. CMM is a standard within the organizations that keeps on improving its processes.
Now, questions arise on how an organization keeps on improving the processes? How and what processes are improved at which level? Is there anything to measure or identify whether the goal has been achieved or not?
Now it is necessary to mention here how CMM contributes towards the system quality. CMM is basically an organizational structure that contributes towards the improvement of itself. But undoubtedly, the improvement in the organizational structure specifically for process improvement has a direct impact on the system quality.
How does an organization work throughout the levels to improve its processes? Does it make any difference for an

organization to improve its level? What is the importance of a level of improvement for an organization? How does the organization's maturity impact software quality in a positive way? To have the answers to all these questions, we shall have to understand the levels of CMM.

## 7. CMM For Quality Improvement

Let us discuss the levels of CMM, and it will make us understand in a better way how it contributes towards system quality.

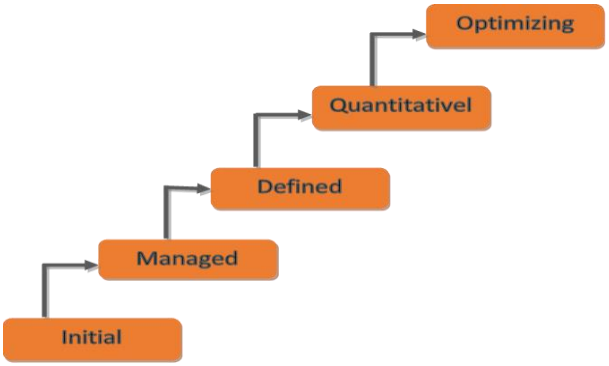


Fig. 6 CMM Levels.

### A. Initial

The organization is on the initial level. It simply means the processes are not refined. At this stage, the organization has ad-hoc level processes.

On the initial level, the organization manages things on an individual level rather than having those on a combined, broader level. This makes the work environment narrow for the whole workflow/ project. There is no concept of measuring the progress of the organization, and no track of progress or the problems is kept. If there is no track of any progress, there can never be any improvement.

There is no management of scheduling the project, along with the absence of budget management. The organization has no control over any such thing. Usually, there is no real schedule. Hence there is a huge gap in terms of decided schedule and budgeting. The organization fails to meet quality targets.

Now consider the scenario an organization is competing with its competitors. There will be no competition unless the organization improves its processes and system quality.

### B. Managed

An organization improves itself for better system quality and reaches a higher level. The requirements are quite managed at this level. There are estimates which are quite close to reality. Also, the organization works hard towards the schedule set. There are proper standards of software which are defined. The organization improves the organizational structure with the proper documentation.

Processes at this stage are planned in a better way. There may be not too much control over cost and budgeting, but still, the organization tries to manage those as well. At this stage, the organization may have a lot of paperwork going on. But this is a better stage as compared to the previous level, where everything was on a basic level, and processes were not managed.

### C. Defined

Later the organization makes the processes more refine and contribute towards a better system quality. At this stage, the organization clearly understands and documents customer requirements. At this stage, the organization tries to make sure that deadlines are met, and everything goes right as the user requirements.

Organizations at this stage try to remove several problems it faced before at early levels. The processes are more managed, and the tasks are well-documented. The organization takes the process and system quality very seriously and works on it as well. There is not only coding going on in the coding phase. The focus is on the proper structure, which is properly managed. The whole organization makes the combine effort to achieve the goal of system quality.

### D. Quantitatively Managed

At this stage, everything ignored earlier with respect to quality is also considered. The organization understands how to work on sub-processes and works for those sub-processes. The organization implements the practices and optimizes the best practices, which may contribute towards a better organizational structure hence resulting in improved system quality. At this stage, the process improvement is quite understandable for the organization, and it clearly knows how and what is making the difference within the organization for better system quality.

### E. Optimizing

For an organization to be on this level simply means that the organization has a better understanding of the analysis of the problem. At this stage, it becomes easy for an organization to resolve such problems. There is continuous improvement within the organization, and the processes at this stage are at the highest level of refinement.

At this stage, the organization is capable of detecting errors at an early stage, which always saves time and cost. All those are resolved at the early stage before they are embedded in the system.

Now it seems easy for an organization to reach on a higher level. But it is not possible for an organization to make it happen overnight. When an organization works for its improvement, then it may take months or years for an organization to be on a higher level.
It may take a continuous hard work of a year or more to upgrade a level.

## 8. Conclusion

We have discussed SCM and CMM in this paper who contribute towards a better system quality. There is no doubt that there are several measures that an organization needs to take before they can deliver a high-quality system. Also, once identified, the change in the process cannot take place without proper management. It has also been discussed on how configuration management improves the system quality. It has become evident that change is important if we want to improve system quality. Also, it has been discussed how the improvement within the organization structure contributes to system quality.
Both SCM and CMM play an important role in improving system quality. Can these two works together? Yes, SCM and CMM work together. An organization has to work on both umbrella activities, which finally result in an improved system quality as an end product.

## References

[1] Bersoff, E. H., V. D. Henderson, and S. G. Siegel “Software Configuration Management: A Tutorial.” (1979)
[2] Babich, W. “Software Configuration Management” , Mass.: Addison-Wesley, (1986)
[3] Berlack, H. Ronald. “Software Configuration Management”. John Wiley & Sons, Inc., (1992)
[4] Pressman, Roger S. “Software Configuration Management”. In: Software Engineering A Practitioner's Approach, McGraw-Hill Companies, (1997)
[5] C. Walrad, D. Strom. “The Importance of Branching Models in SCM. Computing Practice” (2002)
[6] P. M. Duvall, S. Matyas, and A. Glover. “Continuous Integration. Improving Software Quality and Reducing Risk.” Addison-Wesley, (2007)
[7] Essmann, H., Preez, N. du: An innovation Capability Maturity Model- Development and initial application (2010)
[8] TSE, K.M, Innovation Capability Maturity Model- Apple computer case study (2012)
[9] Barkley, B.T integration Project Management New York (2006)
[10] Jakobson C.R., & Southerland J. Scrum and CMMI- Going good to great (2009)
[11] Paulk, M.C. Extreme Programming from CMM perspective (2001).