Maya Hussein

CSCE-3701-01

September 2022

Optimization Methodologies

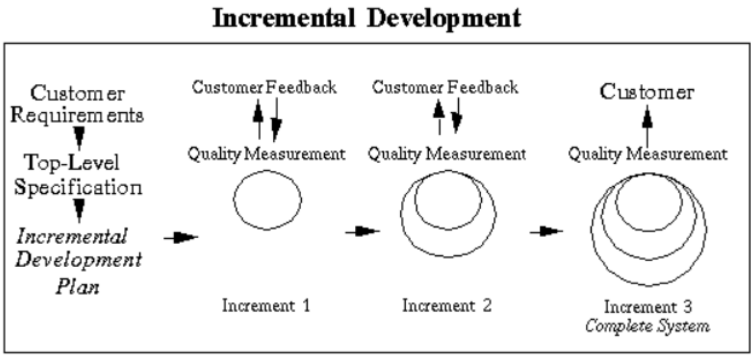
1 Introduction

Developing a software program is a process that requires various organizational techniques and depends on multiple external factors. From gathering requirements to quality assurance and deployment, these software activities can become tedious and completely chaotic if not applied profoundly and somewhat systematically. Software engineers, as a result, have been adopting new methodologies that aid in creating fully functional software that eases the stages of software development. Additional methodologies, on the other hand, have been known to improve such life cycle techniques, enhancing the cycle's overall efficiency and time capacity. Some of these techniques include but are not limited to the Cleanroom Methodology, the Capability Maturity Model, and the Six Sigma Method.

2 The Cleanroom Methodology

The Cleanroom Methodology spans the entire life cycle of agile-based reliable systems. The strategy of that technique is to avoid program defects in all rather than reducing the marginal error. The basis of this approach is to produce a series of increments, each representing a user function that is built top-down into the final product. The specifications of each increment are based on box structures. The structures consist of three views: black, which defines the user views in terms of inputs, output, and response to stimuli; state, which provides a view of internal data to the black box; and clear, which applies functions to the state internal data to satisfy the behavior of the black box. Next, each transition is verified to satisfy the requirement system. [2]

In general, there are five strategic activities in which the cleanroom methodology approaches development. Initially, A state box is implemented to show the system's response to stimuli in a stage named *formal specification*. The code is then *incrementally developed* by refinement of clear boxes, where each increment is a complete iteration of the *structured program*.[2] Below is an overview of the incremental development process.



[?]

If all iterations in the process are controlled, *static verification* and quality assurance begin by applying rigorous code inspections. It is good to note that there is no testing method for the coding section. Finally, mathematical tests are applied under *statistical testing of the system* in parallel with system specifications [2].

The cleanroom approach has been successful and cost-efficient with many applications as the process does not support going back and fixing every defect on the way, so cleanroom engineering teams are not obliged to insert new bugs when fixing old ones. For instance, after the cleanroom implementation, the IBM COBOL/SF restructuring tool has reported a ten-fold defect reduction in eight-five thousand lines of code during testing. After deployment, only seven errors were registered in the first three years of service [2]. However, even though the cleanroom approach has been recorded to reduce costs and defects while producing high-quality products, it is considered time-consuming and overwhelming as the quality assurance phase is embedded with several stages of inspections and verifications.

3 Capability Maturity Model Integration (CMMI)

Another optimization development process is the Capability Maturity Model Integration (CMMI). CMM is a method that is used to rate the level of the software development process on a scale of one to five. The CMM aims to help reduce the maturity levels of a program by setting a list of practices and recommending a general structure for improvement[3]. However, the CMM technique faced issues of contradiction, overlap, and lack of standardization, all of which were later solved by the evolved version, CMMI. There are two types of Capability Maturity Model Integration: CMMI Staged Representation and CMMI Continuous Representation. Staged Representation is ultimately used in software, where it uses an improvement path known as the Maturity Level, while Continuous Representation is mainly utilized in the Systems Engineering Capability Model with four categories known as Process Areas [1].

In Staged Representation, the maturity levels are measured by the achievement of specific tasks or goals that apply to each process area. At Maturity Level 1 *Initial*, the process is normally chaotic in an unstable environment. It is usually software that works; however, they exceed budgets and schedules. At Maturity Level 2 *Managed*, the software has achieved all specific and general goals with complete, planned, and unambiguous requirements. The status of the work is usually visible at defined points. Next comes Maturity Level 3 *Defined*; it addresses all the attributes of Maturity level 2 in addition to having a set of standard processes to fit the criteria of a particular project. Operations are generally managed with a profound understanding of the process activities and the measures of that process. *Quantitatively Managed* Maturity Level 4 occurs when all the specific goals of the processed areas of Maturity Levels 2, 3, and 3 have been achieved. Quantitative objectives, typically functional requirements, are set and utilized in process management, where they are evaluated in statistical terms and interpreted into the software's life cycle. Lastly, Maturity Level 5 *Optimization process* is focused on addressing causes of variation and altering the process to enhance the overall process performance while maintaining all the previously achieved goals in the other Maturity Levels [1].

The CMMI was implemented by a software company *Espírito Santo Informática*; the institution aimed to "venture" into dynamic organizational alterations to increase customer satisfaction by filling the gaps in their software development life cycle. After applying the CMMI technique, the organization was able to see the improvement of the project at Maturity Level 4. As for their future research, the company aims to reach Maturity Level 5 utilizing the CMMI [4]. Hence, with every stage of the CMMI, it is noticed that it ensures producing better quality of the project and provides better insight into the future of the project. However, the CMMI may not be suitable for every organizational project as it is considered expensive to reach prominent levels of maturity. It also may not be suitable for all processes as it does not always ensure security [3].

4 Six Sigma Methodology

A known optimization technique is Six Sigma; it is recognized for using statistics and data analytics to reduce defects while improving cycle times. Like the CMMI, this methodology increases performance and decreases process variation, all while improving employee morale. It is based on five key principles – focusing on the customer, measuring the value stream, discovering the problem, eliminating variation, and creating a collaborative team with a responsive ecosystem. It has two main methodologies, the DMAIC, which is defined, measured, analyzed, improved, and controlled, and DMADV, which refers to define, measure, analyze, design, and validate. DMAIC is applicable to the production of certain products or services, while the DMADV is adopted with existing products that do not meet customer/user requirements [5]. Since the stages are similar with minor variations, they both work with similar fashions. The Six Sigma process begins with *Define* or, more specifically, customer targeting, identifying the client's issue, and setting goals. The second phase *measure* concentrates on the metrics and tools used in the project. The *analysis* comes in the third stage of the methodology, where influential variables are determined, and variations are evaluated using historical data. Process Improvement then evaluates how variables can impact one another and how they can be enhanced. Finally, in the last phase, *control* ensures that the performance objective from phase four is implemented and that the improvements are sustainable [5].

Six Sigma has proven to reduce costs as it provides recommendations before any financial or technical loss occurs, directly impacting profitability. For instance, an automobile company in Taiwan is in the product maturity phase in a saturated market. In return, to improve customer satisfaction, they followed the DMAIC Model of Six Sigma. Results have shown that the company's revenues have increased by about 25% [5]. However, it is also known to produce substantial amounts of empirical data causing it to become time-consuming. Moreover, it being a quality improvement process with specific protocols sometimes leads to an increase in the overall costs. The cost of Six Sigma can be pretty high for small businesses; even large companies need to train an entire team of employees in order to begin the procedure.

All methodologies have proven to show great outcomes in terms of profitability and quality satisfaction. Though Six Sigma and the CMMI are more costly than the cleanroom methodology, the cleanroom methodology is time-consuming. Every model has its benefits and losses and is said to be most effective under specific systems and projects. All in all, these methodologies are additives to the software development process to ease implementation and encourage collaboration.

References

[1] CMMI Product Team, *CMMI for Systems Engineering/Software Engineering, Version 1.1, Staged Representation (CMMI-SE/SW, V1.1, Staged)* Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2002-TR-002,2001.http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=6041

[2] Jorgen Veisdal. *The Nature of Infinity - and Beyond*. Medium, 2018. Pattie G. Dickerson. *The Cleanroom Software Design Process*. The University of Missouri–Kansas City, 2009

[3] Kathryn EE. *Background to Capability Maturity Model Integration (CMMI)*. Microsoft Ignite, 2022.

[4] Liberato, Marco Varajão, João Martins, Paulo. *Implementation and Results: The Case of a Software Company*. 48-63. 10.4018/978-1-4666-7473-8.ch003, 2015.

[5] S. C. Chen,K. S. Chen T. C. Hsia. *Promoting Customer Satisfactions by Applying Six Sigma: An Example from the Automobile Industry*.21-33. 2018.