International Journal of Engineering & Technology, 7 (1.1) (2018) 290-293

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Analysing software quality using cmmi-2 with agile-scrum framework

B Chaitanya Krishna [1] *, A Yeshwanth Srinath [2], N Bhavani [2], G Jaya Sai [2]

[1] *Assistant Professor, Koneru Lakshmaiah Educational Foundation*
[2] *Student, Koneru Lakshmaiah Educational Foundation*
**Corresponding author E-mail: chaitu2502@gmail.com*

Abstract

Capability Maturity Model Integration is the best proposed theory for software quality assurance. Implementing the agile methods linking to Capability Maturity Model Integration and presenting the software. Basically Scrum is one of the best implemented methods. We will link the scrum to CMMI level managed to the scrum methods and define software quality.This Application totally involves with Consumer or customer behaviour while shopping an Item. Seller Companies analyzes this behaviour in order to rate their product or increase the sales of the product. Every Product selling company need to know about market analysis of their product. This project is proposed to provide market analysis of the each product to the seller companies with interaction of the customer purchases. Now days because of increase in media and internet, shopping items through online is increased. Seller companies need to rank their products and they need to administrate products in their companies , that is few products they want to retain, few they have delete/stop but in order to stop or promote a product they need to know about product status in the market. With survey sites sometimes it may fake results. To overcome such kind of drawbacks, this project is proposed, this project directly estimates the behaviour of customer, in search, and buy a product. The result given this project is useful for company to estimate the status of the project in the market.

Keywords: *Capability Maturity Model Integration; Agile; Scrum; Software Quality; Scrum Master.*

1. Introduction

1.1. Software quality

The software quality, in the context of the software engineering can be defined as how well does the software performs the role of satisfying the user needs and meets the requirements based on which it is created .It is the degree to which a system meets specified requirements and customers expectations. It is also monitoring the processes and products throughout the Software Development Life Cycle. The capability maturity model integration is the modern quality standard that is employed to describe the quality of the software. This model preceeds the later software quality models such as the GE model and the Bohem model.[4]

1.2. GE mode

This model took its origin in early 1977 which was developed by the scientist Jim Mc call with an intension to bridge the gap between the users and the developers. The Jim Mc cal in this perspective tried to match the user view with the developers priority The GE model or also called as the Mc call model which was initially used as the model to describe the software quality. The basic view of the model regarding the software quality will be classified three major operations:-

1) Product Revision
2) Product Transition
3) Product Operations

1.2.1. Product revision

The Product revision functions the parameters which can influence the change in the software

1) Maintainability, ability to identify defect and fix.
2) Flexibility, ability to make changes as dictated
3) Testability, feature of validating software requirements

1.2.2. Product transition

This perspective of product transition identifies the factors influencing the adaptability conditions of the software Portability, transferring ability of software from one environment to other Reusability, using existing software components in different context Interoperability, the extent where software components work with ease

1.2.3. Product operations

This perspective the product operation identifies the influencing factors of quality to which software fulfils its specification

- Correctness, matching between specification and Functionality.
- Reliability, extent to which system fails.
- Efficiency, system resource usage.
- Integrity, protection from unauthorized.
- Usability, ease of use.

Copyright © 2018 B Chaitanya Krishna et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

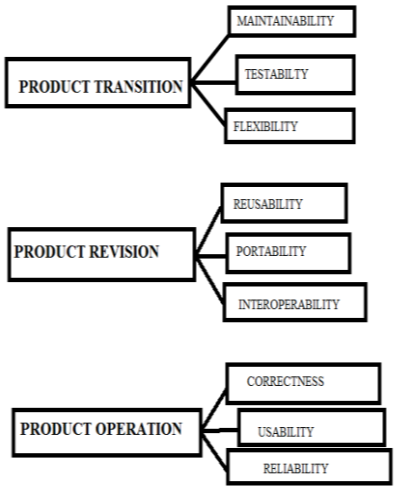

Fig. 1: A Flow Chart on Product Operations.

2. Boehm model

This model integrates risk management and incremental development. This model is also known as Boehm spiral model as its activities involve backtracking. Each loop in the spiral consists of a software process. The inner most loop consists of system feasibility, then the next loop consists of requirements, then the next loop contains design so on. Each loop consists of four sections namely,

1) Objective setting: Project risks are identified and alternative strategies are planned.
2) Risk Assessment and reduction: For the identified risk, analysis is conducted.
3) Development and validation: Development model is chosen. The best development model is throw-away prototyping.
4) Planning: After reviewing the project a decision is made whether to continue or not in the further loop.

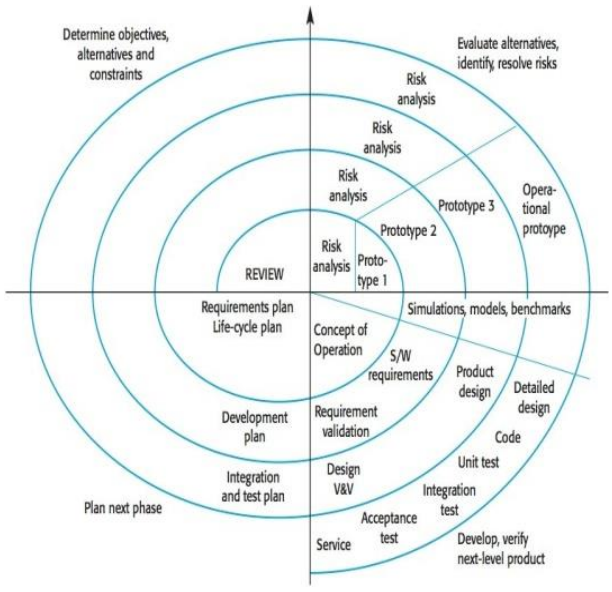

Fig. 2: Bohem Model Process Description.

3. Dromey model

It is model that measures the software quality through the measurement of tangible quality properties. It is a model that identifies quality evaluation differs for each product and holds a more dynamic approach for quality analysis. Dromey model classifies the properties into four categories:

1) Correctness.
2) Internal.
3) Contextual.
4) Descriptive.

This quality model was represented by R.GeoffDromey .It focuses on the relationship between the quality attributes and sub-attributes, and connects software product properties with software quality attributes.

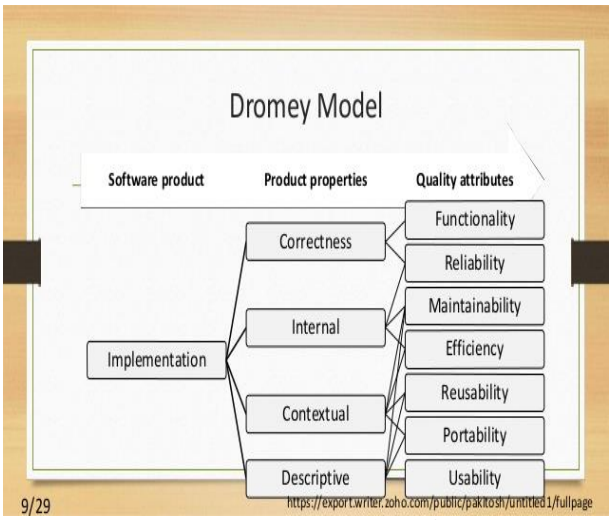

Fig. 3: A Flow Chart on the Implementation of Dromey Model.

4. Capability maturity model integration [CMMI]

[1][2] The capability maturity model integration is a quality standard to certify the quality of the software delivered. The capability maturity model provides a clear understanding to an organisation that what it should do to promote behaviours that provides improved performance in the final software product. A maturity level is defined perfect towards achieving a matured software process. Each level in the model provides a layer in the framework for continuous process improvement. The latest developed model for software quality assurance is capability maturity model integration and it is being used to manage system complexity and performing verification and validation of requirements, by using this model we evaluate the requirements model by following the five maturity levels and ensure the quality of the software process. The CMMI helps us to solve the following software questions when we get struck in the middle of the process:

- How will we know about what we are good at?
- How can we ensure weather we are improving?
- Weather the Requirements change process is useful?
- In what way can we say that our end product will satisfy the end user?
- What will say that our products are good at maximum extent?

The capability maturity model integration can also be termed as the comprised collection of the Process areas. The CMMI is not and never a process, it's a guide to tell 'What's' but not 'how's' related to the software development

According to the thesis of the software engineering institute (SEI-2008) Capability maturity model integration helps to integrate traditional organisational goals, set of the operations, priorities and provide guidance for the software quality process and provide the point of reference appraising the process. The capability maturity model exists in two major representations they are:

- Staged
- Continuous

The design of the continuous representation of the CMMI allows the user to concentrate on the specific set of the process which is tagged as important to complete the immediate business objectives. Whereas the design of the staged CMMI representation provides the standard and stable sequence of improvements and can be

employed in comparing maturity of the different organisation the staged representation can also be tagged as platform independent. [10].

There are mainly five levels which are to be called as the maturity levels of the capability maturity model integration:

Level 1- Initial

At this particular maturity level the process are usually adhoc which means unpredictable. And also there will not be any provision of the stable environment. The maturity level Initial is characterised by the tendency of over committing and abandoning the process at the time of crisis.

Level 2-Managed

At the maturity level Managed of CMMI; an organisation can said that it had achieved the specific and generic goals of Managed process areas. At this maturity level the requirements, processes, work products, are managed in accordance to their document plans. [5].

Level 3-Defined

At the maturity level Managed of CMMI, an organisation can said that it had achieved the specific and generic goals of process areas of both the maturity levels Managed and Defined.

Level 4-Qualitatively managed

At the maturity level Managed of CMMI; an organisation can say that it had achieved the specific and generic goals of process areas of both the maturity levels Managed and Defined and also the qualitatively managed. At this maturity level there will be a selection of sub process that will contribute to the total processes performance.

Level 5-Optimizing

At the maturity level Managed of CMMI, an organisation can said that it had achieved the specific and generic goals of process areas of both the maturity levels Managed and Defined and also the qualitatively managed and also the optimizing which are nothing but the above fellow processes. The maturity level optimizing focuses on the continous improving performances by the both innovative and incremental technological improvements.

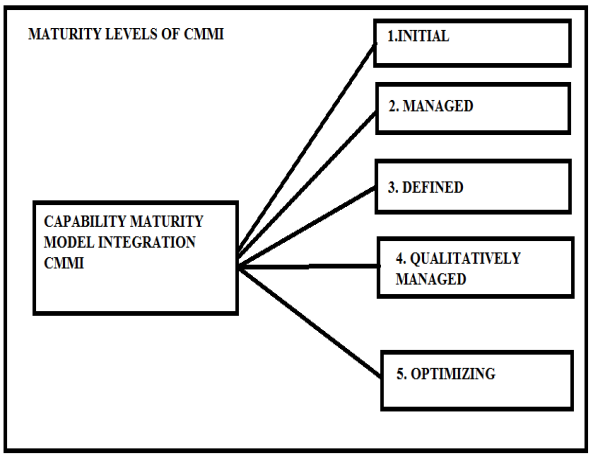

Fig. 4: A Flow Chart Depicting the Stages in CMMI.

5. Agile software development

Agile methodology in the software development can be explained as the group of the software development methodologies that are mainly based on Incremental and Iterative approach. Agile model divides the work product into further incremental builds which are further provided into iterations each iteration typically lasts from about one to three weeks Where the self-organizing and collaboration teams play the role to give out the solutions and requirements. The term AGILE MANIFESTO was first introduced in the year 2001. The functionality of the agile software development is that it mainly focuses on the process adaptability and customer satisfaction by rapid delivery of the working software product. Of course the each and every stage of the iteration the agile model involves the functional teams working on the areas like planning, requirement analysis, design, coding, unit testing , acceptance testing. [8] [2] [9].

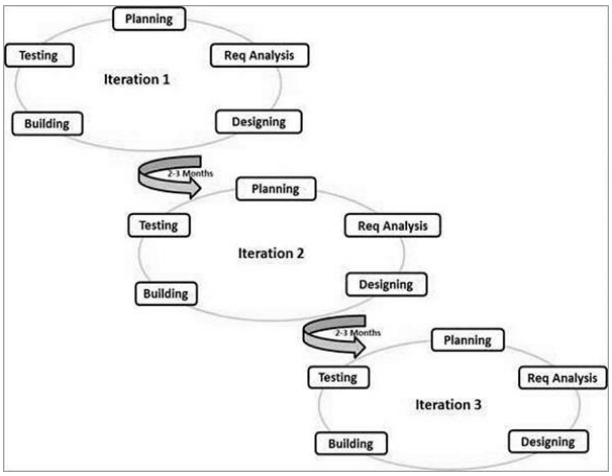

Fig. 5: A Diagram Depicting the Iteration Process in Agile.

6. Scrum

In this project we are particularly using the scrum with agile framework. The scrum is an incremental and iterative framework for the achieving the software product development. In the process of the scrum, mainly contains the product owner who creates the prioritized list of the tasks which are to be executed which is called as the product backlog. Sprint the common amount of the time period which will be given to the team members which will be generally from (2 to 4 weeks) pulls out the task from the top of the prioritized tasks, which will be named as a sprint backlog and decides how that tasks are to be implemented. There will regular inspection, that is, the day to day inspection to access the tasks progress which will be called as the daily scrum. The scrum master who will supervise the team members and keeps them focused on to the goal. At the end of the sprint the work should be potentially shippable, which means ready to be handed over to a customer. The final stage of the sprint will be the sprint review at which the total tasks and processes performed at that particular sprint will be reviewed. When the first sprint is completed another set of tasks will be performed selected from the product backlog. No true and secure system to estimate the status of the product in the market.100% cannot depend on online ranking Surveys may not give full details Conducting product surveys require man power Seller company administrators are suffering and spending much time in taking the decision to promote/ stop a product. This project is proposed to provide market analysis of the each product .it is necessary for analyzing product (i.e.) the particular product is getting profit or loss. When user is getting loss he/she to make decision to provide offers (or) to decrease actual price. This situation is prevented by using agile method. [3]

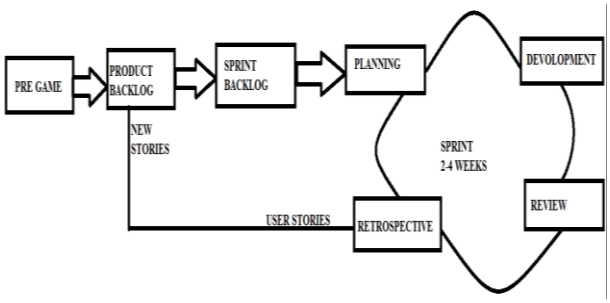

Fig. 6: A Flow Chart Depicting the Scrum Framework Process.

Role	Responsibilities
PRODUCT OWNER	The product owner in scrum represents the interest of every one with a stake in the project and also maintains the product backlog. [7]
TEAM	The team is responsible for developing the functionality. Generally the teams are self managing self organized and they are responsible for the task of turning the product backlog into increment of the functionality within iteration. The team members collectively are responsible for the success of each iteration and the project as a whole
SCRUM MASTER	The whole scrum process will be guided under the survivalence of the scrum master and also equally responsible for teaching scrum to the members of the project and implementing the scrum in a project so that it fits within an organisation's culture and still delivers the expected benefits.

7. Conclusion

The main aim in the project is to analyse the software quality by comparing the software with the predefined model named as the agile model. Basically the agile is a framework in that framework the implementation of the scrum methodology gives out the required conclusion regarding the software quality. In this regard, the stage in which the software quality is measured has come into light, the Dromey model which was primary model for the explanation of the software quality, and the stages in it. Then followed by the Bohem model and the Mc' call model. The CMMI which was the latest implementation for the software quality has been employed in this model which consists five major stages through which the software quality can be improved in each stage. The main aim of the CMMI model is that it reduces the complexity of the system and frees out the software quality. In this model, the online shopping software, which consists of many fields such as the customer details, product increment. Each field in the model can be compared to the methodologies of the agile scrum methods with CMMI, and the software quality can be derived. This model which demonstrates the customer behaviour can lay path by showing key features of the agile scrum methodologies and following the guidelines of the CMMI the software quality can be derived.

References

[1] G. Alleman, "Blending Agile Development Methods with CMMI", Cutter IT Journal, vol. 17, no. 6, June 2004.

[2] C.R. Jakobsen, J. Sutherland, "Scrum and CMMI Going from Good to Great", Agile Conference, IEEE, pp. 333-337, Chicago, Auguts, 2009. https://doi.org/10.1109/AGILE.2009.31.

[3] Scott Ambler. Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process. Wiley Computer Publishing, 2002.

[4] B. Boehm, "Anchoring the Software Process", IEEE Software, vol. 13, no. 4, pp. 73-82, 1996. https://doi.org/10.1109/52.526834.

[5] T. Kahkonen, P. Abrahamsson, "Achieving CMMI Level 2 with Enhanced Extreme Programming Approach", proceedings of the 5th International Conference on Product Focused Software Process Improvement, pp. 378-392, 2004. https://doi.org/10.1007/978-3-540-24659-6_27.

[6] Julian M. Bass, "Influences on Agile Practice Tailoring in Enterprise Software Development", AGILE India (AGILE INDIA) 2012, pp. 1-9, 2012. https://doi.org/10.1109/AgileIndia.2012.15.

[7] Maria Paasivaara, Casper Lassenius, "Scaling Scrum in a Large Globally Distributed Organization: A Case Study", Global Software Engineering (ICGSE) 2016 IEEE 11th International Conference on, pp. 74-83, 2016, ISSN 2329-6313.

[8] Jeffrey A. Livermore, "Factors that impact implementing an agile software development methodology", SoutheastCon 2007. Proceedings. IEEE, pp. 82-86, 2007.

[9] Hesam Chiniforooshan Esfahani, Jordi Cabot, Eric Yu, "Adopting agile methods: Can goal-oriented social modeling help?", Research Challenges in Information Science (RCIS) 2010 Fourth International Conference on, pp. 223-234, 2010, ISSN 2151-1357.

[10] Minna Pikkarainen, Annukka Mäntyniemi, "An Approach for using CMMI in Agile Software Development Assessments:Experiences from Three Case Studies", Accepted for SPICE 2006 conference that will be in Luxemburg at 4–5th, May 2006.