Agile Software: Ensuring Quality Assurance and Processes

Narinder Pal Singh[1] and Rachna Soni[2]

[1] Dept. Of Computer Science, Gobindgarh Public College, Alour, Khanna
narindersonu@gmail.com
[2] Dept. of Computer Science, D.A.V. College for Girls, Yamuna Nagar
rsoni63@gmail.com

Abstract. In the present scenario when the software systems are getting increasingly complexed. The time lines and schedule are getting tightened day by day. The processes need to be expected as adaptable rather than rigid. The process of the development also needs to be redesigned. The old concept of sequential phase must be updated with the iteration. The processes must ensure the user acceptance with the accepted level of quality of the software. The need of time is that process of software development to be reviewed. The concept of Agility can be used to provide good quality solutions for the upcoming software system. The sustainable solution will be considered now onward that will be capable of maintaining quality, acceptance of changes at any time with a minimum cost and rescheduling of each phase of development. Agile Software Development can be a winner of coming future. But on the downside for which many times the agile processes are criticized are inability to work with the CMMI environment, this paper proposes a new model for the agile software development, that includes Customer feedback and project documentation as its major element that makes the agile development more auditable, accountable and process centric.

Keywords: Agile software, Agile processes, Software Quality, Agile solution. Planning.

1 Introduction

The software in itself is a very large entity which contains a large number of processes to be carried out and a number of conditions to be satisfied before it is to be delivered as a golden release to the client. One of the very hard question to answer for the industry yet is, what will be criteria of acceptance, what will be the processes to be followed which can meet with the need of the time and ongoing project requirements. As each project has its distinct requirements and need. The question becomes even severe because the software is not an physical entity/commodity for which levels of quality can be framed exactly. In order to design a quality solution a number of proposal are made with a different theme and architecture. But one question which none of them was capable to answer is change management. All the solution somewhere takes the software requirement as a sequential/static entity, but actually this is the most volatile entity in the whole project development life cycle. This one single assumption makes whole of the development at a risk. There are available "a number of solution for the project management, but still the solutions for the change management are still lacking". The unmanaged changes start hampering whole of the project life cycle and all the bad devils of Schedule Slippage, Cost slippage, Product adherence to the Standards, Quality Assurance become the biggest myths of the project documentation. As the product and requirement fails to agree with each other. In order to deal with the problem of change management with quality assurance, agile solutions can be adapted which are yet not accepted at a large. The paper will try to highlight some of the open problems that are limitation of the practiced approaches of software development. This paper is also an effort towards how agile development can be helpful in quality assurance.

2 Literature Review

Agile manifesto is a step towards the new practice of software development, In which more emphasis is given on the working people i.e. development team, communication and Customer. Agile implementation considers software as a continuously evolving dynamic entity while accepting the changes. In agile methodology emphasis is given more on the developing working software [Mike Cohn][11] by stepping back from heavy documentations. Agile methods have introduced Customer as an integral part of Project Development Life Cycle. The project is started as small daily meetings and works in the months long sprints [Hesam Chiniforooshan][9].
During these meetings each development progress is considered with consultation of customer and planned for the next iteration. The benefit of agile methods are remarkable as feedback on the development is readily available to keep the project in direction. Agile process having many good things are still sometimes negated for valuing active working code development only and ignoring the Planning and process driven approach. The methods of communication is also direct from one member to other. No process and plan centric approach is followed by the agile. In other words each person is managing his processes at its own, and there is no otherwise control over the autonomous development teams [Minna Pikkarainen] [10]. There are some serious queries in this regards.

- The Process is going to be distributive rather then centric, what will be the Planning to control the overall project.
- Agile process assumes the self disciplined [Minna Pikkarainen] [10] independent team members as its core. which is not possible all the time.
- In Agile processes that lack central management, how team competency, team player and role selection for current assignment, communications [Hesam Chiniforooshan] [9] and reporting channel will be established. What will be flow of information?
- Management opened queries like what will the project schedule, progress reports, cost analysis for features updates done time to time for customers.

Agile processes steps back from documenting process which yields uncertainty for the project management team, because the project completed and ongoing project does not produces any matrices, creating an illusion between change, maintenance and rework. agile methodology says embrace the changes, but what will be the final cost estimations [Minna Pikkarainen] [10], schedule, final delivery dates, quality checks which is no where written or documented. Which makes a good methodology week enough to be accepted for large projects over the years. The project which will last over a couple of years, it is likely that the team member continues to join and leave the team. How such project can be looked after and managed for quality which lack the documentation and does not have any control over the autonomous teams.

A new model in this direction has been proposed which consider Documentation, Phased Customer feedback and Iterative development of software as its core components. The model and introduces documentation at its core will be able to answer all the opened queries of management, quality assurance ensuring correct project direction. It has been seen that agile concept mixed with waterfall methodology leads to better far more stable results.

3 Pros and Cons of Waterfall Model

Till the time, the most used and accepted model has remained Waterfall Model, with its pros and cons. The model presents a very rigid structure to the development community. As the model ignores and does not welcome the changes that are likely to be made to the software time to time [Victor Szalvay][5]. There are distinct phases in the model which are almost isolated from each other. Because the *"flow of information is not by communication between the phases, but it is by transition of one phase from other"*. So the feedback concept is totally missing in this case. The other horrible story is that there is nothing available in between to review, until one has crossed all the stages of development. "*There is no concept of testing or reviewing what is going on. Because there is nothing as output till the design phase is over*". The Testing and End User also comes in the role when the deliverable is prepared

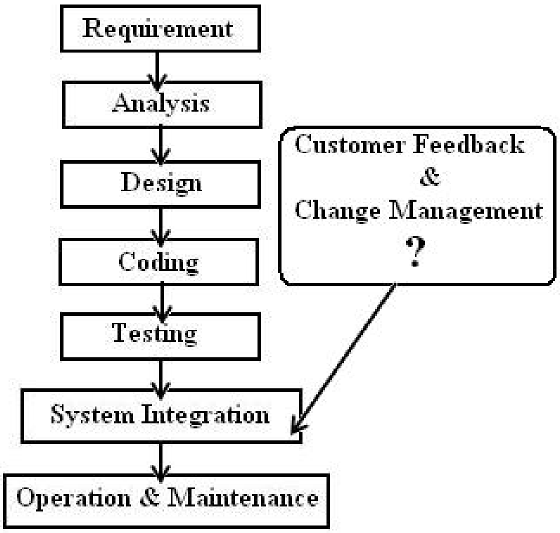

Fig. 1.

after coding. After the first release is made, it is quite practiced that changes are sidelined and this may be due to defective processes or the lack of synchronization of project documentation with the changes suggested.

There is no question to get the UAT (User Acceptance Testing) get performed and knowing about the limitations of the product or change management. The assumption followed in the model is very rigid and opposes the changes during the development. Working with this approach leads to serious drawbacks in the system. This has been proved by a number of surveys conducted all over the world.

- The Chaos Report (1995), [1] The Standish Group research survey shows a staggering 31.1% of projects will be canceled before they ever get completed. Further results indicate 52.7% of projects will cost over 189% of their original estimates.
- The OASIG Study (1995) [1] survey says, The IT project success rate quoted revolves around 20-30% based on its most optimistic interviews. Bottom line, at best, 7 out of 10 IT projects "fail" in some respect.
- The KPMG Canada Survey (1997) [1] findings are, Over 61 % of the projects that were analyzed were deemed to have failed by the respondents. More than three quarters blew their schedules by 30% or more; more than half exceeded their budgets by a substantial margin.
- The Bull Survey (1998), [1] the French computer manufacturer and systems integrator, a survey in the UK was conducted to identify the major causes of IT project failure in the finance sector. The key findings of survey was missed deadlines (75%), exceeded budget (55%) poor communications (40%) inability to meet project requirements (37%).
- The Robbins-Gioia Survey (2001) [1] - 51 % viewed their ERP implementation as unsuccessful 46 % of the participants noted that while their organization had an ERP system in place, or was implementing a system, they did not feel their organization understood how to use the system to improve the way they conduct business.
- IT Cortex Conclusion [1] The statistics presented here all converge to establish that: an IT project is more likely to be unsuccessful than successful about 1 out of 5 IT projects is likely to bring full satisfaction the larger the project the more likely the failure
- Larman, in the UK [8] shows that of 1,027 projects, only 13% did not fail, and waterfall-style scope management was the "single largest contributing factor for failure, being cited in 82% of the projects as the number one problem.
- Larman 1995, in a study of over $37 billion USD [8] worth of US Defense Department projects concluded that "46% of the systems so egregiously did not meet the real needs (although they met the specifications) that they were never successfully used, and another 20% required extensive rework" to be usable.
- Larman , In a study of 6,700 projects [8], it was found that four out of the five key factors contributing to project failure were associated with and aggravated by the waterfall model, including inability to deal with changing requirements, and problems with late integration.

- Larman, over 400 waterfall projects[8] reported that only 10% of the developed code was actually deployed, and of that, only 20% was used.

There is a long list of stories like these showing too high failure reports. *“The only factor responsible was non responsive waterfall modeling for the change management, and ignorance of the customer from beginning to end. There was no role and importance given to customer feedback given anywhere from the customer side, who has to ultimately approve the product. Which leads to such devastating results?”*

The Quality of such project is always under the question. *"In case of the Waterfall model it is no where guided or there is no such phase in which one can assure that whether the project is aligned in the right direction with the project specifications after consideration of all the changes. Another major issue with the model that Quality Assurance has not been given place during the phase transition to feedback over the work done".* It is the major drawback of the model that Quality Assurance comes into the action when the all phases of the project are over. It is then tested whether the project has gone in right direction. *"Till the first release of the project, the end user or customer is not able to express his feedback over the development that has been over. Which leads to serious loop wholes and gaps between specifications and implementation. The project developed in such a way becomes very vulnerable to requirement and specification gaps, design issues and integration issues and ultimately collapses at the UAT (User Acceptance Testing).* Agile lifecycle can be an answer to all these problems of changes management and Quality Assurance.

4 Barrier to Agile

- Existing process of software industry that does not allows the development teams to communicate with the customer directly.
- Project Mangers does not want to engage the customers because it is taken as hindrance to pace of development process.
- Development team is use to develop the software as a sequential entity.
- Customer Feedback is taken as disturbance to the development, although this is the only communication that can bridge all the gaps.
- Not Suitable for CMMI Processes.

5 Agile Solution

Software is an intangible entity without any physical attributes so it is very hard to lay down the acceptance and rejection criteria's and any scaling parameters for ensuring the quality. In order to ensure the quality of the software, there must be direction alignment if the Customer Specification and its Implementation. The Traceability matrix must be able to clear the gaps of Specifications and Implementation. *In order* *to bridge the gap of Requirement, Change Management, Actual Specifications and Implemented Specifications and to improve the quality of the software the "Customer" must be brought in the development cycle. The Process of Software Development must be redesigned to 'Release Development'.* Each release will cover a part of the overall functionality. The process of system integration must be changed to the cumulative release integration. Each new release must cover the existing functionality with the new feature and must be thoroughly tested and sent for the customer feedback. *The end of the project will always be in a high quality software already gone through the UAT(User Acceptance Testing).* We proposes the following model of Release Development in accordance to the Agile Software.

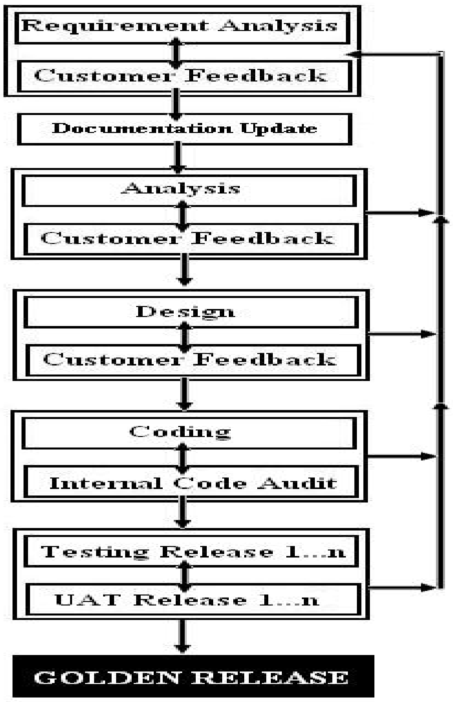

Fig. 2.

In the model proposed, all the stages have been designed ensuring customer satisfaction and quality of the software. In our terms *"Software Quality can be defined as developing a solution to perform in accordance to the customer Specifications, adaptable to changes for which it is developed and rejects the domain gracefully for which it is not developed".*

The solution proposed meet the every need of the dynamic environment of software development and ensuring the quality. There is no phase left in the model for which the customer feedback or audit is not performed. The model is based on *"Release Development"* Concept. In which each release has gone for the approval of customer satisfaction. The software developed on this approach ensures a high degree of acceptance rate. As each step has undergone through the UAT(*User Acceptance Testing).*

6 Discussion

The Agile process for which they are criticized and does not give acceptance for the CMMI processes is lack of documentation. As "Agile manifesto [2][3], "Prepare the document when it is necessary". On the other hand the CMMI is largely process driven and is based on continuous improvement and maturity model [Martin Fritzsche] [6]. But this is very hard to accept that Agile methods can not be compatible with CMMI processes. The Agile Architecture has changed one thing that is bringing the customer in the developmental model [Martin Fritzsche] [6]. *"This only one change has a major positive change on the Cost, Schedule, Requirement Analysis, Change Management as all that will be finalized will be going to prove itself in the direction of customer satisfaction only, which can be named as Quality"*

- The Agile process for which are criticized is lack of proper and validated documentations. *The proposed model has given due advantage to this to make it CMMI and Quality compatible.* There will not be a single change that will not be recorded and communicated to whole of the developmental team, which is a definite step in the direction of process improvement and synchronization. For the CMMI process the User Stories [Martin Fritzsche] [6] can be taken as base of requirement analysis. The record keeping for whole of the discussion with the customer can be done to present as proof on some later stage to resolve an issue.
- On the basis of discussion with the customer, and suggested changes and there impact, More Effective schedule and Cost Estimates can be made that is again a requirement of CMMI processes [6].
- This is the only feature of Agile Processes that customer is approving each functionality of the software as it is made available. so the Cost to implement the changes are minimum in the Agile Processes because the changes are acknowledged at the same stage in which they are to be made. This always results in a positive impact on the cost and schedule of the project, which is also one of the objective in CMMI.
- The major role of the project management is to give the right direction to the project, and avoid any kind of schedule slippages. Agile process are an

answer to this question, as the direction of the project is always aligned with customer, which goes in parallel in the customer feedback. Each and every thing of project is in consultation with customer which gives a great quality improvement, effective management.

- CMMI all objectives can be met with agile solution with a little changes in the system. The process will give there maximum in the right direction reducing the failure rate of the project almost to negligible.

7 Conclusion

Software development is not a Sequential process. So it is not possible for a Sequential Model to meet the needs of development process. It is a dynamic and repetitive Process, where each revision is equally important and impart a major impact on the overall development of the software. The documentation is the integral part of any project. This is the repository from where all the functionality is clear and visible. But the problems with which process lack is not the heavy and large documentation, but lack of synchronization ,change management and regular customer feedbacks. There are processes to update the documentation in every company, whatever they are so, but each one is lacking the process of managing the changes in a positive way and to accept them as a part of regular development. Agile process can be an answer to the solution. If we incorporate one change in the development process that is currently going, the success rate can be far better then at present. The need of change to be incorporated in ***Every Phase of Waterfall Model Must be Converted to a Release of that phase and must be sent for the customer feedback and the changes must be reviewed for their impact and seriousness on the rest of module and priority of the change must be accessed properly. The product made by following this approach will be already free from the flaws of changes, and will clear always its final test from the user side that us User Acceptance Testing.*** The developed software using this method will be maturing a prototype towards the final direction and will always be in the shape and type it was assumed to be.

References

1. The IT-Cortex website, Failure rate, `http://www.it-cortex.com/Stat_Failure_Rate.htm`
2. The Agile System Development Life Cycle website (2005-2010), `http://www.ambysoft.com/essays/agileLifecycle.html`
3. The Manifesto for Agile Software Development (2001) website, `http://agilemanifesto.org`
4. Vijayasarathy, L.R., Turk, D.: Aagile software development: a survey of early adopters. Journal of Information Technology Management XIX(2) (2008)
5. Szalvay, V.: An Introduction to Agile Software Development (November 2004), `http://www.danube.com`

6. Fritzsche, M., Keil, P.: Agile Methods and CMMI:Compatibility or Conflict? e-Informatica Software Engineering Journal 1(1) (2007)
7. The Benefits of Agile Development webiste, http://www.versionone.com/Agile101/Agile_Benefits.asp
8. The Waterfall Approach: a Critique website (November/18/2003), http://www.adaptionsoft.com/adios_waterfall.html
9. Esfahani, H.C., Et. Al.: Adopting Agile Methods: Can Goal-Oriented Social Modeling Help? In: 4th International Conference on Research Challenges in Information Science (RCIS). IEEE, France (2010)
10. Pikkarainen, M., Wang, X.: An Investigation of Agility Issues in Scrum Teams Using Agility Indicators. In: The Irish Software Engineering Research Centre (2008), http://www.lero.ie
11. Cohn, M., Ford, D.: Introducing an Agile Process to an Organization, pp. 74–78. IEEE Computer Society, Los Alamitos (2003) 0018-9162/03/$17.00