

Overpowering Challenges in Agile Software Development: Principles, Future Directions

Manas Kumar Yogi¹ Karri. Anupama Reddy² Kalaga. Sree Lekha³

¹Assistant Professor ^{2,3}Student

^{1,2,3}Department of Computer Science and Engineering

^{1,2,3}Pragati Engineering College, Surampalem, Kakinada, A.P

Abstract— this paper presents the inherent challenges faced by practitioners of agile software methodology. The challenges will force the developers to revert back to old mechanisms. So, we also propose methods to overcome their fears about agile strategy to develop a software. We have developed this paper in to categories and in each category we state the obstacle and how to overcome that obstacle. Agile software has been advocated by six sigma companies as being both effective and sufficient. Effective because it produces working, defect free software and sufficient because it meets all short term, long term requirements. Agile software methodology supports individuals and interactions over processes and tools. Documentation is minimal and customer collaboration is given more priority than contract negotiation. Agile software development embraces change rather than saying it's not in the plan. It progresses in response to user feedback. In agile environment the project adapts itself to changes as rapidly as it is possible.

Key words: Agile, Scrum, Waterfall, Obstacles

I. MANAGEMENT OBSTACLES

Senior managers in the organisation who are not familiar with agile methods won't support agile principles for software development. Also, misunderstanding in deploying extreme programming with agile methodology may decrease quality of the software. Remoteness from traditional approaches induces a feeling of insecurity while undergoing software development. Another management issue is regarding lack of suitability for their own review and assessment process. Paper planning as in traditional approaches of software development are absent in agile methodology. So we can say that managers of agile software development always face the fear of unknown.

Overcoming this challenge is educating the managers by external consultants who can deal with managerial concerns but assuming the managers are willing to co-operate. The best solution is to pick up a project of less managerial significance and implement it by applying agile principles. Finally, convince the senior management if the inherent advantages of an agile software development process.

II. FAILED PROJECT SYNDROME

It refers to the belief in the futility of the agile methodology .Few practitioners say that agile wont work for few projects. It's just waste of time and money of the organisation.re-implementation cost is unbearable in a situation where agile methodology fails. The main reasons for a failed agile project are multiple. It has been known that for such a project, none of the project members had ever done agile project before. No proper documentation was involved. They even did not design the software model and finally refactoring was completely ignored. The main reason for

failed project-syndrome is inexperience. If developers have got some prior experience hen agile projects will never fail.

III. DEVELOPER-RESISTANCE

Most of the software developers worked alone on a module and when the situation arises to work in agile teams, it is difficult for them to perform effectively . Subsequently they resist adoption of agile methodology. Also, few developers should be given freedom to work in isolation so that they solve complex problems. Such developers cannot work in tandem with a team. This results into a barrier for adoption of agile methodology. As most of developers worked alone when they were undergraduate students, they never understood the strength of collaborative work.

IV. CUSTOMER OPPOSITION

It is a equally daunting task to convince the customer about agile methodology with compromising the quality and functionality of the software. No detailed review plans for the customer are available at any stage of agile software development .At each level involvement is variable ,so a degree of uncertainty creeps in their mind. The main strategy to overcome customer-opposition is to convince the customer by an insider (or) an outsider who has trust in the customers perspective as well as the developers perspective. 50% of the job done is customer opposition to introduction of agile methodology for software development.

V. ALIGNING "VISION" WITH "ITERATIONS"

Another significant issue we would experience is that the agile team would want to self-organise and start writing the code immediately, very fast and deliver the working code .while this is not inherently bad to start quickly, but this could lead to quick frustration, and even create immediate conflicts between the people working on the project.

Solution for this problem could be first creating a vision, documenting a product plan and prioritising use cases. this might not take much time like a waterfall model, but may take some weeks of thoughtful interaction and market analysis. Also development team should think on architecture ,user experience etc.

VI. CLARIFYING THE ROLE OF PRODUCT OWNER

Selecting a wrong person or incorrectly defining the role of a product owner , would leave the agile team limping along, at the whims of a control freak bent on driving personal opinions in to the product. A product owner can be made responsible to be as the "voice of customer". Its a big challenge in selecting, defining and empowering the role of product owner in agile process.

The solution to this can be, the product owner spending a significant amount of time directly with the

development team and participating in every iteration review and approve software releases. Also the project manager can be made to participate in planning meetings, agree on priorities and then allow to decide on regular decisions.

VII. CONTRACTUAL DIFFICULTIES

some of the problems we may encounter are when the customer is unable to specify all the requirements and also has no clear idea about it which he would like to implement in his product. Also the problem may arise if the supplier does not meet all the required specifications specified by the user.

All the buyers should be pre-educated about the principles they are going to use in developing the project. Also there is need to develop a level of trust between ourselves and the clients. They must be said that they'll be allowed for daily interaction in the project and their regular feedback will be considered in order to meet the requirements of them effectively.

VIII. ADAPTABILITY OF PROJECT TO CHANGING REQUIREMENTS

Buyers must be explained that by adopting an agile approach, as the requirements change with time, these changes can be naturally and simply passed through to the project team.

IX. FAMILIARITY WITH AGILITY

Another major obstacle would be lack of knowledge of how to start and run such a project. Some of the questions that may stop us from adopting the agile principles is:

A. How do we estimate the cost of software to clients at start of project?

first the buyer must be asked how much budget there are ready to invest for the project. And the buyer must be explained all the issues involved in the project openly and honestly.

B. How to decide how much iteration would be there?

number of iterations that would be there in the project can be decided based on the length of project, size and complexity.

C. How to know what will be there in each iterations?

we should roughly estimate what features, functions, use cases will be planned for each iteration. but only the current iteration will be planned in detail first. and the based on that the remaining iterations will be planned.

D. How to decide how long an iteration should be?

we can decide this as if the problems are complex then use larger iterations and if the problems are less complex then use smaller iterations.

X. DEVELOPING A WATERSCRUMFALL PROCESS CHALLENGE

Hybrid process creates great strain on the organisation due to management team following one process and the development team with different process philosophies, terms and metrics. Agile requires the plan to be flexible

reprioritized and revised so it described as "waterscrumfall". it's really business as usual with a traditional process of defining a complete product up front and then the developing team using an integral agile process to conduct the work break down process to deliver code. But often the real testing and real development, doesn't even start until testing of the expected deliverable starts. this is way to leverage the power of ability in software development.

XI. PROPOSED SOLUTION

Product road map milestones and market releases developed in the waterfall model must be aligned completely with agile sprints and software releases. if the development team is practicing agile, they must create deliverables that track to the plan and provide early warning of what is really getting completed. To guide the development team's iterative approach, the marketing and sales team must clear on what customers deem most important and how market dynamics are impacting solution requirements to guide efforts with every agile sprint. communication of progress and product deliverables must also be spoken in both agile and business teams. Any agile approach used by the development team must support all business needs and address all stakeholder concerns.

XII. CONCLUSION

Agile software adoption is difficult basically due to lack of knowledge of how to start and run such a project. The challenges we presented in this paper are manifold and we have tried our best to discuss the obstacles faced by software developers following agile methodology. We conclude that introduction of agile software mechanism is difficult as estimation to the clients at onset of the project is difficult. Predicting number of iterations is uncertain and estimating length of each iteration is also difficult. Finally it's evidently proved that adoption of agile software will prove to be a winner in future software development methodologies.

REFERENCES

- [1] Alexander, C. The Timeless Way of Building, Oxford University Press, New York, 1979.
- [2] Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. and Angel, S. A Pattern Language, Oxford University Press, New York, 1977.
- [3] Auer, K. and Miller, R. Extreme Programming Applied: Playing to Win (The XP Series), Addison-Wesley, New York, 2002.
- [4] Bass, L., Clements, P. and Kazman, R. Software Architecture in Practice, Addison-Wesley, Reading, MA, 1998.