

Insights on Various Meeting Requisites in Handling Agile Projects: Shared Experiences from Indian Software Organizations

N. Ganesh

Department of Computer Science and Engineering,
Jaya Institute of Technology, Jayavel Nagar, 631204 Kanchipadi, India

Abstract: Many Indian software organizations nowadays are keen to switch their project development from traditional software development to agile software development. During their shift in transition many software organization stick to the practices adopted in traditional methodology and they fail in their project when they change state to the agile practice without knowing fully well about the agile software development. This study exclusively focus on the various levels of planning and its meetings starting from planning at large, like initial planning, strategic planning meeting, release planning meeting, iteration planning meeting. Most of the Indian software companies research either with extreme programming agile methodology or with scrum framework. This study focus on the scrum framework and discuss about the daily scrum meetings, sprint burn down meetings and the retrospective meetings. This study will be highly useful for the first time agile practitioners as well to the larger crowd as this paper is devised after making a thorough scrutiny and discussion from large number of experienced practitioners working on agile software development projects.

Key words: Agile software development, meeting prerequisites, planning at large, strategic meeting, release planning meeting, iteration planning meeting, daily scrum meeting, retrospective meeting

INTRODUCTION

Traditional software projects will have their own hierarchy as team lead, project lead, project manager and so on. Project manager of traditional projects will assign task will set deadlines and will monitor the work on a daily basis. Agile team members have the liberty at their work and at the same time they know their commitment and their responsibility. Members who are willing to transition from traditional to agile development will feel the difference in their work. They feel independent and they are at liberty to identify task and to execute it.

Agile practices makes team members self motivate, self assign the task and to self organize, thereby getting a feel of self empowerment (Whitworth and Biddle, 2007). Self empowered teams become sole owner of fulfilling the task assigned by them rather than getting assigned by manager as carried out in traditional development approach. Team members pick their task from story boards or scrum boards and individual team members evaluate their stand through daily stand up meetings and team members evaluate themselves through retrospective meetings (Abrahamsson and Warsta, 2003). Project progress is monitored with the help of information radiators such as burn down charts and story boards and with meetings such as Iteration planning meeting, review planning meeting and release planning meeting (Ancona and Caldwell, 1988). To get the stories as tasks from the

story boards, a strategic meeting and planning at large by the product owner and other stakeholders are carried out (Boehm and Turner, 2003).

Product owner or the customer representative provides requirements in the form of user stories. User stories are divided into small tasks by team members during the planning meetings. Based on small tasks, the number of iterations is estimated. Scrum master allows the team to work on estimation. Story points are decided based on collective decision making. Story points are the number of burn hours required to complete the specified task. During the initial days after transition from traditional development approach, the team may not be comfortable in estimating and planning and situation changes as experience level in working with agile projects pile up.

Pressure during iteration period is a common factor for the members of the team handling agile projects. With iteration pressure, it would be very difficult for team members to make a systematic end to end learning on a subject matter. So, the team members will learn only the needed concepts and procedures relevant to the project context.

MATERIALS AND METHODS

The starting point for the current research is to obtain a realistic picture of the agile methodologies as currently

practiced by various practitioners in India. The initial information was based on the observation of collective functioning of agile groups, direct face to face interviews, communication through mail, telephone and social networking sites. Armed with the terminology and established traditions of agile developers as found in published literature, the practitioners of agile methods were approached for getting responses.

It was found that most of the agile practitioners in India were practicing either Scrum or XP or a combination of the two. Scrum focuses on the project management approach (Abrahamsson and Warsta, 2003; Dyba and Dingsoyr, 2008). The fundamental agile practices such as iterative and incremental development (with varying iteration lengths), estimation and planning of user stories and tasks, status report meetings (such as daily standup), frequent release of working software and retrospective meetings to make a review of the earlier iterations were followed by agile practitioners (Thangasamy, 2012a, b). It was also found that several of the practitioners collaborate regularly with their customers. Improving the interaction with the customer and the team is discussed in (Ancona and Caldwell, 1988). A few practitioners felt the heat in not getting clarity in their requirements. This is due to lack of customer involvement as most of their customers were remotely located or customer may speak in a different language that the agile team may not understand. Some studies have described such individuals supporting customers by translating technical language to business language (Mann and Maurer, 2005). Involvement of knowledgeable customers are said to be collaborative, responsible, authorized and committed are discussed in (Boehm and Turner, 2003). More about gathering requirements (Taylor *et al.*, 2006) and issues arise in eliciting requirements due to lack of customer involvement is discussed by Ganesh and Thangasamy (2011). Agile software development is successful only when there is a proper support from the top management and a good rapport with the customer (Moe *et al.*, 2010).

Interaction with experienced agile developers enabled consolidation of many new ideas relevant to the conduct of various meeting which lead to the successful agile development (Ganesh and Thangasamy, 2011). All these ideas were combined together and common conclusions arrived at were taken as recommendations to practitioners who are keen and new in working with agile project development for the first time.

A developer has shared his experience as “That was the first meeting I had after transitioning from traditional development to the agile development. The team members gave me few cards and I did not know what to do with the

cards. Then the coach taught me how to work with estimation and now after a few months I could size something without the cards and knew exactly what may be the size of a story without even thinking about the card. It was so natural that I got used to it” (Agile developer, Chennai, India).

Meetings: For any kind of meeting the tools have to be made available based on the need of the meeting. Some of the tools are whiteboard markers, flip charts, mike, stop watch and so on. Along with this, a meeting agenda has to be circulated to the potential participants who attend the meeting. A formal signature on the attendance of the attendees would add value. Meetings on planning can be classified into Strategy planning meeting, Iteration planning, Iteration Review and Release planning meeting.

Planning at large: Planning at large implies that planning that is carried out from scratch, at the iteration level of planning till the release level. Continuous planning is essential if it is to be planned at release level and at iteration level. Planning activity has to be carried out based on priority. Priority has to be assigned by the scrum master for each story or the feature placed on the story boards. Story boards and sticky notes are prevalent when working with scrum framework (Ganesh and Thangasamy, 2010a, b). Members take up a story, write them on sticky notes and fix them onto story board or scrum board.

Story board will be classified in to “To Do”, “Doing” and “Done”. “To Do” category tasks are yet to be done (Ganesh and Thangasamy, 2011). Tasks will be written in small white paper and will be stuck onto scrum board. Team member can pick up task that he/she desires. Business priority assigned to the task is marked on the note and the member is expected to pick the task based on the business priority and not fully based on his / her own interest. An experience shared by one of the developer is as, “The intention is to deliver business value as soon as possible that you take items that are most vital from point of view of business” (Agile developer, Hyderabad, India).

Team member has to write his name in the sticky note and have to place in the “Doing” category. When task is completed by a particular team member it has to be moved to the “Done” category. This makes a team member to self evaluate his strength and to self assign his task and gets self motivated by working out the task that member desires. Figure 1 show the sticky notes fixed on the story board as found in the office of an Indian Software Organization.



Fig. 1: Sticky notes on a scrum/story board

The stage “Done” can be achieved only when “To Do” items is completed in all respect and if it is checked in to the source control and has to get a sign off from the client thereby remaining hours of task becomes zero. Prioritized features have to be split into small January 6, 2017 technical tasks. Small technical tasks can be easily completed rather taking up bigger features at a stretch. At iteration, the features are picked up based on priority either from the product backlog or from the parking lot. As many smaller segments a feature is segregated so easy will be in completing the task.

Strategy planning meeting: Top management officials such as Directors, Chief Executive Officer, Chief Technology officer and the like, customer and/or his representatives and the agile coach are mandatory for this strategy planning meeting (Ganesh and Thangasamy, 2011). It is mandatory to conduct such meetings before start of any project for a minimum duration of two hours. Customer is involved in all the stages of product development starting from identifying task till attaining continuous feedback (Grisham and Perry, 2005; Korkala *et al.*, 2006). During this meeting, vision of the project, mission of the project, goals and objectives of the projects have to be discussed (Nerur *et al.*, 2005). Minutes of Meeting, risk factors and the milestone date have to be recorded.

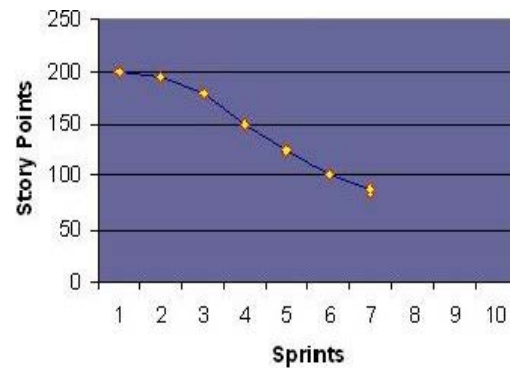


Fig. 2: Release burn down chart

Release planning meeting: Stakeholders like product owner, agile coach and proposed delivery team members should be available for this meeting. As release planning meeting is one of the most vital meeting, it has to be conducted for the whole day or depending upon the size of the project (Ganesh and Thangasamy, 2011). The essence of release planning meeting is to know the project progress. The project progress is monitored based on the burn down charts. The Burn down charts are classified into two groups, viz., the release burn down chart and the sprint burn down chart. Release burn down chart is a graph which helps visualize total points remaining for release at the end of each sprint. A typical release burn down chart is shown in Fig. 2.

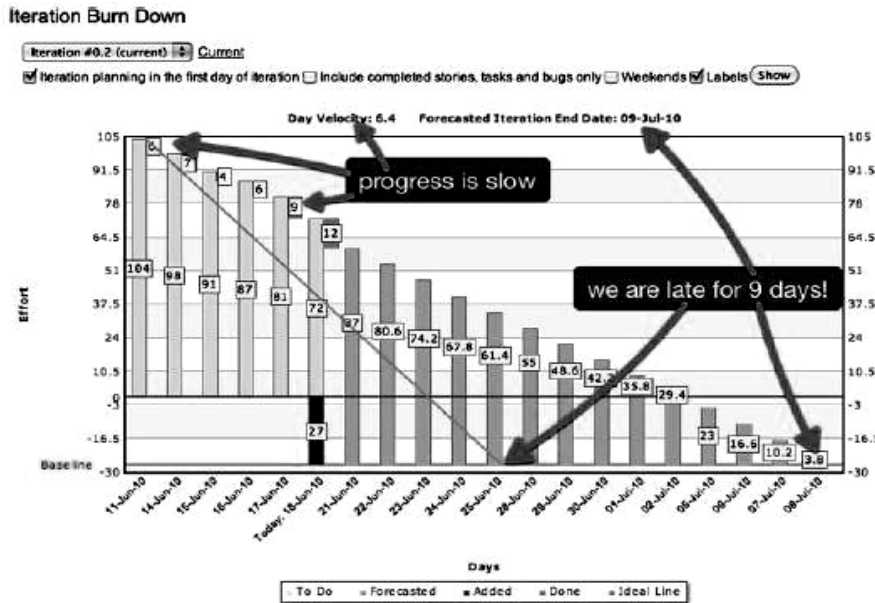


Fig. 3: Sprint burn down chart

Sprint burn down chart shows estimated number of hours required to complete sprint tasks. A sprint holds the number of iterations and minimum marketable feature delivery shown to the customer at the end of every sprint. Usually, each sprint holds two iterations. Sprint duration varies from a fortnight to a month. With the help of sprint burn down chart, customer can have a visual representation of functionality delivered and the functionality that are yet to be delivered and overall team productivity. Feedback is collected after delivering the working version of the software or the minimum marketable feature at regular intervals. Figure 3 illustrates the sprint burn down chart. This was pointed out by one of the developer as “Sprint burn down makes the customer at ease. They can know the tasks that are to be delivered to them. They can have a rough idea of how things are progressing” (Agile developer, Bengaluru, India).

A facilitator should convene the meeting. A facilitator has to be preferably a common person who is not part of the project. Apart from convening the meeting, the role of facilitator is to prepare agenda, jolt down the happenings of the decisions, risk factors and other issues of importance. In this meeting, the product owner has to play a key role in prioritizing the goals and objectives. These prioritized goals have to be kept in the product backlog which is used as user stories. Tools such as sticky notes, flip-charts, story boards, whiteboards and mike have to be used for the meeting. Agile coach has to assign and finalize team members for project deployment. Team

members should be allowed to express their views, interact on the user stories and the approximate days required for completing the user story. Team should produce a report about the Minimum Marketable Feature that they wish to deliver at the end of every sprint. It should also contain details such as possible assumptions and the delivery dates. Such report will be called the release plan. Release plan has to be submitted to the customer. Initial sign off has to be obtained from customer in order to proceed. Preparation of release planning report is a good practice to showcase the shared responsibility of what has to be delivered at the end of each release and collective ownership among the members of the team. This practice will make the product owner and the team to be aware of the focus for the forthcoming release and the desired release dates.

RESULTS AND DISCUSSION

Agenda preparation by the facilitator: The facilitator of a meeting has the responsibility to prepare the introductory note for the meeting, goals, agenda updates, Product vision, metaphor description by the product owner, Time-boxing required for each release and iteration. Facilitator should also calculate the capacity with respect to workload per iteration for the full delivery team. This helps to ensure the budget flow required for the project. He is responsible for Sign off document for deliverables and the scheduled date of delivery. Facilitator should determine dependencies and find solutions for the same

from the members present in the meeting. He should make arrangements to review the discovered risks and issues that arise. It is the responsibility of the facilitator to ensure commitment by all attendees of the meeting.

Iteration planning meeting: Product owner, agile coach and team members have to be present for this meeting. They have to join hands in reviewing the user stories before it is taken up for the consequent iteration. Iteration planning meeting has the resemblance to release planning meeting. Members of the team should be given equal work load. Agile coach has to break the epic (bigger stories) in to small user tasks. These prioritized small tasks written as sticky notes have to be placed in the story/scrums boards. Results of individual teams have to be evaluated in the presence of agile coach and product owner to check dependencies which were not seen during the release planning meeting. Sign up from team members can be had before starting the iteration. Before starting the iteration, the task taken up by the team members have to be agreed by agile coach. Dependencies should be carefully dealt with. Iteration planning meeting should be held for less than two hours. Iteration planning meeting will increase the confidence level of an individual team member.

Iteration review meeting: Agile coach and the team members should attend this meeting. After showing the working deliverable of the MMF to the customer, feedback has to be obtained from the customer and it has to be jolted and maintained by agile coach. Incomplete stories have to be identified and it should be considered for the next iteration or it should be placed in the parking lot or the product backlog so that it could be taken up at a future date based on the need. The iteration review meeting has to be limited to one hour duration. By having an iteration review meeting, the members in the team will get a chance to share their experience and as well learn from their experience.

Daily scrum meeting: Before starting the daily scrum meeting, the team has to follow certain rules. The rules denote the way the meeting has to be conducted. The Daily Scrum meeting should be held in the same venue at an appointed time on every working day. Mornings as the first task will be the best time for conducting the meeting. This will be of great help to know the team member's progress of work that was carried out the previous day, the hindrances that they faced, the task that is to be completed for today and the future course of action on the task that is assigned by the scrum master. All Team members should be punctual in attending. Late comers should not be entertained by the scrum master. If for some

unavoidable reasons if a team member is not in a position to be physically available for the meeting, then it is his responsibility to delegate a member on his behalf and the designated member has to report the status of work that was carried out. As a first round of discussion, the Scrum Master should start the meeting with the person to his/her left and proceed till every member present in the discussion hall has spoken. Each Team member should respond only to the following questions and no other discussion should be entertained.

An agile developer has said that "Four developers and one Scrum Master. The meeting was crisp and concise. The members discussed what they had achieved the previous day and then discussed what they plan for the current day. They were also able to spot and resolve dependencies and impediments by informing each other of their daily progress. There was a relaxed and professional behavior throughout the meeting venue. Other than this, no other questions were raised. The meeting lasted for about twelve minutes. The Scrum Master supplied additional information for resolving certain issues and asked one of the team members to resolve the issue that another team member faced. Then the team went back to their regular work" (Agile developer, Bengaluru, India).

Scrum team and individual member progress are assessed through daily scrum meeting. A scrum meeting will last long for a maximum duration of fifteen minutes and only three vital questions are asked (Agile Coach, Bengaluru, India).

The daily standup meeting will act as a good practice that will give shared responsibility among team members and will serve as a self-checking tool. Any slackness in progress will be immediately visible during a meeting and brings on peer-pressure to deliver. Product owner, scrum master and team members who were responsible for developing a particular sprint will be present in the meeting (Scrum Master, Chennai, India).

The scrum master is responsible to move the meeting briskly from one person to the other. Team members should address the entire team and it has to be an egoless team. At a time, only one person should be allowed to talk during the meeting. Others should listen without any commotions. Discussions on any issues should be taken up separately after the Daily Scrum Meeting. Non Team members can be allowed to attend the Daily Scrum Meetings but they should not be allowed to discuss anything during the course of meeting. It is the responsibility of the scrum master to limit the non team members so that the meeting can be conducted in a focused manner. Minutes of meeting has to be recorded and to be sent as a mail to all the meeting participants.

Retrospective meeting: Retrospective meeting is connected at the end of every sprint. This meeting is conducted to share the experiences among the developers and the lessons learnt during the tenure of the entire project. Recommendations for the conduct of retrospective meetings have been classified under three different categories as before, during and after the meeting. Retrospective meetings provide valuable opportunities for the team to learn from their past experience. After the meet they are in a position to evaluate them and improve their future potential (Derby and Larsen, 2006).

Before the start of retrospective meeting: Retrospective meeting should be held at end of every sprint. Agile coach and the team should collectively decide on the meeting participants. Member participation should depend on the topic that is to be shared. A facilitator should be present for each meeting. The facilitator should be a preferable member who does not belong to the current team.

During the course of retrospective meeting: Product owner, agile coach and the members of the team have to be present for this meeting. Meeting should have a brain storming session. Team should share their experiences on what went well and what went wrong during the previous iteration. The members have to come out with a solution plan. Based on the issues that the team faced, proper corrective action plan has to be designed. All team members who were involved in that particular project iteration should be made available for discussion. By conducting retrospective meeting, the team members will get a chance to share their experiences and paves way for collective ownership within the team.

After conducting retrospective meeting: The improvements have to be carried out in the forthcoming sprint. The things that went on well should be shared as a mail to the members outside the team.

Common mistakes of the retrospective meeting: The scrum master or any one member should not dominate the entire team. This will result in a negative impact which may finally end up in not giving chance to the other members of the team to share their view points. The team members should not skip the meeting and has the right to participate in the discussion. Unnecessary tools like laptops, mobile phones and other electronic gadgets should not be brought inside the discussion hall which will disrupt the smooth conduct or will divert the attention of the team. Non-team members should not be allowed

during discussion. This will make the meeting obtrusive. Meeting proceedings should not be discussed outside the hall. Whatever proceedings that are held, it should be within the walls.

Don'ts in a retrospective meeting: The member should not blame or find fault on others for not having performing the things as expected. Members should not end up in conflicts between other team members. This may as well affect their career growth. The team members should not be allowed to give feedback on poor performers. "Ifs" and "Buts" should be avoided during the course of the meeting.

General guidelines for conduct of retrospective meeting: Meeting should start and end with the stipulated time. Time boxing is a major theme that is to be considered. White Boards, markers and mike should be effectively utilized. Action points from the product owners have to be derived. Every task or discussion completed during meeting has to be displayed in the board in order to avoid repetitive discussion.

CONCLUSION

This study exhaustively discussed about the various meetings carried out when implementing the agile software projects. The entire study was based on the experiences shared by the practitioners who were working in the scrum framework which is part of the agile software process. Special focus was given to the retrospective meeting and the doses and don'ts and the common mistakes in the retrospective meetings. This paper will be highly helpful for the software development communities who have recently transitioned from the traditional software development to agile software development.

RECOMMENDATIONS

Recommendations for overcoming the common mistakes: The meeting should be conducted in an interesting and energetic manner. It is the responsibility of the facilitator to ensure that the discussion adds value. Interaction among team members should be motivated and encouraged.

REFERENCES

Abrahamsson, P. and J. Warsta, 2003. New directions on agile methods: A comparative analysis. Proceedings of the 25th International Conference on Software Engineering, May 3-10, Portland, Oregon, IEEE Computer Society, pp: 244-254.

- Ancona, D.G. and D.F. Caldwell, 1988. Beyond task and maintenance defining external functions in groups. *Group Organiz. Manage.*, 13: 468-494.
- Boehm, B. and R. Turner, 2003. Rebalancing Your Organization's Agility and Discipline. In: *Extreme Programming and Agile Methods*, Frank, M. and D. Wells (Eds.). Springer, Berlin, Germany, ISBN:978-3-540-40662-4, pp: 1-8.
- Derby, E. and D. Larsen, 2006. *Agile Retrospectives: Making Good Teams Great*. Pragmatic Bookshelf, New York, USA., ISBN:9780977616640, Pages: 170.
- Dyba, T. and T. Dingsoyr, 2008. Empirical studies of agile software development: A systematic review. *Inform. Software Technol.*, 50: 833-859.
- Ganesh, N. and S. Thangasamy, 2010a. Adoption of scrum methodology: Insights from India. *Int. J. Software Eng. Technol.*, 1: 1-6.
- Ganesh, N. and S. Thangasamy, 2010b. Integration of agile software development with the existing organizational practices and thereby improving the software process. *J. Technol. World*, 1: 316-320.
- Ganesh, N. and S. Thangasamy, 2011. Issues identified in the software process due to barriers found during eliciting requirements on agile software projects: Insights from India. *Int. J. Comput. Appl.*, 16: 7-12.
- Grisham, P.S. and D.E. Perry, 2005. Customer relationships and extreme programming. *Proceedings of the Workshop on Human and Social Factors of Software Engineering*, July 2005, ACM, St. Louis, Missouri, pp: 1-6.
- Korkala, M., P. Abrahamsson and P. Kyllonen, 2006. A case study on the impact of customer communication on defects in agile software development. *Proceedings of Conference on the AGILE 2006 (AGILE'06)*, July 23-28, 2006, IEEE, Finland, Europe, ISBN:0-7695-2562-8, pp: 1-11.
- Mann, C. and F. Maurer, 2005. A case study on the impact of scrum on overtime and customer satisfaction. *Proceedings of the Conference on Agile Development Conference*, July 24-29, 2005, IEEE, Washington, USA., ISBN:0-7695-2487-7, pp: 70-79.
- Moe, N.B., T. Dingsoyr and T. Dyba, 2010. A teamwork model for understanding an agile team: A case study of a Scrum project. *Inform. Software Technol.*, 52: 480-491.
- Nerur, S., R. Mahapatra and G. Mangalaraj, 2005. Challenges of migrating to agile methodologies. *ACM. Commun.*, 48: 72-78.
- Taylor, P.S., D. Greer, P. Sage, G. Coleman and K. McDaid *et al.*, 2006. Do agile GSD experience reports help the practitioner?. *Proceedings of the 2006 International Workshop on Global Software Development for the Practitioner*, May 23-23, 2006, ACM, Shanghai, China, ISBN:1-59593-404-9, pp: 87-93.
- Thangasamy, S., 2012a. *Essentials of Software Engineering*. Wiley, India.
- Thangasamy, S., 2012b. Lessons learned in transforming from traditional to agile development. *J. Comput. Sci.*, 3: 389-392.
- Whitworth, E. and R. Biddle, 2007. The Social Nature of Agile Teams. *Proceedings of the IEEE Conference on Agile Development Conference*, August 13-17, 2007, IEEE, New York, USA., ISBN:0-7695-2872-4, pp: 26-36.