International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 07
June 2017

Challenges in Agile Software Development

A.M.CHANDRASHEKHAR[#1]

MANASAMITHRA P[#2]

[1] *Assistant Professor, Department of Computer Science & Engineering, Sri Jayachamarajendra College of Engineering(SJCE),JSS S&T University Campus, Mysore, Karnataka, India*

[2] *M.Tech 2nd Semester, Department of Computer Science & Engineering, Sri Jayachamarajendra College of Engineering(SJCE),JSS S&T University Campus, Mysore, Karnataka, India*

Abstract- **Software Engineering is an emerging field in Computer Science, since there is a need of systematic development of software product. The most popular method, engineers have been following was Waterfall or traditional approach. But it became obsolete to fulfill the customers' need more quickly and efficiently. Agile Software development (ASD) is the software development method where design, development and testing are interleaved to evolve the solution with the effort of self organizing and motivating team. ASD is one of the most widely used Software development method today. Agile methodology has many sub methods like Extreme programming (XP), Scrum, Adaptive Software Development, Lean etc. All the Agile approaches significantly increase the software development performance, customer satisfaction. Due to the inability of the traditional software development processes such as waterfall, spiral model to satisfy the customer, organizations slowly stepped towards adapting the agile methods. Agile Software methods are more informal and it provides the team members the freedom to work in their own way leading to more productivity in the work. Although there are several advantages, it also has some challenges. This paper gives an overview of the challenges in ASD along with possible solutions.**

Keywords- **Adaptive Software Development, Agile Software Development, Challenges, Customer Satisfaction, Extreme Programming (XP), Lean, Scrum**

I. INTRODUCTION

Software development is the process of producing the software product by following systematic steps called software development process. Process involves requirement collection and analysis, Design, Development, Testing and maintenance. It is very important for any software developers to follow these steps for the successful software product delivery. But in this rapidly growing world, there is tough competition in every field including software production. So, everyone wants their product to be quickly delivered. Software developers should be capable enough to accommodate changes to the product as customer requirement changes continuously. Traditional software method mentioned earlier did not provide solutions to satisfy rapidly changing customer needs more quickly and efficiently. As an alternative solution, agile methodology has evolved. This method provides more flexible and productive approach to software development.

Remainder of the paper is organized as follows. Section 2 discusses Transition from Traditional to Agile. Section 3 provides explanation about different types of Agile Software Development methods. Section 4 is the challenges involved in agile process followed by conclusion and reference in section 5 and 6.

II. TRANSITION FROM TRADITIONAL APPROACH TO AGILE

Traditional software development methods are plan driven approach, which requires organized process flow and milestones [3]. But in the late 19th century and early 20th century, traditional approaches tend to become obsolete because of their extensive documentation which makes the overall work time consuming. It did not satisfy the continuously evolving requirements of the customer in a fast and efficient fashion. In this period people expected more quickly delivery of the product with slight compromise in the quality. Agile methodologies have come into existence to deliver software products with limited time and cost constraints (Shown in Figure 1), rapidly changing and complex business requirements.

Traditional approach focuses on documentation which is the only progress measuring tool. But this will consume more of Developer's time in writing than in coding. Agile mainly focused on core part of the process which is development or coding which made it capable of responding to on demand changes in the software in very less time. Customer interaction is very important in successful production of the software. Traditional approach does this interaction only in the beginning and end of the process. But that may not be sufficient. Possibility of missing any functionality and misunderstanding of any requirements is more in the case of traditional approach. Agile provided customer interactions throughout the process to overcome the shortcomings of traditional approach. Agile works with the principle that 'All are equal" means that any problems come in, it is the responsibility of the whole team to solve it. Agile has overcome many disadvantages of traditional process. So, software development companies started extensively adapting agile development approach to their project to with stand in the competitive world.

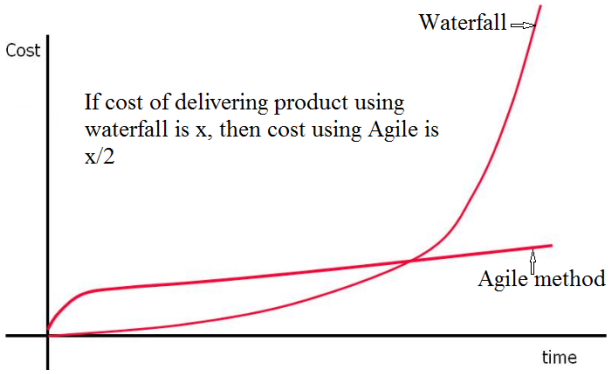

Figure 1. Reason for Transition from Waterfall to ASD

III. WORKING OF AGILE DEVELOPMENT

Agile software development is a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. There are different methods in ASD, namely Extreme programming (XP), Agile modeling, Adaptive Software Development, Lean Development, Crystal methodologies family, Feature-Driven Development, Scrum, Agile Unified Process [2].

Extreme Programming (XP): Extreme Programming was introduced by Kent Beck in 2000. It provides a set of simple, specific principles and values that helps in all the four phases of software development namely planning, coding, designing and testing. It accepts changes at any time during the development. It supports develop-feedback cycle.

Agile Modeling: Agile Modeling (AM) was established by Scott Ambler in 2002. This includes principles of XP along with knowledge of purpose of modeling [2]. It focuses on stakeholder participation and correctness of modeling.

Adaptive Software Development: This method mainly has three phases namely Speculation, Collaborations and learning. Its distinguishing features are mission driven planning, component based focus, explicit consideration of risks, emphasizes collaboration for requirements gathering and emphasizes learning throughout the process.

Lean Development: Lean works with principles that eliminate the waste and build the quality in product. Eliminating the waste means that avoid useless meetings which consume time, eliminate unnecessary documentation and tasks. Deliver the product very fast by avoiding the inefficient ways of working and optimizing each individual's work in the team.

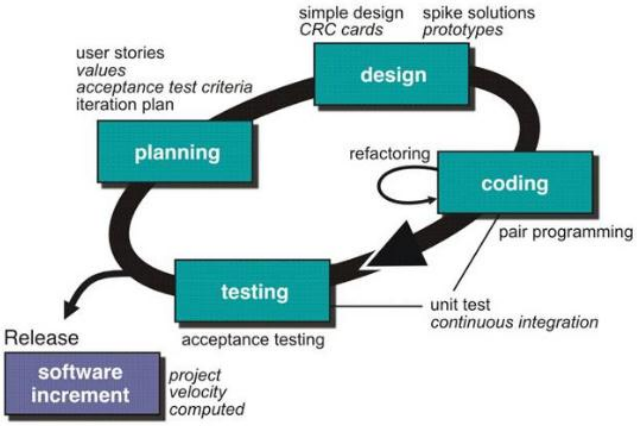

Figure 2. General representation of ASD

SCRUM: SCRUM methodology was initiated by Ken Swaber in 1995. SCRUM came up with the idea of simplicity in process, documentation and communication. This is achieved by dividing the process into tasks. Three roles involved are product owner, SCRUM team which includes developers, testers, and others, and Scrum masters for keeping the team focused on the goals [9]. Particular iteration is expected to complete within certain time called sprint.

Agile Unified Process (AUP): Uses Agile principle by keeping the concepts of Rational Unified Process (RUP). It uses the techniques which are used in Agile modeling and Test driven Development.

IV. CHALLENGES IN AGILE SOFTWARE DEVELOPMENT

Although Agile has many benefits as mentioned in Section II, it also leads to many challenges. Here, the challenges are briefed according to their priority. For a project to be successful using ASD, team should effectively overcome all the challenges with suitable solutions.

- Team Management

Team management includes selection of team members, coordination between team members, relationship among the team members, behavior and other team practices. Since Agile Development is aggressive, all the team members should be excellent team players, quick learners, flexible to manage with changes, self learners and good speakers. Practically, it is impossible to have all humans with same characteristics with all the above mentioned qualities. So, selection of team members is very difficult task. When mixture of all types of people is present in the team, there should be a good relationship and collaboration between them. There is no concept like single individual rolling

entire team. All members in the team should do all the work which is called as collective ownership. So, it is necessary to help among them to achieve the target within sprint. Main principle of agile is to accommodate changing customer requirements. So, the developers should be quickly learn the technologies from external source or by self learning needed for making these changes. Also, communication skills of the team members should be very good to convey the changes that they have made to other team members as well as to convince the customer that output is as expected. All the work will be waste, upon failure in convincing the customer. Lack of training in certain technology or uncertainty in specific practice will be a challenge [4].

First and foremost thing to manage the team members is that selection of right members through extensive interview process so that team will find the members with almost all the qualities that the agile needs. Also, while selecting the members for each agile lane, make sure to include people with variety of technical excellence, work experience to strengthen the team. Team lead and the project managers should be capable enough to keep the team active by conducting extra fun activities and providing the promotions, bonus etc so that interest of working will be retained in the team members.

- Distributed Team

Agile supports geographically distributed teams. It lowers the production cost and provides an opportunity to include skilled people around the globe [1]. But the challenges such as Distance difference, Team configuration, customer communication, project characteristics, organizational factors and Human factors [5] exists. Since teams are at different locations, communication between the teams is difficult as they cannot conduct face to face meetings which would have been more effective in conveying the subject. Trust among the team members is very important. Without mutual trust project will almost be a failure. Agile considers customers as part of the development team. So, including onsite customers in the daily stand-up calls is very difficult as they will have to attend all the team meetings which are distributed across the globe. This will become a burden for customers even though different works in different time zones. The lack of knowledge about the project characteristics, such as whether the product is generic or bespoke, structure reduces the knowledge sharing of teams. Organizational factors like the tools used such as email tools, instant messaging, licensed software will be a challenge [24]. Human factors include pronunciation, culture etc. Adjusting all different team practices is a difficult task.

As communication technology is grown up to meet any challenges, it can provide an environment which provides effect of face to face meetings. Making all the joined people involved in the meeting, there will be less possibility of staying away from the system even though joined the meeting for the purpose of attendance. This can be done by scrum master by addressing each individual.

- Requirement Prioritization

Requirements are provided by the customers. Developers should develop in the iterations with critical requirement in the first iteration, next critical requirement in the next release and so on. But the problem is customer will think every requirement is mandatory and not able to prioritize it according to developer's perspective. Also, project involving multiple stakeholders and multiple customers, it is difficult to prioritize the requirements [1].

Analyzing the requirement from the technical perspective is difficult for a non technical person like business stakeholders. But as a solution to this, technical persons are involved in the client's side as well.

- Lack of Documentation

Although there are many advantages of having minimal documentation in agile development, it leads many problems. Customers are less skilled persons with less knowledge about technical aspects. Difficult to convey the requirement changes to the customers without documents. So, it will be easy for them to understand the progress via documents rather than the explanations or the deliverables. For the beginners, it will be easy to understand if some documents are available. Since the requirements are not well documented, team will not get a clear picture about overall system which may sometimes lead to wrong understanding of the module. Also, team will not have evidence in the future in case of audits in the organization which will lower the rating of the team.

Documentation helps in learning for new developers. But conducting the Knowledge Transfer (KT) session to deliver the concepts will be more effective than the documents. It will help the fresher to understand more clearly even though initially takes some time.

- Changing and Over scoping requirement

ASD is a method where the requirement changes are most welcomed. But continuous involvement of the customer may lead to unnecessary and unsystematic changes. Since the developers will be explaining the progress in meetings, there is a chance of misunderstanding by the customer so that they will ask to change some of the functionalities even though developed functionalities are correct [1]. If the requirement complexity is not understood properly in the initial stages, there will be unexpected changes during the iterations which will kill the time. Improve the communication skills of the developers and cultivate the capacity to convince the customers. This will prevent the clients from misunderstanding the functionality developed

- Organizational challenges

If an organization wants to change to Agile, it is not a one day task. There should be continuous collaboration between business and IT. Management will treat ASD as another software development method and insist to adopt as soon as possible which will make the developers to start with incomplete setup for the ASD. There is a possibility that Teams successfully adopt agile but operate in an environment where wider organizational structures are more traditional [4]. Organization culture should be such that there should be trust over the entire process by the management. As a whole, there should be full support from the senior management for the agile process as there is no formal documentation. With proper planning this challenge can be overcome [21].

- Agile Software Development Process Challenges

Agile process includes the activities like deciding the sprint length, predefining the deployment date, last date for development and testing start date, completion of regression testing etc. Problem here is the sprint length may be too short or too long. Because of extended sprint date, there will not be specific date on which customer gets the functionality get deployed in production. Using of different process types, testing for regular compliance are also challenges related to process [1]. Due to the continuous support for the deployment, developers may feel uncomfortable incase regularly same persons are getting the work.

Incorporating automated testing rather than the manual testing will help to address this issue. Try to keep the sprint length in an average that suits both developers and the customers. Keeping the small releases such as 2-3 weeks rather than months will help address this problem.

- Progress Monitoring and Feedback

Studies found that ASD is lack of progress monitoring, has long or no feedback loops, and often seize invisible progress. Related to feedback, ASD (especially Scrum) has sprint review phase at the end of the sprint where the Product Owner and key stakeholders assess the progress of product deliverable and provide feedback. Often, team has difficulties to have direct contact with customer. Consequently, because feedback from customer is only received after the customer starts using the product, there is a possibility of changes to the product at a late phase of development that may cause a lot of rework. Furthermore, the progress of the work is not well presented. It can led to lacking of progress awareness of stakeholder. There is also little attention given to reviews and meetings from management [1].

Pair programming makes the task easy which implicitly does the code review and provides the feed back to one another who are working in pair by sitting together in a single machine. Unit testing and regression is implemented for the reason of feedback quickly. Project managers should conduct team meetings at least once in a week so that all the team members will know the status of the deployed product in the production. Project manager will have the detailed information due to attending the meetings with clients more frequently than others.

- Shared Decision making

Agile software Development supports decision made by many people in collaboration rather than a single man leadership. But there are mainly three challenges regarding this are, the process of aligning strategic plans with iteration plans, the process of allocating resources and the process of performing tasks. Lack of shared understanding, not understanding the complexity, No arena for solving conflicts, not involving the team are the major problems between product owner and the development team. All the team members in the agile team must be self motivated such that they should ask the work as they finished the current work. Also, some people who are idle should help others in finishing their work which is not implemented in reality many times. Important decisions not aligned, missing a clear prioritization, missing a definition of done, Conflicting priorities within the company, Low committing to the plan are the risks involved in resource allocation. Not confronting each other, Lacking team orientation, Unrealistic plans, Decision-hijacking, Lack of knowledge, Technocracy includes in performing tasks together [7].

- Continuous Integration (CI) and Deployment

There are challenges like merge conflicts, deployabilty status of builds includes in continuous integration. Maintaining strong version control system like IBM clear case or TFS in Visual Studio will eliminate the risk of code merge. These tools are used in order to make the continuous integration of the module as they are build [15]. Though the deployment is continuous, if it is planned and assigned in routine to the team members will eliminate the burden.

Continuous includes many risks such as infrastructure problems which include it should have proper hardware and software to handle the Deployment process. Deploying product by adapting small batches and seamless upgrades are other risks. Product quality may decrease due to bug slip and lack of code review leads to low test quality. Strong Version control system is necessary to maintain single code branch and resistance to changing database schemas due to unplanned code changes. Developers may have to face the overtime work many times due to immediate deployments. Other risks include changing team roles, product marketing, and customer adoption; shorten customer feedback, technical product writing, and customer feature discovery [6].

- Agile in non-agile environment

If we adopt the agile process where it actually does not match that environment, there will be lot of problem that might be faced. There will be lot of limitations of organizations, changing organizational culture and sustainability [4]. For example, Agile encourages automated testing since it needs to be released very fast. If team is still stick to manual testing, then that will not suit the agile. Simplicity is the main motto of ASD. If the team is not accepting the design to keep simple or disagreement in learning the technology which makes the task simpler, then whole idea of Agile will be lost [11]. Sometimes people will not follow the coding standards which makes difficult for others to understand the code. Refactoring is the method to keep the code simple.

- Communication Challenges

Main theme of ASD is the communication between developers and client. If there is lack of communication, success rate will be very low. If clients are not involved in the stand up calls, they will not aware of the progress and become isolated from the team. It is better to meet the offshore team at least once in two years. It will be difficult to convey the intended messages in the meetings in case of poor communication skills [8]. This can also happen because of the uncommon language pronunciation. Communication using video conferencing tools and desktop sharing will not be effective as much as face to face meetings. Involve the customer in the process as much as possible[12]. Provide innovation ideas and assets to the customers which will motivate them and help increase the revenue of both.

- Customer cooperation

In the continuous development process of Agile, cooperation of customer plays very important role. They should be involved in the development team. Unavailability of the customer makes the interaction minimal and work progresses slow. If the team is geographically distributed, poor coordination among multiple vendors or stakeholders leads to discrepancy. There is also possibility that the customer has not clearly defined the user stories which makes the developer understanding difficult. Customer interaction will help clarify the doubts [14].

- Testing Challenges

Agile also has some difficulties in testing. Most of the times, the role of the developer and tester is blurred. So it is not enthusiastic to continue with their work [13]. Also, it is possible that the testing team is from different domain and lack of knowledge about the technology they are currently working in. Testers should support the development tem through their valuable feedback.

- Other Challenges

Other common risks include Testing Challenges, Neglecting the Need for Technical Excellence Training, difficulty in Testing, Project Management issues, Product quality, lack of training, No architecture, issues in Budgeting during contract agreement, meeting non functional requirements etc [1, 10]. When the migration to the new technology is in need, lack of technical training will make the organization lagging behind. The competitor will win who gave training. Different modules are developed in different agile lanes; testers may have to repeat some test cases which are related. Since there is no complete architecture and it is impatient, there is a compromise on product quality. Since the product is developed within limited time, efficiency, reliability, and maintainability are maintained only to a certain extent.

V. CONCLUSION

Software development became main concentration of software developers in the early 20th century. Different software development processes has been evolved. ASD is the dominating method today, due to its capacity to respond quickly to the changing business requirement. There are several agile development methodologies that will help improve the process in different ways like XP, Adaptive software development, Scrum, Lean development etc. But it also imposes certain challenges. This paper discussed the different challenges involved in various agile methodologies. The challenges are not restricted to particular role. It may be with respect to developer, Customer or project managers or contractors etc. If the organization has not overcome these challenges, that will be a bad threat on their performance as well on the profit and revenue. So, possible solutions to the corresponding to the challenges are discussed.

REFERENCES

[1] Widia Resti Fitriani1, Puji Rahayu2, Dana Indra Sensuse, "Challenges in Agile Software Development: A Systematic Literature Review", ICACSIS, 2016

[2] Malik Hneif, Siew Hock Ow "Review Of Agile Methodologies In Software Development", International Journal of Research and Reviews in Applied Sciences, Volume 1, Issue 1, October 2009.

[3] Rami Hasan AL-Ta'ani, Rozilawati Razali, "Prioritizing Requirements in Agile Development: A Conceptual Framework", Procedia Technol., vol. 11, no. Iceei, pp. 733 739, 2013.

[4] Peggy Gregory a , Leonor Barroca b , Helen Sharp b , Advait Deshpande b , Katie Taylor a, " The challenges that challenge: Engaging with agile practitioners' concerns", Information and Software Technology, April 2016.

[5] Yehia Ibrahim Alzoubi a, Asif Qumer Gill a, Ahmed Al-Ani, "Empirical studies of geographically distributed agile development communication challenges: A systematic review", Information and Software Technology, 28 August 2015

[6] Gerry Gerard Claps, Richard Berntsson Svensson, Aybüke Aurum, " On the journey to continuous deployment: Technical and social Challenges along the way", Information and Software Technology, 23 August 2014.

[7] Nils Brede Moe, Aybüke Aurum, Tore Dybå, "Challenges of shared decision-making: A multiple case study of agile software development", Information and Software Technology, 28 November 2011.
[8] Suprika V. Shrivastava, Urvashi Rathod, "Categorization of risk factors for distributed agile projects", Information and Software Technology, 21 July 2014.
[9] Zahra Askarinejad Amiri, "Challenges And Weaknesses Of Agile Method In Enterprise Architecture", International Journal of Computer Science & Engineering Survey (IJCSES) Vol.3, No.6, December 2012
[10] https://dzone.com/articles/top-3-challenges-for-agile-transformation
[11] A. M. Chandrashekhar and K. Raghuveer, "Confederation of FCM Clustering, ANN and SVM Techniques of Data mining to Implement Hybrid NIDS Using Corrected KDD Cup Dataset", Communication and Signal Processing (ICCSP) IEEE International Conference,2014, Page 672-676.
[12] AM. Chandrashekhar and K. Raghuveer , "Improvising Intrusion detection precision of ANN based NIDS by incorporating various data Normalization Technique – A Performance Appraisal", IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014.
[13] A. M Chandrashekhar A M and K. Raghuveer, "Diverse and Conglomerate modi-operandi for Anomaly Intrusion Detection Systems", International Journal of Computer Application (IJCA) Special Issue on "Network Security and Cryptography (NSC)", 2011.
[14] A. M Chandrashekhar A M and K. Raghuveer, "Hard Clustering Vs. Soft Clustering: A Close Contest for Attaining Supremacy in Hybrid NIDS Development", Proceedings of International Conference on Communication and Computing (ICCC - 2014), Elsevier science and Technology Publications.
[15] A. M. Chandrashekhar and K. Raghuveer, "Amalgamation of K-means clustering algorithem with standard MLP and SVM based neural networks to implement network intrusion detection system", Advanced Computing, Networking, and Informatics –Volume 2(June 2014), Volume 28 of the series Smart Inovation, Systems and Technologies pp 273-283.
[16] A. M. Chandrashekhar and K. Raghuveer, "Fusion of Multiple Data Mining Techniques for Effective Network Intrusion Detection – A Contemporary Approach", Proceedings of Fifth International Conference on Security of Information and Networks (SIN 2012), 2012, Page 178-182.
[17] A. M. Chandrashekhar, Jagadish Revapgol, Vinayaka Pattanashetti, "Big Data Security Issues in Networking", International Journal of Scientific Research in Science, Engineering and Technology (*IJSRSET),* Volume 2, Issue 1, JAN-2016.
[18] Puneeth L Sankadal, A. M Chandrashekhar, Prashanth Chillabatte, "Network Security situation awareness system" International Journal of Advanced Research in Information and Communication Engineering(IJARICE), Volume 3, Issue 5, May 2015.
[19] P. Koushik, A.M.Chandrashekhar, Jagadeesh Takkalakaki, "Information security threats, awareness and cognizance" International Journal for Technical research in Engineering(IJTRE), Volume 2, Issue 9, May 2015.
[20] A.M.Chandrashekhar, Yadunandan Huded, H S Sachin Kumar, "Advances in Information security risk practices" International Journal of Advanced Research in data mining and Cloud computing (IJARDC), Volume 3, Issue 5, May 2015.
[21] A. M. Chandrashekhar,Muktha G, Anjana D, "Cyberstalking and Cyberbullying: Effects and prevention measures", Imperial Journal of Interdisciplinary Research (IJIR), Volume 2, Issue 2, JAN-2016.
[22] A.M.Chandrashekhar, Syed Tahseen Ahmed, Rahul N, "Analysis of Security Threats to Database Storage Systems" International Journal of Advanced Research in data mining and Cloud computing (IJARDC), Volume 3, Issue 5, May 2015.
[23] A.M.Chandrashekhar, K.K. Sowmyashree, RS Sheethal, "Pyramidal aggregation on Communication security" International Journal of Advanced Research in Computer Science and Applications (IJARCSA), Volume 3, Issue 5, May 2015.
[24] A.M.Chandrashekhar, Rahil kumar Gupta, Shivaraj H. P, "Role of information security awareness in success of an organization" International Journal of Research(IJR), Volume 2, Issue 6, May 2015.
[25] A.M.Chandrashekhar, Huda Mirza Saifuddin, Spoorthi B.S, "Exploration of the ingredients of original security" International Journal of Advanced Research in Computer Science and Applications(IJARCSA), Volume 3, Issue 5, May 2015.
[26] A.M.Chandrasekhar, Ngaveni Bhavi, Pushpanjali M K, "Hierarchical Group Communication Security", International journal of Advanced research in Computer science and Applications (IJARCSA), Volume 4, Issue 1,Feb-2016.