Moving Towards Agile Testing Strategies

Dr. Kiran Kumar Jogu[1] and Dr. K. Narendar Reddy[2]
[1]IBM India Software Lab, Hyderabad, India
kirankumarj9@gmail.com
[2]CVR College of Engineering, Dept. of IT, Hyderabad, India
knreddy@cvr.ac.in

Abstract **- Testing is a vital activity for delivering a high quality software product to the customers. Often testing accounts for more project effort and time than any other software development activities. Traditional software development process models are being used for long time in software development. Recently, the software development in the industry is moving towards agile due to the advantages provided by the agile development process. One of the main advantages of agile software development process is delivering the high quality software to the customers in shorter intervals. Another important advantage is, the agile process models embrace the changes in requirements at any stage of software development. Due to these advantages, software industry is showing interest in software development using agile process models. One of the agile process models which is being used in the software industry is, the "scrum". Since testing plays a major role in the success of the product, it is given a lot of importance in software development. Testing strategies for conventional process models are well established, but these strategies are not directly applicable to agile testing without modifications and changes. One of the important current research areas is the agile software testing strategies. The main objective for any agile testing strategy is to reduce the testing time and at the same time ensuring the software quality. In this paper, a strategy for agile testing in the scrum software development environment is proposed and presented. The sprint activities which form the context for the proposed testing strategy are also proposed. The tools which are helpful for automation which is part of the strategy are presented. The advantages of the proposed strategy are highlighted. Case study which used the proposed strategy is presented, it indicated that the number of bugs reported by customer reduced significantly in scrum agile development using the proposed testing strategy**.

Index Terms **– Agile software development, traditional development, scrum, software industry, testing strategy, automation.**

I. INTRODUCTION

Traditional software development process models are being used for long time in software development. Present business demands the software products to be delivered in shorter intervals and software development environment having capability to embrace change at any stage of development. Traditional process models have difficulty in responding to change which often contributes success or failure of a software product [1]. Software requirements are dynamic which are driven by industry market forces. Agile approach to software development is suitable to such situations [2], [3]. Hence, more software companies are making a transition to agile software process models from traditional (plan-driven) software development process models (like Rational Unified Process (RUP), Waterfall, or V-model). Some of the key factors for success in an agile testing approach are: adopting an agile mindset, automating regression tests, collaborating and obtaining feedback from customer [4]. Some issues may arise when transition is made from traditional development to agile development. Common issues for agile models after migration from traditional models were identified in [5]. They are related to testing, test coverage, coordination overhead, and software release. In this paper we focused on testing related issues. Agile methods employ short iterative cycles, with prioritizing the requirements which actively involve users. Agile process models are iterative, incremental, self organizing and emergent [6]. One of the agile process models which is being used in the software industry is "scrum". Scrum agile process model is defined in [7], [8]. In agile software development, testing is a vital activity for delivering a high quality software product to the customers. Often testing accounts for more project effort and time than any other software development activities. Since testing plays a major role in the success of the product, it is given a lot of importance in software development. Testing strategies for conventional process models are well established, but these strategies are not directly applicable to agile testing without modifications and changes. One of the important current research areas is the agile software testing strategies. The main objective for any agile testing strategy is to reduce the testing time and at the same time ensuring the software quality. In this paper, a strategy for agile testing in the scrum software development environment is proposed and presented. The tools which are helpful for automation which is part of the strategy are presented. The advantages of the proposed strategy are discussed and highlighted.

The remainder of this article is structured as follows. Related work is briefly described in Section II. In Section III, the scrum agile process model is described. In Section

 DOI:10.32377/cvrjst0515

IV, the scrum agile testing strategy is given. Subsequently, conclusions are presented and future directions are proposed.

II. Related Work

Software industry is transitioning to agile methodologies from traditional approaches. One of the popular agile process models which is being used in software companies is "scrum". Scrum main characteristic is, continuous deployment of working product increment after each iteration (sprint). As per the survey on agile methods given in [9], 54% of the software companies who are using agile methods are using Scrum. In the survey conducted by [10] on agile projects in different countries found that six critical factors contribute to agile project success. These factors are: agile software engineering techniques, customer involvement, project management process, team environment, team capability, and delivery strategy. One of the attributes related to the critical factor "agile software engineering techniques" is testing strategies. To address the above mentioned critical factor and its associated attribute, currently research is being carried out on agile testing strategies [11], [12]. In this direction, authors of this paper proposed a testing strategy for scrum agile software development environment.

III. Agile Software Development Using Scrum

To provide consumers with continuous deployment of new features rapidly with the capability of embracing change at any stage of development, scrum is ideally suited for this purpose [7], [8], [13]. The scrum agile model is an iterative, incremental process of planning, development, testing, and deployment. In scrum at the end of each iteration (sprint) a working increment is released and deployed. In XP (eXtreme Programming) at the end of an iteration, the working product may not be available. Hence, scrum leads to continuous deployment when compared to XP. Due to scrum's main characteristic of continuous deployment software industry is transitioning to scrum agile software development. The scrum model is depicted in the following diagram (Fig. 1) adopted from [7]. The model shown in Fig.1 is depicting the artifacts of their underlying activities. The main framework activities of the agile process model are: Creation of product backlog, Planning, (Creation of sprint backlog and expanding the sprint backlog), and Sprint (consists of development activities). The scrum activities are performed by the scrum team which consists of product owner, development team, and scrum master.

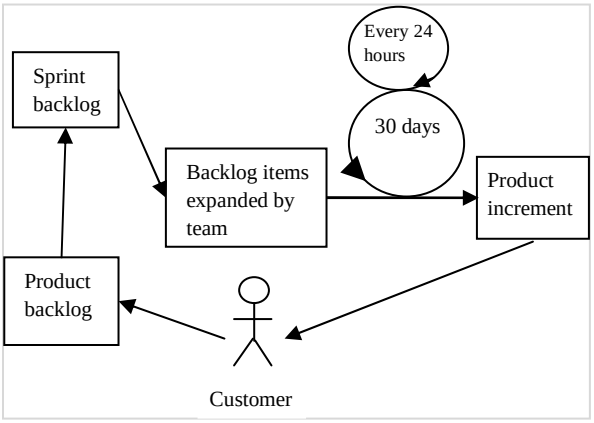

Figure 1. Scrum agile process model

Product owner is responsible for creating and maintaining the requirements in product backlog. He/she creates stories for the requirements in the product backlog. Development team is responsible for developing the product by implementing the features in sprint backlog. The development team is cross functional. Cross functional means, team is responsible for design, development, testing, and deployment. The responsibility of the scrum master is to ensure that the scrum process is followed properly by the team. The scrum activities lead to the following artifacts: product backlog, sprint backlog & task list to achieve sprint backlog, and working software product increment respectively. These artifacts are briefly discussed below.

Product backlog: The required product features or requirements by customer are identified added to product backlog. Features are prioritized as desired by the customer. The main source of agility in scrum model is the prioritized requirements list, which is flexible product backlog [8], [14]. Changes are inevitable. As the needs of the customers change the product backlog is continuously reprioritized. Hence, the software development is flexible. New features are selected from the backlog continuously & integrated and released as a working product increment at the end of the sprint. This means that one can deliver with increasing functionality more frequently, which provides flexibility and the opportunity for adaptive planning [8].

Sprint backlog: During first part of planning product owner and development team together decides which features (user stories) will be part of the next sprint. The high priority features from product backlog are given preference. These features in this backlog are addressed during the sprint. Typical time-box for a sprint is 30 days. The changes (addition of new features) to the features in the ongoing sprint will not be accepted. But, changes (new features) can be added to the product backlog while the sprint is in progress.

Expanded sprint backlog: During second part of planning development team analyses the user stories (features) in the sprint backlog and divides each user

DOI:10.32377/cvrjst0515

story in its tasks. These tasks are handled by different development team members during sprint.

Working software product increment: During sprint development activities are carried out iteratively. Scrum meetings are held daily, typically of 15 minutes duration. Team discusses about the progress and what to be done in next 24 hours. At the end of sprint (30 days), working software product increment is delivered (deployed). Delivered product is evaluated by the customer to ensure that the features in the sprint backlog are implemented.

Testing is important, because it is carried out to ensure the software product quality. And moreover, success or failure of the product depends on testing. Hence, the authors of this paper focused on testing. The proposed testing strategy for scrum model is given in the following section.

IV. Proposed Agile Testing Strategy for Scrum

A. Proposed Strategy

Scrum is a framework for developing software products [15]. Various processes and techniques can be proposed and employed within the framework. Scrum framework specifies the following activities: planning, (Creation of sprint backlog and expanding the sprint backlog), Sprint (consists of activities which can deliver a working software product increment implementing sprint backlog features in a given time-box(typically 30 days)). To propose the strategy for testing, first the sprint activities need to be proposed. One of the possible set of sprint activities can be eXtreme programming (XP) type development activities. The XP development activities could be: design, test driven development & refactoring, integration & regression testing, and validation testing before release. XP activities may not produce a working product increment after completing iteration(s) (in a given time-box). This may be because of the fact that this model is not based on predefined time-box based product release, hence the authors of this paper proposed sprint activities which can deliver the working software product in predefined time-box. The proposed sprint activities are shown in Fig. 2. The activities are: design, development (coding), and testing. They are performed iteratively to produce a working product increment in a given time-box (sprint). The proposed testing strategy is based on these proposed sprint activities. The proposed testing strategy for scrum process model is given in Fig. 3. The sprint activities are carried out iteratively to implement the features (user stories) in sprint backlog. The team for sprint contains scrum master and development team. Development team is cross-functional. They will be able to perform design, coding, and testing (unit testing and integration testing). Some of the development team members (testers) can be specifically meant for regression and functional testing. The responsibilities of the testers are: to plan and update test cases for sprint stories, automate test scripts if possible, execute the tests and report defects, and run regression tests and functional tests at the end of the sprint. Testers are also responsible for testing non-functional tests such as load testing and performance testing.

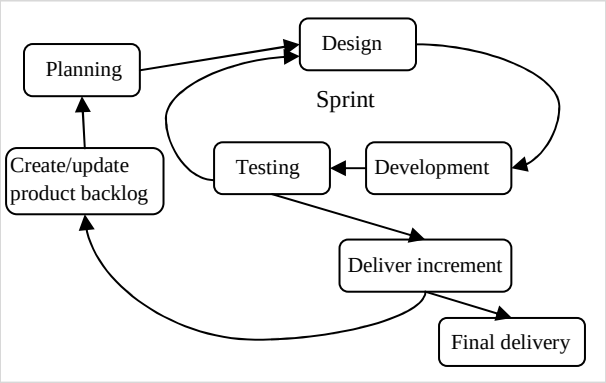

Figure 2. Proposed sprint activities for scrum

The testers in scrum agile software development participate in scrum ceremonies which includes sprint review, planning, daily, and retrospective meetings. The proposed testing strategy for scrum model is depicted in Fig. 3. Testing strategy contains: unit testing, continuous integration, and regression testing which are carried out during the sprint. Whereas, functional & non-functional testing and user acceptance testing is carried out at the end of the sprint. The testing tasks during a sprint are incremental and iterative. Unit testing is done by the developer for finding the logical errors in a module. The bugs found in unit testing are debugged before integrating with other modules. Continuous integration is performed daily. Continuous integration enables to complete the increment in the scheduled sprint time. Regression testing is done after every integration test to ensure that newly integrated module has not introduced any new bugs. Functional test cases are created based on sprint backlog stories and executed at the end of the sprint.

Unit testing, integration testing, regression testing, and functional testing are automated. These testing tasks are conducted repeatedly and frequently, hence, automation will help to reduce the testing time. Since these tests are conducted iteratively on small number of features they increase the likely hood of finding bugs early in the project in intermediate releases (sprints) and in turn reduces the likely hood of magnifying & propagating the bugs to the final project. Because of this fact the quality of software product is better in agile software development. During deployment the product increment is tested by the user which is known as user acceptance testing (UAT) to ensure that all the user stories specified in the sprint backlog are actually implemented. This testing is done manually. In addition to testing functional

DOI:10.32377/cvrjst0515

requirements, it is essential to test non-functional requirements. Some of the typical non-functional requirements are: load testing, security testing, and performance testing. Tools are used for testing non-functional requirements. These non-functional tests are executed at the end of the sprint. Some of the tools which aid in automation for different agile testing tasks are given in Table I. All the members of the development team should be able to use testing automation tools.

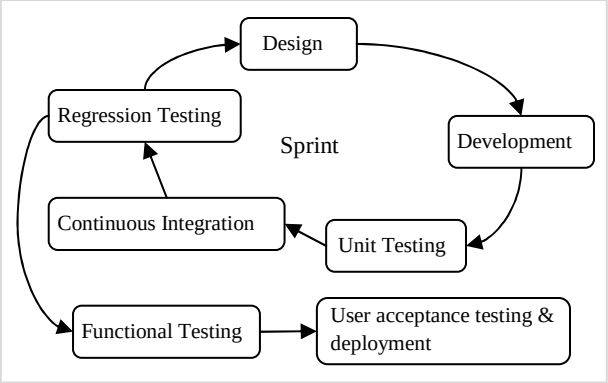

Figure 3. Proposed testing strategy for scrum

B. Automation

Software testing automation is key for the agile testing. Irrespective of agile methodology, testing automation becomes the core of agile testing [12]. The purpose of software testing automation is to automate software testing activities. Manual testing is time consuming. Manual testing is not suitable for scrum agile testing where continuous deployment is required in shorter intervals. Moreover, since testing tasks are conducted iteratively during a sprint, through testing automation testing time can be reduced considerably. Tools are available to automate all the testing activities. With automation testing efficiency can be improved and testing time can be reduced which enables to deploy the working product increments in shorter intervals. Some of the typical tools which are helpful for testing activities are given in Table I.

C. Advantages of Proposed Strategy

The proposed testing strategy in the context of proposed sprint activities offers following advantages.

- Strategy is simple.
- Refactoring overhead is not there.
- Simple design, which provides only implementation guidance.
- Development (coding) is based on design, hence less chances of errors.
- Automation is used in testing tasks, hence reduces testing time.
- Since review on testing is done daily, any mistakes can be rectified immediately.
- User feedback after every delivery helps to improve the testing activities.
- Reduces development time, improves productivity and product quality.

TABLE I.
TYPICAL AUTOMATION TOOLS FOR SCRUM AGILE SOFTWARE TESTING

Testing activity	Tools		
Unit Testing	JUnit	xUnit	MockRunner
Continuous Integration	Hudson	Fit, FitNesse	Green Pepper
Regression Testing	STAF	IBM Rational Functional Tester	VersionOne, FitNesse
Functional Testing	Selenium, Avignon	WinRunner, Cucumber	FitClipse
Non-Functional Testing	JMeter	Benerator	CLIF
Testing management	Testlink	IBM Rational Quality software	HP Quality Center

D. Case Study

Software development using scrum delivered a better quality product. This is due to the fact that only few features are added to every new increment and testing is done on few features only, which increased the chances of finding the bugs. In addition, after every iteration release, customer gave the feedback on bugs on those features, hence they could not be propagated to next iteration, otherwise, they would have got amplified and increased the number of bugs in the next iteration.

The proposed approach is applied on the real-time ETL tools which are being used by the customers. The Fig. 4 shows the ETL process.

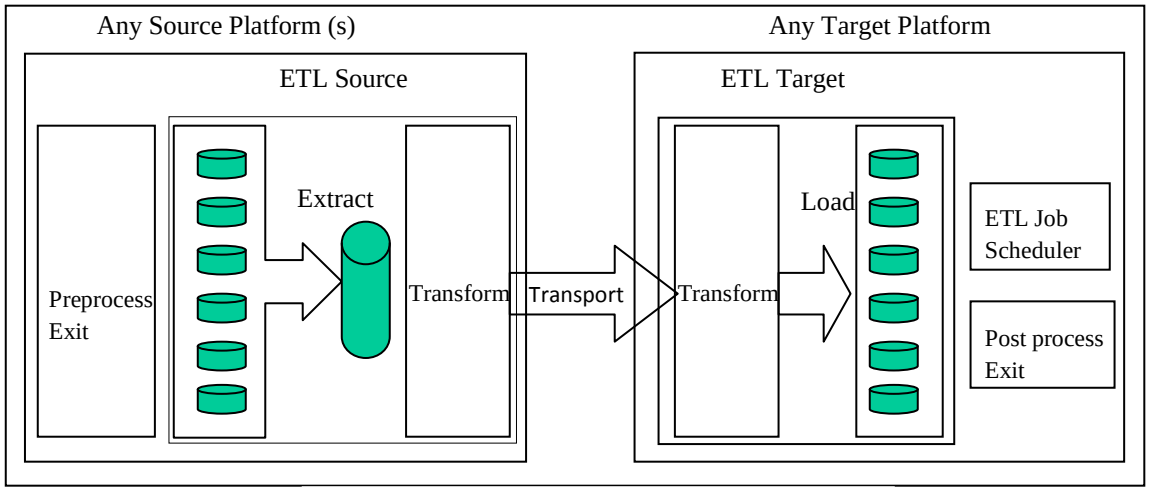

Figure 4. The ETL process

The ETL stands for "extract, transform and load", is the set of functions combined into one tool or solution that enables companies to "extract" data from numerous databases, applications and systems, "transform" it to appropriate format, and "load" it into another databases, a data mart or a data warehouse for analysis, or send it along to another operational system to support a business process.

The Fig.5 describes the number of bugs reported from the field in one year after the full product is released to the customer. Two ETL tools, DB2 ETL tool and Sybase ETL tool are developed using the scrum agile process using the proposed testing strategy. These two tools developed in the agile are compared with a similar ETL tool Teradata ETL tool which was developed in traditional development model and released to the customer. This study indicates that the number of defects received from the field for the products developed in the scrum agile process are reduced by around 50%, as compared to the products developed in the traditional model.

This reduction in the number of bugs reported from the field for the products developed in the agile process is due to two factors: first, there is a continuous feedback from the customer after every iteration in the agile development process. Second, in every iteration the testing of the features in the current iteration and regression testing of the features already delivered in the previous iterations will be done, which makes sure that the new features do not inject any regression issues. This indicates that the proposed scrum agile testing strategy reduces the number of bugs reported from field. Hence, the proposed agile testing strategy improves the quality of the products developed in the agile process.

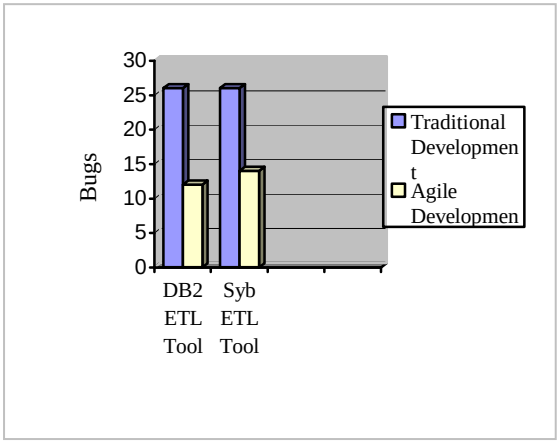

Figure 5. Traditional Vs Agile development

CONCLUSIONS

The software development in the industry is moving towards agile due to the advantages provided by the agile development process. Two main advantages of agile software development process are: delivering the high quality software to the customers in shorter intervals and having the capability of embracing the changes in requirements at any stage of software development. In majority of the situations scrum model is preferred because it delivers working software product increment in a predefined time-box (typically 30 days). Delivering a working product increment in shorter intervals (30 days) gives business advantage to the customers. Testing in agile process model plays a vital role. Testing strategies

DOI:10.32377/cvrjst0515

for traditional process models are well established, but these strategies are not directly applicable to agile testing without modifications and changes. A strategy for agile testing in the scrum software development environment is proposed and presented. The context (sprint activities) in which the proposed testing strategy to be applied is also proposed. Typical tools which are helpful for automation are given. The advantages of the proposed strategy are highlighted. Case study which used the proposed strategy is presented, it indicated that the number of bugs reported by customer reduced by 50% in scrum agile development when compared to traditional development. This indicates the improvement in product quality in scrum agile development using the proposed testing strategy.

FUTURE DIRECTIONS

More number of case studies from different domains and applications need to be studied to get further insight into the research areas of agile software testing strategies.

ACKNOWLEDGMENT

The authors are very much thankful to the CVRCE management and IBM India Software Lab (Hyderabad) for providing research environment and support for carrying out the research. Authors express sincere thanks to Principal and Dean (Research), CVRCE, for their encouragement to do research and publish papers.

REFERENCES

[1] L.Williams and A.Cockburn," Agile software development: it's about feedback and change", IEEE Computer 36(6), pp. 39-43, 2003.

[2] J.Highsmith and A.Cockburn, "Agile software development: The business of innovation", IEEE Computer, 34(9), pp-120-127, 2001.

[3] C.R.Jakobsen and J.Sutherland, "Scrum and CMMI going from good to great", Agile Conference (AGILE), pp. 333-337, 2009.

[4] L.Crispin and J.Gregory, Agile testing: A practical guide for testers and agile teams, Addison-Wesley, 2009.

[5] Kai Petersen and Claes Wohlin,"The effect of moving from a plan-driven to an incremental software development approach with agile practices: an industrial case study", Empirical Software Engineering, Vol.15, Issue 6, pp.654-693, 2010.

[6] B.Boehm and R.Turner, "Management challenges to implementing agile processes in traditional development organizations", IEEE Software, 22(5), pp.30-39, 2005.

[7] K.Schwaber and M.Beedle, Agile software development with scrum, Prentice Hall, 2002.

[8] K.Schwaber, Agile project management with scrum, Microsoft Press, 2004.

[9] VersionOne, *7th Annual State of Agile Development Survey*. Retrieved in November 2013 from http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf, 2013

[10] T.Chow and D.Cao, "A survey study of critical success factors in agile software projects", The Journal of Systems and Software, pp. 961-971, 2007.

[11] Theodore D. Hellmann, Abhishek Sharma, Jennifer Ferreira, and Frank Maurer, "Agile testing: past, present, and future – charting a systematic map of testing in agile software development", Agile Conference (AGILE), pp. 55-63, 2012.

[12] E.Collins, A.Dias-Neto, and V.F.de Lucena, "Strategies for agile software testing automation: An industrial experience", IEEE 36th Annual Computer Software and Applications Conference Workshops (COMPSACW), pp. 440-445, 2012.

[13] Puneet Agarwal,"Continuous SCRUM: Agile management of SAAS products", *ISEC' 11: Proceedings of the 4th India Software Engineering Conference*, February 2011.

[14] K.Lukasiewicz and J.Miler, "Improving agility and discipline of software development with the Scrum and CMMI", IET Software, pp. 416-422, 2012

[15] K.Schwaber and J.Sutherland,"The scrum guide: The definitive guide to scrum: The rules of the game", https://www/scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf#zoom=100, 2013.