

Need for Social Media Approach in Software Development

Vasuki Sundaramoorthy* and B. Bharathi

Department of CSE, Sathyabama University, Chennai – 600119, Tamil Nadu, India;
Js.vasuki@gmail.com, bharathivaradhu@gmail.com

Abstract

Objective: This paper discusses the need for social media approach in various software development processes to make it more collaborative which will result in increased productivity and better quality of software delivered. **Methods/Analysis:** To achieve more collaboration in software development, social interface methods are to be used. Usage of Social media methods like tagging, posting and commenting improves the communication among the team members. **Findings:** Software development typically is carried out by different teams working in different geographic locations and so the need for higher collaboration among teams is important for success of a project. Traditional ways of documenting the requirements, creating test cases, defect tracking have not been modified to leverage the enhanced connectivity that the current generation has. There is a need to create a new method to perform the software engineering tasks in a more interactive way. This will help in leveraging current social media techniques to implement the efficient practices from existing software development methodologies. **Applications/Improvements:** Collaborative platform can be used in software development, project management and also to evaluate the productivity and performance of a team member

Keywords: Collaborative Software Development, Software Development, SDLC using Social Media Techniques

1. Introduction

In today's scenario software development happens in multiple locations spread across the world and in multiple time zones¹. Be it developing applications for companies operating in Insurance, Banking, and Capital markets domain or working on graphics for a Hollywood movie, teams sit in different geographies and require greater collaboration to complete the project within the stipulated timeframe². When people from different organization cultures come together to execute a project, they do not have a common platform to learn complex methods, procedures and tools³. Best practices of software development cannot be implemented if it is perceived as complex and cumbersome by the team involved in distributed software development⁴.

Companies have adopted different software development models based on the nature of projects they do as well as the expertise they have in a specific model. This certainly restricts their ability to leverage collaborative

ways of working. Existing software development methodologies like Waterfall, Agile, V model, Spiral, RAD etc., do not have any usage of Social media techniques which are prevailing in today's world. Hence there is a compelling need for a new methodology to take advantage of Social media techniques, so that software development is more collaborative in nature⁵. There are tools like IBM Jazz, Syde⁶, and Social CDE⁷ which have been developed to facilitate more collaboration in software development. There also have been experiments done using social media for software development in student projects⁸, study and techniques for timely and efficient recommendations for facilitating coordination between developers⁹ and the effect of socio-Technical congruence on software development process¹⁰. Research is being done on studying the significance of coordination in success of software development projects and options to make them more efficient. In this paper we will take a look at how certain software development activities happen in the current state and what would be the benefits of using a social media like

*Author for correspondence

interface in performing them.

2. SDLC Methodologies

SDLC, Software Development Life Cycle is a process used by software industry to design, develop and test high quality software. Objective of an established SDLC is to produce software that meets or exceeds customer requirements, reaches completion within estimated effort and cost.

- SDLC is a framework defining tasks performed at each step in the software development process.
- SDLC can be customized to suit the dynamics of the teams that are working in the project.

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. Phases involved in SDLC are as shown in Figure 1.



Figure 1. Phases in SDLC.

Some Popular software development life cycle methodologies are as below¹¹

- Waterfall
- Prototype Model
- Agile
- RAD
- Iterative Model
- Incremental Model
- V Model
- Spiral

Waterfall methodology evolved more than 50 years ago, but the usage has deteriorated over the past two decades. The latest methodology that was defined is Agile which has its roots from the Agile Manifesto established in February 2001. There has been no new software development methodology in the past decade which has been widely accepted by development teams across the world. All these models have been established prior to the current social media era which has changed the way we interact with others.

Even though the models differ in the methods software development is done, the core activities of software development remain the same across all these different methodologies (Figure 2). All these models will have Requirements management, Design, Coding, Unit, System testing and user acceptance testing performed, but each will vary in the way these activities are performed or in the timeline in which each of these is performed¹². For example, requirements are gathered in the form of User Stories for Agile, whereas the requirements are captured as text in predefined templates for waterfall model.

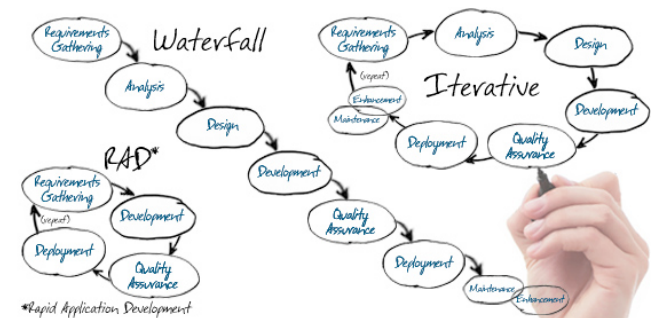


Figure 2. Sequence of phases in different SDLC methodologies¹³.

As per Gartner research there is a need to create a digital workplace that enables new and effective ways of working. Gartner states that need of the hour are a strategic rethinking of how people engage and how tools help employees get their work done. Report from Gartner further suggests that by 2018, 50% of team coordination and communication will occur via mobile group collaboration apps. Also the research states that more organizations will lose their market position due to “digital business incompetence”. Reason being lack of holistic response to how consumerization trends change how work is best accomplished. Conclusion of the research from Gartner is that it is high time to design an engagement focused digital workplace that empowers high impact performers and unleash their creativity.

Let us consider the activity of maintaining traceability matrix, requirements management, configuration management and knowledge management in its current state irrespective of the methodology used and the possible benefits a new methodology based on social media techniques will provide.

3. Requirement Management

Requirement management is one of the critical aspects of the software development. Requirement gathering is done by the business analysts in the team. Business analysts interact with the customer team and document the requirements in a software requirement specification or Business requirement documents¹⁴. If the approach followed is agile, then requirements are gathered as user stories. Unified Modeling Language (UML) which helps in visualizing and documenting is also widely used to capture software requirements¹⁵. Business analysts have multiple discussions with the end user or business team to refine the requirements. Both the functional requirements which deals with the functionalities of the system and nonfunctional requirements which deals with the other features of the system like accessibility, usability, security, scalability, performance, robustness, availability.

Some of the issues faced in the current scenario are

- Incorrect version of requirement document used by the development teams
- Since the documents run into multiple pages, business users do not actively review the requirements. This results in changes to requirement documents at a later stage of development.
- Nonfunctional requirements gathered in the documents are often overlooked by the developers as they concentrate more on development of components related to functional requirements

By using social media like interface, each of the requirements will be posted similar to a user status being posted in social media. This will solve the incorrect version of the document being used by the development team as they will always have the latest version of the requirements available to them¹⁶. Also whenever there is a requirement change, the development and testing team can be notified using the notifications techniques that are so familiar. For the business users, it will be easy to review the requirements one at a time and provide their sign off

on each individual requirement. This will help in a thorough review of the requirements and thereby minimal rework at a later point of development. Nonfunctional requirements can be captured with a specific tag related to them so that the development team takes adequate measures to consider the nonfunctional requirements during the design stage¹⁷.

One advantage of having a social media interface is that the development and testing team need not wait for the entire requirement document to be completed before starting their task. Requirements which are reviewed and accepted by the business team are readily available to the development team¹⁸. Another advantage is that if implementation of a certain requirement is not technically feasible, development team can point out the difficulties of implementation during the requirement gathering and can provide the business team with alternative approaches. This will alleviate concerns of the end users that the alternative approach is not discussed with them upfront and they are surprised to know that the features they have requested is technically not feasible to implement.

4. Traceability Matrix

By definition, a traceability matrix is a document, usually in the form of a table that correlates any two baselined documents that require a many-to-many relationship to determine the completeness of the relationship. A well maintained traceability matrix helps us to ensure all the components are tested thoroughly resulting in increased code Quality. Creating and maintaining a traceability matrix is a critical activity that is required to ensure all requirements are coded and tested and the software is defect free. In current environment, more and more companies are not maintaining traceability matrix, as they believe it is an additional step and skip to adhere to deadlines. Some do it at the start with base version of requirements, but fail to update it when the requirements change continuously. A sample traceability matrix between use cases and Test cases is as shown in Figure 3.

Some of the requirement management and test management tools provide options to maintain traceability matrix. However in some scenarios testing tools are being created for to support testing in windows phones¹⁹. These tools are created with specialized knowledge and built for a specific purpose. Whenever the tool does not support traceability by interacting with the business requirement tool, it is the responsibility of the tester to ensure

Reqs IDs	Reqs Tested	UC 1.1	UC 1.2	UC 1.3	UC 2.1	UC 2.2	UC 2.3.1	UC 2.3.2	UC 2.3.3	UC 2.4	UC 3.1	UC 3.2	TECH 1.1	TECH 1.2	TECH 1.3
Test Cases	231	3	2	3	1	1	2	1	1	1	2	3	1	1	1
1.1.1	1	x													
1.1.2	2		x	x											
1.1.3	2	x											x		
1.1.4	1			x											
1.1.5	2	x												x	
1.2.1	1		x												
1.2.2	1			x											
1.2.3	2				x		x								
1.2.4	2					x		x							
1.2.5	2								x	x					

Figure 3. Sample Traceability Matrix.

traceability by downloading the requirements in excel spreadsheet from requirements management tool and manually update the test case numbers against the same. The whole process is cumbersome and requires a lot of manual intervention. This results in failing to write new test cases or modify related test cases, whenever there is a requirement change. Cost of fixing a defect after the code has been deployed in live environment is very high compared to identifying and fixing it during the development stage.

Instead of traditional tools for requirements management, test management, if we adopt a social technique, requirements and its corresponding test cases can be linked with hashtags. A hashtag is a type of label or metadata tag used on social network and micro blogging services which makes it easier for users to find messages with a specific theme or content²⁰. Users create and use hashtags by placing the hash character (or number sign) # in front of a word or unspaced phrase, either in the main text of a message or at the end. Searching for that hashtag will then present each message that has been tagged with it. Test cases which are linked to the requirement that has undergone change, can be easily identified and updated by searching for the hash tag related to the requirement that has changed. Since requirements, test cases and defect logs are all maintained in the same place, this results in avoiding buying separate products for requirement management and test management.

5. Configuration Management

As per ITIL V3 configuration management is defined as the process responsible for maintaining information about Configuration items required to deliver IT Service, including their relationships. This information is managed throughout the lifecycle of the Configuration item.

Configuration Management is part of an overall Service asset and Configuration Management process.

Companies use products like PVCS, TFS to perform configuration management for maintain versions of project related documents and source code. All these products are based on the concept of Check in and Check out for maintaining the integrity of the configuration item. When a base version of a document is created, it is checked in the system. Whenever further changes needs to be made to the document, users Check out the document from the configuration management tool. Once a document is checked out by a user, it becomes read only for others preventing any changes that can be done to the version by others. This process of Check in and Checkout repeats till all the changes to the documents are completed and the final version of document is ready.

The current mechanism for configuration management still places the emphasis on the end user to ensure that the correct version is updated in the system. For example a user can check out a document and can check in next version of the document which can contain incomplete statements. Only when the document is reviewed the team will know that the current version of the document is incomplete.

Since software development will be done in an environment similar to social media, there will be no separate tools for storing various versions of the documents. Software requirements, Test Cases and Defect Logs will be in a format that ensures only the latest change is available to the person who consumes the information. Also edit access for the respective posts made will be with the same individual and hence the chance of the same being updated by some other user is negated.

6. Knowledge Management

Knowledge management is increasing becoming a key aspect for today's software companies. Utilizing the best practices based on a similar work carried out earlier or taking an alternative design approach based on the lessons learnt from similar implementations, save significant time and effort for the development team. Knowledge management deals with capturing the knowledge gained by an individual or a team in a systematic fashion so that it can be retrieved by other teams looking for the information²¹. Most of the major software companies in the world use portals hosted in intranet for sharing knowledge within their company. Upon completion of any

project, development team's focus will be either on fixing the issues post deployment or working on inception phase for a new project. These result in little or no time for creating documentation of best practices followed and lessons learnt in any successful project.

Capturing knowledge gained will not require a specific intranet portal as it is achieved implicitly by using a social media like interface²². By using specific tags for best practices and lessons learnt from each project, it will be easier for other development teams to access this information. Also this will reduce the time required for creating lengthy documentation for this purpose, which most of the developers are not keen to create. The other advantage is that this provides an opportunity to search for information based on the people involved in a specific implementation.

7. Other Benefits

Traditional Software development has been driven by various tools which are usually built for specific purposes. For example, HP Quality Center for testing and defect management, IBM Rational DOORS for requirement gathering. While each of these products has been individually brilliant, it still requires some level of integration to get data that would reflect the current status of the project and work items.

Some of the products in this space, have option to integrate with very few popular products in this space. For example, integration of IBM Rational Doors with HP QualityCenter may be more easily done, compared to integration with open source test management products like Bugzilla. This creates a necessity that a company has to go with specific pair of products to reap the benefits out of the integration. Most of the tools are provided through the licensing option with yearly renewal. Now with the software development team, located across the globe, there is a need for Global licensing options which are costly.

Also involved is the cost involved in training users in these products. So whenever an employee who has been using these products, has to be replaced there is a training effort and cost spent on the new employee. Further there is a need for qualified administrators for maintaining these products. So the default option is to go for support with the product owners and this will lead to additional costs.

More over current products are not highly collaborative. Most of the products are still installed in a physical machine such as desktops/Laptops or virtual machine, and this would require users to get connected to the company's network, before logging in to these products. So if all it requires is a single click to sign off a requirement or to update a defect status, the user still has to go through a lot of steps before connecting to the application.

By using a social media like interface to handle software development tasks, we can avoid the need to buy specific products for requirement management and test management. This will result in significant reduction of cost related to procuring licenses, support cost for the products, costs incurred for training project team members on the same. Since the interface will be simple, it will be easy to learn and also it will increase the communication among the team members. Also there is an option of hosting the proposed solution in cloud, which will result in improved accessibility.

8. Conclusion

While there are attempts being made to ensure more collaborative work environment, we have to relook at entire software development life cycle to redesign the different tasks performed so that it can be done from anytime, anywhere using the various social media techniques that have evolved over the past decade. This will definitely lead to increased productivity of the team and a better collaboration in delivering a quality product within the specified timeline.

9. References

1. Nguyen T, Wolf T, Damian D. Global Software Development and Delay: Does Distance Still Matter? Proceedings of 3rd Intl Conference Global Software Engineering, India. 2008 Aug. p. 45–54.
2. Dullemond K, van Gameren B. What distributed software teams need to know and when: an empirical study. Proceedings of 8th IEEE Int Conf on Global Software Engineering, Italy. 2013 Aug. p. 61–70.
3. Dullemond K, van Solingen R. Increasing Awareness in Distributed Software Teams: a First Evaluation. Proceedings of 9th Int'l Conf Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 13), USA. 2013 Oct. p. 325–34.

4. Derntl M, Renzel D, Nicolaescu P, Koren I, Klamma R. Distributed Software Engineering in Collaborative research projects. Proceedings of 10th International Conference on Global Software Engineering (ICGSE 2015), Spain. 2015. p. 105–9.
5. Begel A, DeLine R, Zimmermann T. Social Media for Software Engineering 2. Proceedings of FSE/SDP workshop on Future of software engineering research, FoSER '10, USA. 2010 Nov; 33–8.
6. Hattori L, Lanza M. Syde: A Tool for Collaborative Software Development. Proceedings of 32nd Int'l Conf Software Eng (ICSE 2010), South Africa. 2010 Mar . p. 235–8.
7. Calefato F, Lanubile F. SocialCDE: A Social Awareness Tool for Global Software Teams. Proceedings of 9th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 13), USA. 2013; 587–90.
8. Juarez-Ramirez R, Pimienta-Romo R, Ocegueda-Miramontes V. Supporting the software development process using social media: experiences with student projects. Published in 2013 IEEE 37TH Annual Computer Software and Application Conference workshops, Japan. 2013 Jul . p. 656–61.
9. Blincoe K, Valetto G, Damian D. Facilitating Coordination between Software Developers: A study and Techniques for Timely and Efficient Recommendations. IEEE Transactions on Software Engineering. 2015 Oct; 41(10):969–85.
10. Kwan I, Schroter A, Damian D. Does Socio- Technical Congruence have an Effect on Software Build Success? A Study of Coordination in a Software Project. IEEE Transactions on Software Engineering. 2011 Jun; 37(3):307–24.
11. Pressman R, Maxim B. Software Engineering: A Practitioner's Approach. 8th Edition. WCB/McGraw-Hill: Columbus, 2015 Dec.
12. Kerzner HR. Project Management 2.0.1st Edition. John Wiley & Sons Inc: Hoboken, 2015 Jan.
13. Article title. Available from: <http://cloverleafsolutions.com/methods/software-development-lifecycle.html> . Date accessed: 05/11/2015.
14. Carillo de Gea JM, Nicolas J, Fernandez Aleman JL, Toval A, Ebert C, Vizcaino A. Requirements Engineering Tools. IEEE Software. 2011 Jul; 28(4):86–91.
15. Bharathi B, Kulanthaivel G. A Simple Method for Deriving LQN-models from Software-models Represented as UML Diagrams. Indian Journal of Science and Technology. 2012 Feb; 5(2):1–7.
16. Treude C , Storey MA. Awareness 2.0: Staying Aware of Projects, Developers and Tasks Using Dashboards and Feeds. Proceedings of 32nd Int'l Conf. Software Eng., New York. 2010 May; 1:365–74.
17. Treude C, Storey MA. Bridging Lightweight and Heavyweight Task Organization: The Role of Tags in Adopting New Task Categories. Proceedings of 32nd Int'l Conf Software Eng. 2010 May. p. 231–4.
18. Renzel D, Behrendt M, Klamma R, Jarke M. Requirements Bazaar: Social Requirements Engineering for Community-Driven Innovation. Proceedings of 21st IEEE Int Requirements Engineering Conference (RE), Brazil. 2013 Jul . p. 326–7.
19. Julian Menezes R, Albert Mayan J, Breezely George M. Development of a Functionality Testing Tool for Windows Phones. Indian Journal of Science and Technology. 2015 Sep; 8(22):1–7.
20. Treude C, Storey MA. How Tagging Helps Bridge the Gap between Social and Technical Aspects in Software Development. Proceedings of 31st Int'l Conf Software Eng, Canada. 2009 May. p. 12–22.
21. Vigneshwari S, Aramudhan M. Social Information Retrieval Based on Semantic Annotation and Hashing upon the Multiple Ontologies. Indian Journal of Science and Technology. 2015 Jan; 8(2):103–7.
22. Phosaard S, Wiriyapinit M. Knowledge Management via Facebook: Building a Framework for Knowledge Management on a Social Network by Aligning Business, IT and Knowledge Management. Proceedings of the World Congress on Engineering 2011, London, U.K. 2011 Jul; 1–6.