Volume 4, Issue 4, April 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Document Driven Approach for Agile Software Development

Vishvadeep Tripathi, Arvind Kumar Goyal

School of Computer Science and IT,

Devi Ahilya University, Indore, India

Abstract— some voice has been raised for less use of documentation in the projects which are developed using agile methodology. Lack of documentation creates a lot of problem in maintenance of the product. In this paper we have identified the document selection factors and proposed documentation approach for agile development. We have suggested new techno-functional role that will create required documentation which will be useful in the transition of the software system from development to maintenance

Keywords— document, agile documentation, documentation, Techno functional expert, Team Collaboration, document selection;

I. INTRODUCTION

In all the software engineering practices documentation has its own importance. It is a medium of communication. Documentation is oldest recommended practice in software development and still has its own importance. In traditional methodologies usually requirement priority is specified only once. In agile methodology requirement priority has been specified in the beginning of every development cycle. While doing requirements prioritization in traditional Methodology business values, risk, cost and decencies are the factors which are considered , business value, changes to existing functionality, bug fixes and refactoring are considered while doing prioritization in agile methodology.

Agile methodology practice believes on working software rather than comprehensive documentation. *Assumption is documents are much smaller than traditional approaches*. Lack of documentation might create long term problem for agile teams. Documentations are used for sharing the knowledge between people. Documentation reduces knowledge loss whenever any key team member becomes unavailable.

In Agile methodology focus is given on the coding and giving the result to client in short span of time. Agilest believes that documentation should be brief and precise. It should contain the information about the core part of the system. It may contain only source code and user stories, other information communicated in short meeting and not recorded anywhere. It might be possible that there might not be any other document like specification, design document, process document maintained as a part of agile process.

Agilest have never mentioned that how much document is required but they mentioned that only enough document is required. They have not mentioned that what all document will be enough. They have not given importance to document rather given importance to working software. Agilest believe that quality of the system can be achieved by the pair programming, continuous integration, automatic testing and refactoring rather than documentation.

Some time it seems that agile methodology is good for small development having short term perspective. By reducing documentation production cost can be decreased and system can be in production early but in long term during the maintenance phase it can create problem.

Mainly documentation becomes more important once product gets deployed to production and transitioned to maintenance. Future maintenance of any product depends on the product's documentation, lack of documentation for business requirement, system requirement, and technical design can create potential problems during transition to maintenance.

For large sized and complex project documentation play a very important role. Such project needs user requirement documents, architecture documents, design decision documents , source code and source code review document , Risk and issue management document ,user guide document, test plan document. Documentation used to provide necessary information available to the team members.

Projects which are running using traditional methodology contain so many documents and Team members looking for specific information can easily get lost. Second problem is documents getting out dated as project progresses, this require stringent process in place.

In agile only limited document created and agile documentation will be effective only when it is light weight and that it does not contain unnecessary documentation. It will provide all the information which is relevant to readers. Agile methodology documents should be of high quality, accurate, up-to-date, highly readable, and concise and well-structured.

The remaining of this paper is organized as follows. Section 2 contains literature review. Section 3 and 4 presents the finding of this paper in the form of document selection process and document requirement for agile development. Finally section 4 concludes the papers

II. LITERATURE REVIEW

Literature in the field of documentation requirement in agile methodology contains a large body of research work. However, we focused this literature survey on identification of documentation selection factors and minimum documentation requirement in any software project development.

Forward and Lightbridge provided the observation that documents contents will be relevant even if it is not up to date however keeping document up to date is a good practice. Documentation is a tool of communication and latest technologies should be used to create it so that validation and verification of the document will be easy [1].

Juyun presented five issues and challenges which include documentation, communication, user involvement, working environment, and Scrum ceremonies. He has mentioned that if above mentioned issues and challenges are addressed and resolved before starting any agile project, organization will have fewer difficulties in delivering high quality software product using Scrum [2].

Stettina and Heijstek presented their work they have specified that documentation alone is insufficient. So combination of the documentation and face to face communication is required for producing high quality software using agile methodology [3].

Stettina and Kroon presented their study and mentioned that during the handover process of agile project to maintenance, three independent phases namely Environment, System and Architecture need to be taken care are having impact on handover process. They have suggested documentation artefacts should support the learning process and it should include maintenance staff in the development process that will significantly help improving the handover process from development to maintenance [4].

Souza et al done the surveys of the agile projects and they have named documents which are important and have to be present to maintain the software which has been moved to production and going to be maintain by maintenance team. They have confirmed that source code and comments present in it are one of the important artefacts to understand and maintain the software systems. Data model and requirement specifications which are present in form of user stories or use cases in case of agile are also very important documents. These documents are very useful when we are in maintenance phase of system life cycle. [5].

Mattson presented the details of the performed study and specified maintenance trenches on the lifecycle problems related to system and process. Author found that main problem of agile process was engineer did not pay enough attention to create documentation. According to the authors study documentation is one of the very important factors for sustainable organization. According to him still agile practitioners are confused on what all documents need to be created as a part of agile development which will be helpful to support the product [6].

Rubin Eran and Rubin Hillel presented the relationship between software and documentation. They suggested approach in which one should incorporate domain documentation in agile development keeping the process adoptive. Suggestion of including the system design also has been given which use domain documentation [7].

Sillitti et al. presented their study and they have listed similarities and differences between agile and documentation driven approach. Agile companies believe on tight interaction with customers from very beginning of the project it increase very satisfactory relationship with the customer. Agile methodologies are more customer centric and flexible then document driven traditional methodologies [8].

Hoda et al. presented their study and tried to find out how much documentation is enough in agile software development. They have described some patterns which can be helpful for software development team to define what all documents will be enough in their own context. They suggested agile teams can also use these document patterns to overcome common challenges of adopting agile methods [9].

Stettina et al. investigated between two different documentation practices and agile development. They found that writing documentation for agile team needs task specialization and if it has been done by the less experience professional then it can create problems. They have suggested that if documentation is going to be delivered to the other teams as a part of the package delivery of the software product, producing document should be communicated well in advance and accepted by the team as proper product [10].

Hoda et al. presented documentation strategies and practical example of them which can be used to overcome various challenges which can be faced on day to day implementation of agile methodology. They have presented strategies which include 1) documenting electronic back-ups of physical paper artefacts 2) documenting changes decisions made by customers 3) documenting Business terminology into a project dictionary for more effective requirements elicitation 4) Documenting the traditional way when collaborating with non-agile teams 5) Documenting positive customer feedback [11].

Uikey et al. presented frame work which represent conceptual view of documentation and they have introduced technical writer along with other scrum roles. Also they have presented the relationship of documentation and technical writer which we can help to enhance the productivity and maintainability of software [12].

III. IDENTIFIED DOCUMENTATION SELECTION FACTORS

We perform literature survey and did interaction with agile practitioners and indentified document selection factors while selecting documentation for any agile project. Based on our extensive literature survey we identified factors which need to be taken care while selecting documents for any agile project. In Agile methodology importance has been given to working software instead of comprehensive documentation. In agile methodology frequent customer involvement required, customer needs to be present with the team. Whenever team needs any help from customer, ask any questions

response should come from the customer immediately or as early as possible. Most of the communication happens in the form of oral communication.

A software document may be described as an artefact intended to communicate information about the software system if documentation is correct, complete and consistence then it will be power full tool for developers to gain success.

According to traditional approaches, software development processes should incorporate three iterative phases: Analysis, Design, and Implementation. The key difference between agile approaches and traditional approaches is that traditional approaches explicitly document the knowledge which has been gathered during different phases of the life cycle but agile approaches tend to refer to source code as the only documenting artefact.

Studies has been proved that following purpose need to be full fill by the documents which are produced in software development life cycle.

- Documents should be able to provide work transparency and interpersonal/intra team communication
- Documentation should have required information which will be helpful for future maintenance of the software system.
- Higher management wants information of time and budget they wants status of the project. If they need estimation, budget and status details at any part of the project with documents we should be able to provide them on time.
- After delivery of the project/product there will be users who are going to use the system and higher management should have information which they can use for future projects with challenges and learning from the projects.
- Documentation should help to gain the knowledge of the product if any key team member has been moved from the project.
- Documents should give enough knowledge to new developer or new user of the system.

In agile methodology most of the communication is done informally. Oral communication takes times and it distracts from the original questions some time. Some time it is very difficult to find person who has correct answer. Many discussions happen informal and there is no documentation created for that and engineer forgets the decision and reason as time progress. As requirement changes very frequently and as per agile methodology team has to accept the changes and implement it. Engineers do not remember why they did the changes after some time and who has told to implement it. As there were no system flow document present engineers are not certain of what changes they have made and where. If any new developer joins he asks basic questions because of lack of system knowledge. Developers do same mistake repeatedly. If any old developer leaves the team then valuable knowledge of the system will be loss because of lack of documentation. It might be possible that all the developers will not have whole system knowledge and only few old and experience engineer will have full system knowledge. As developers not have full knowledge of the system and they do the development without knowing full system Architecture deteriorates substantially. As there is no documentation present it may be possible that product and process

Information is not understood properly which can create problem in future as system documentation not created. At the time of scaling up the software system it will be very difficult for team to coordinate. Systems have become more difficult to change and understand. If communication is not proper and in written form it may be possible that testers will misunderstood the functionality and testing will not be done as thorough as it should be done. Unit testing need to be done in proper manner and automated. It is important to cover at least 90 percent code with the automated unit test cases.

In the beginning of the software development management and key team members of the team has meeting and decides what all documentation they are going to have as a part of the system development. Decision to have a particular document should be taken on the basis of following criteria.

Product Type: type of product is one of very important factor from which management decide what all documents need to be created. For example if type of product is complex, big and required so much business knowledge then functional requirement and because of complexity non-functional requirement also need to be maintained.

Project Size: if project is big then more detail documentation will be required as big team is going be needed to complete it. So much communication between different team will be needed and it needs documents. Proper authorization testing will also need. If size is good then budget and the scheduled

Time to implement the project also increases. And for the budget and the schedule, we need to have the status reporting in proper manager and generate reports on a regular basis which will be share to higher management. If project is large we need to have process document as well.

Product Environment: Both the environment in which product is going to be deployed and in which project development is going on. If teams are sitting in different places then environment is more important. If product is complex and going to be deployed in the highly scalable and highly available system then documents need to be maintained.

Team experience: experience of the team is very important if team is having experience in the development of the agile then it will be easy to do the implementation but if team is new then it can create a lot of problems. For the unexperienced team there is need to have a proper record that could help understands the weak and the strong points of the project.

Customer requirement: customer requirement decide the importance of the documentation. We are talking above over all requirement of the system. If customer is changing their requirement so frequently the proper record of the

documentation is necessary and it should be maintained. You should also know that what all documents customer wants as a product delivery.

User's type and level: as number and type of user of product is going to increase more documentation is going to required. Users are divided into groups and different group is going to use different part of the software product. User wants to have to proper document which help them to use the product and maintain the product.

IV. MINIMUM REQUIRED DOCUMENTS

We have identified following documents need to be created and marinated in the complex software development. These documents are useful for the understanding the system and can be used for the system maintenance.

Contract Document: Contract is an agreement between two parties. This document can have physical copy as well and enforceable by law. Contract document used to identify existing of the project. It has proper sign of both the party's customer and higher management of the team (company) who is going to implement it.

Vision Document: This document is created by the higher management at the very beginning of the project. It is useful for the senior management of customer and the company who is going to work on the product and project managers. This document consists of vision of the system and current estimated cost, benefit of the product, schedule of the delivery and key milestones.

Project Overview Document: this document should have basic information of the all the aspect of the product. It should have information about need of the system, which will be primary contact of for the information, what all technologies are going to be used for the development, what will be the operating process of the system.

Process Documents: Higher management needs the documents which have the information about the plan, estimates and schedules, milestone of the system. Spring plan are created in the part of the sprint planning. Overall plan created for the release, daily status report needs to be created for the higher management, burn down, burn up chart created and which shows the progress of the system. Documents should contain risk and issues these documents need to be present and are called non-technical documentation.

Requirement Documents: Requirement's document defines what the system does. It consists of artefacts such as business rule definitions, use cases and user stories or important user interface prototypes.

> **User stories/Use cases**: in agile development some customers are providing user stories some provides use cases in some project it comes hybrid of user stories and use cases. User stories are written from the end user's prospective it has very simple language which user normally use and they focus on the benefit or result that they get, results can be treated as acceptance criteria. Use cases used to describe interaction between the system and actors who are going to use the system. Use cases have the detail information.
>
> **Non-functional requirement:** Customers usually focus on core business functionality and ignore NFR's such as scalability, portability, maintainability, performance etc. customer gives focus on easy to use requirements and tendency to ignore critical requirement like security and performance in early stage of the project results in major issue as system become more mature and ready to deployment. These issues need to be taken care of early stage of the requirement.
>
> **Domain Knowledge**: It should be created and constantly available for reference. It can be used to validate requirements explicitly in some source code the way updated requirements are embedded in code should be shared between developers. Software development should follow users' view as requirement's comes in form of user stories which has user's view rather than system view. Domain knowledge document should consist all the business terminology and it should be available for each and every team member which will help the new developer's to understand the business and reduce the defect density

System flow and Design document: System flow document used to provide an overview of the system and helps the team members understand the system. *This document is for the development developers and maintenance* Developers. The document includes technical architecture showing the technical details of the system, like how it's working and what all its components, etc. Technical and business architecture shows that the system documentation should at least include high level requirements for the system. Developers should create low level design which includes flow of the module. They should create class level diagram and relationship with the classes. It also includes sequence diagram. Now days we have software in place with that these documents can be created. Review of these documents should be done by the techno functional expert. Design document should also include data models it should have physical as well as logical data model in place which will help in maintenance or new developer's to understand the system.

Support Documentation: Support document is used to support staff after the software has been developed and deployed to production. This type of documentation may include support guide, user guide and manual guide. These documents can be only created by working closely with the developers in any story development or design meeting. It should be responsibility of the techno functional expert to create it and update is as long as system evolves. This document should have trouble shooting steps for common problem which can occur in the system.

Test plan documents: Organizations normally use tests cases to capture complete requirement. Test cases should be linked with the requirement and requirement should link to design document and design document should be linked to production code. Traceability needs to maintain. If you have linking it is very easy to find out code if anything goes wrong into production. Even if in development you get very quick feedback if code is not giving you expected result.

Agile practitioners believe on the working software and software should be delivered to the customer every sprint. Unit testing plays an important role to make sure the quality software intact. Agile users suggest automation of the unit testing

need to be done and it should cover 90 percent of the code. Whenever any code is going to be check-in into the repository build process need to be executed and full round of testing needs to be done. It helps to find out the problem in the early phase itself.

For integration testing and functional testing as well test automation is preferred practice in agile development. These scripts should be written and cover all functional aspect of the system. it needs experience. We suggest techno functional expert should review these documents and make sure that automated testing has covered the entire scenario.

Lessons learnt document: It should contains all the challenges which has been faced in the process of agile development. It should have both types of challenges management challenges as well as technical challenges. After every sprint retrospective meeting needs to be conducted and all the issues and challenges has to be incorporated in it. Positive Customer feedback needs to capture in the lesson learnt document. The technical functional expert document the experiences and impediments observed in scrum process, which can be used as reference in future projects.

Document management system and version control system: In the agile methodology document management system plays a strong role. As we all know agile methodology prefer to the project for which requirements are uncertain and changing very frequently. Requirement tracking needs to be done and it has been suggested that software tools should be used for that. For example JIRA is one of the tools which can be used for this purpose. In agile so much communication done in informal manner which is present in the form of past it , white board writing etc. This information need to be capture via digital camera and should be stored so that in future requirement tractability can be done.

Role of techno functional expert: this person should play key role and should be involve in the process from the contract time. Techno-functional expert should be involved in the process of determining the required documents of the system. Participation in the technical discussion, discussion with the higher management, review process should be done by the expert. There is a big gap between the knowledge and thinking of business person and technical person. Agile mentioned that customer and team should directly collaborate and technical person can do the development on the basis of information given by the customer. Sometime developers understanding are different because of less business knowledge and experience in that domain. In practical scenario it is not possible that for each and every small business clarification customer is available. Agile practitioners has given the complain that even if agile methodology give emphasis on customer involvement should be high but they are not getting customer availability on time . It delays their development and developer's slog for nights to complete the development on time. It has been observed that if we have big and complex project which has complex architecture then developers are getting difficulties to convert business requirement to technical form. In this case architecture of the product deteriorates.

We suggest there must be one techno-functional expert role in the agile development project. This expert should good amount of business knowledge and having expertise in the technical architecture. Expert should be able to see overall picture of the software which include business knowledge, architecture knowledge and design knowledge of the system.

Techno-functional expert should be responsible for following tasks.

1) Providing support to development on converting business requirement into technical form.
2) Providing support to business and explain them usability of the particular requirement
3) Give suggestion to business and educate them as per the technical feasibility of the requirement
4) Creating user documents which will be used by the user after delivery of the product
5) Making sure that architecture documents are up to date and having all the relevant information
6) Making sure that technical designs are up to date.
7) Making sure that code has been written as per the business requirement and full fill the design need
8) Making sure that code review check-list has all the points which should be taken care to full fill the design and architecture needs
9) Make sure that development has been done as per the architecture proposed by system and uniformity of the development has been maintained.
10) Making sure that developers are writing proper comments in the code so that code will become maintainable.

Good communication skills and able to establish good relationship are key point which techno-functional person required. Techno functional expert should be involve in every step of the software product development. From the initial planning meetings, through the stages of programming, and on to final packaging, techno-functional person will generate and review the documentations. Involvement in each and every phase needed for comprehensive coverage of each and every facet of product. It removes the risk of accidentally leaving out something important when documentation has details of full product and delivered as a part of complete software package.

V. CONCLUSIONS

In this paper we presented result of our study, we performed detailed literature survey and then interacted with the experienced agile practitioners having exposure of working in big complex application development which needs rich knowledge of domain. To conclude we have identified document selection factors which can be used before selecting any document for agile development. We have given the name and purpose of the document which are very necessary for successful and high quality delivery of the software product. These documents will play a vital role in the transition of the product from maintenance to development.

REFERENCES

[1] Forward A. and Lethbridg T. C., "The relevance of software documentation, tools and technologies: a survey," In Proceedings of the 2002 ACM symposium on Document engineering, pp. 26-33.

[2] J, Cho, "Issues and Challenges of agile software development with SCRUM." Issues in Information Systems Vol. 9, No. 2, 2008, pp. 188-195.

[3] Stettina C. J. and Werner H., "Necessary and neglected?: an empirical study of internal documentation in agile software development teams," In Proceedings of the 29th ACM international conference on Design of communication, pp. 159-166.

[4] Stettina C. J. and Kroon E. ,"Is There an Agile Handover? An Empirical Study of Documentation and Project Handover Practices across Agile Software Teams," In 19th ICE & IEEEITMC International Conference, The Hague, Netherlands.

[5] de Souza S.C.B , Nicolas A. and de Oliveira K.M, "A study of the documentation essential to software maintenance," In Proceedings of the 23rd annual international conference on Design of communication: documenting & designing for pervasive information, pp. 68-75. ACM, 2005.

[6] K.M. Mira, "Problems in agile trenches," In Proceedings of the Second ACM-IEEE international symposium on Empirical Software Engineering and Measurement, 2008, pp. 111-119.

[7] Rubin E. and Rubin H., "Supporting agile software development through active documentation," Requirements Engineering Vol. 16, no. 2, 2011, pp. 117-132.

[8] Sillitti A et al. ,"Managing uncertainty in requirements: a survey in documentation-driven and agile companies," In Software Metrics, 2005. 11th IEEE International Symposium, pp. 10-pp.

[9] Hoda R., James N. and Stuart M., "How much is just enough?: some documentation patterns on Agile projects." In Proceedings of the 15th European Conference on Pattern Languages of Programs, p. 13.

[10] Stettina C.J.,Heijstek W, and Fægr T.E.,"Documentation work in agile teams: The role of documentation formalism in achieving a sustainable practice," In Agile Conference (AGILE), 2012, pp. 31-40.

[11] Hoda R., James N. and Stuart M., "Documentation strategies on agile software development projects." International Journal of Agile and Extreme Software Development 1, no. 1 (2012): 23-37.

[12] Uikey N.,Suman U. and Ramani A. K., "A Documented Approach in Agile Software Development," International Journal of Software Engineering (IJSE), Vol. 2, No. 2, 2011, pp. 13-22

