Evaluating the planning process in agile development methods: A systematic literature review

Khalid Khan 1,*, Ibrahim Hassan 1, Usman Waheed 2, Ayub Latif 1, Fadzil Hassan 3

1PAF Karachi Institute of Economics and Technology, Karachi, Pakistan
2Usman Institute of Technology, Karachi, Pakistan
3Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia

A R T I C L E  I N F O

Article history:
Received 16 July 2019
Received in revised form
15 March 2020
Accepted 17 March 2020

Keywords:
Agile
Planning
XP
Scrum
Crystal

1. Introduction

Agile is a software development approach in which requirements and solutions evolve through the collaborative effort of small cross-functional teams and their customers (end users). These teams are self-organized to a major extent and perform adaptive planning and development to achieve early delivery. The focus is on flexible and quick response to change. There are various agile development methods available designed for different circumstances. Some focus on the practices while others focus on managing the workflows. Some deal better with requirements specification and development, whereas some seek to cover the full development life cycle.

Apart from figuring out how the product would be made, Agile planning also helps the software team in measuring and converting the user stories into production-ready software. The task list of an Agile Project is termed as Master Story List. It contains all the requirements that customers want in their required software. The conversion speed of user stories into software is termed as team velocity. This team velocity is used for calculating the team’s productivity and for setting timelines and commitments. Every Agile Methodology, or maybe a combination of these, requires a bird view to plan every phase that the team would be facing. This planning also helps in making commitments, resource planning, and allocation.

The paper presents a systematic literature review (SLR) of various agile development methods in the context of the planning process. Section 2 presents the methodology, which includes the development of research questions as well. In section 3, we discuss various agile development methods with respect to planning processes. Section 4 concludes the paper.

2. Research methodology

In our research, we have used Systematic review to analyze the current studies, findings, and comparing the results. In this review, we have collected and analyzed data through multiple existing research papers and studies. We have formulated multiple research questions and have made findings to answer those. Table 1 list the final selection of previous studies analyzed in this work.

The study has two objectives:
1. To review planning processes and requirements in various agile development methods
2. To identify the best agile development method(s) with respect to the planning process.

The research questions are mandatory to develop a base for SLR. We have extracted some research questions that will be answered in this research. The findings and results of this study will be based on these questions, which are given in Table 2.
To get the maximum relevant studies, we used a search strategy (Kitchenham et al., 2009). This strategy consists of the following components.

Science Direct, IEEE Explore, Springer Link, Scopus, Wiley Online Library, and ACM Digital Library were searched manually to extract the relevant existing studies.

In our study, the search terms have been made by extracting out the necessary keywords of the research area. We have searched keywords as planning in, concatenated with the Agile Process name. Not only had this, but we used question-type keywords too. We have also used the names of planning phases of the Agile Methodologies to get accurate and crystal-clear results. Some examples include:

1. Planning in Extreme Programming.
2. How to plan using Scrum?
3. Sprint Planning Meeting.
4. Planning Poker.
5. Crystal Focus Phase.
6. ASD Speculation Phase.

The research sources are found through oxford press, IEEE, and multiple international Journals, Google Scholars, and Research Gate. We ensure that no counterfeit sources are used in our study.

Fig. 1 demonstrates the type of study, and we used it in our study. We excluded the studies which are not directly or indirectly relevant to our research topic. The rejection of research papers depended on moral standards and to check whether the focused study is going along to the catchphrases, parameters, and phrasings that are significant to this exploration. The studies were excluded from our study when:

- Unverified or unauthentic web links.
- Papers have weak pieces of evidence.
- Weak and outdated references.
- Non-perceived conferences.
- Obscure authors.

To identify the relevant studies, the inclusion criteria are used. It is the qualification criteria for any journal article or research paper to incorporate into this exploration. Incorporation criteria must react to the targets of the study. Many existing studies were reviewed and explored for relevant and useful data, which ranges from conferences, websites, books, and articles. The study material research papers were included when:

- Authentic and official websites.
- Each article or paper had matching research parameters and keywords.
- Studies by authentic and known authors.

### 2.1. Study selection

Fig. 2 demonstrates no. of studies included by each stage of our SLR. The filtering of papers was
done by inclusion/exclusion criteria in the following way.

Fig. 2: Phases of the search process

**Phase 0:** 3198 relevant papers were found by searching string on six aforementioned digital libraries.

**Phase 1:** We selected 408 papers by this phase. In this phase, we read the title and keywords of the papers and filter the selection. If there was any doubt regarding the retrieved papers, we transferred the doubtful papers to the next selection round for another in-depth investigation as it was impossible to analyze the papers by reading keywords and titles.

**Phase 2:** The abstracts and conclusions of the selected studies were analyzed to make sure that all were relevant to our SLR's objective. The selected papers count went to 153 when we applied the inclusion and exclusion criteria on the abstracts and results of the included papers.

**Phase 3:** The final selection round includes the reading of a full text of the included studies by the second phase. A paper was included in our SLR if it met all inclusion criteria. Papers that were less than six pages or papers whose complete texts were unavailable due to limited access or irrelevant papers were excluded. Furthermore, the quality of studies was also analyzed to exclude low-quality papers due to reputation venues, etc.

Twenty papers were selected by this last phase. A critical piece of the study inclusion, data extraction, and synthesis phases have been done by the first author. In every phase, we recorded the reasons for incorporation or prohibition choice for every one of the papers. These reasons were then used for reassessment and discussion with other authors to determine if a paper should be incorporated or not. A double-check, which used a random number of included papers for every step, was done by the second author.

### 2.2. Data extraction and synthesis

The relevant information was extracted from the selected papers. This information is based on data items that would help in answering the SLR's research questions. The separated data was put away in an MS Excel Spreadsheet for further investigation.

Descriptive Analysis was used to analyze the data items, mentioned in Table 3. In order to identify the agile method in the selected papers, we used data item D5. Five steps of thematic analysis (Braun and Clarke, 2006) was followed as detailed below:

1. **Familiarizing with data:** We endeavored to peruse and analyze the extricated research types to shape the underlying thoughts for analysis.
2. **Generating initial codes:** In the second step, we extricated agile techniques from each paper. It ought to be noticed that now and again, we needed to recheck the papers.
3. **Searching for themes:** For every data item, we endeavored to consolidate diverse starting codes
produced from the second step into potential themes. (4) Reviewing and refining themes: The research type and agile methods from 3rd step were verified against each other to acknowledge what themes had to be consolidated with others or excluded (e.g., lack of enough evidence). (5) Defining and naming themes: Through this step, we characterized clear and compact names for each exploration type.

Table 3: Related studies analyzed in the work

| #  | Data Item                                      | Description                                      | RQs          |
|----|-----------------------------------------------|--------------------------------------------------|--------------|
| D1 | Author(s)                                     | The author(s) of the paper.                      | Demographic data |
| D2 | Year                                          | The year of publication of the paper             | Demographic data |
| D3 | Title                                         | The title of the paper                           | Demographic data |
| D4 | Publication Type                              | The type of publication (e.g., journal paper)    | Demographic data |
| D5 | Agile Methodology                             | The agile methodology focused on the paper      | Demographic data |
| D6 | Objectives of planning in various Agile methods |                                                 | RQ1          |
| D7 | Time of planning in various Agile methods     |                                                 | RQ2          |
| D8 | The documented outcome of planning in various Agile methods |                                 | RQ3          |

2.3. Results

We highlight the results extracted from the aforementioned activities in the following subsections. These results do have some minimal interpretations of us, which we will reflect in the discussions section. Here, we are only mentioning demographic, and research design attributes information: Studies distribution, research types, study context, and data analysis type.

Fig. 3: Number of selected studies published per year

3. Planning in agile development methods

By Agile Planning, we calculate estimates and resources that would be required for the software project. The Planning phase helps the development team and stakeholders to discover unidentified risks that could arise during software project development, deployment, and maintenance. This phase can also be made iterative to plan in detail of smaller tasks.

3.1. RQ1: Planning objectives of agile methods

The planning phase of different agile methods can have different goals, but they are only successful if they are clear and focused. XP planning focuses on the product in delivery. LEAN planning clarifies the confusion. Sprint planning produces Sprint Backlog and Resource Allocation. Crystal planning emphasizes on goal definition and individual tasks. FDD is for drafting and allocating initial schedules and responsibilities. DSDM focuses on outline planning of far off phases and detailed planning of the next phase. MSF produces Product Vision and Acceptances Tests. ASD helps realize the uncertainties in complex problems. AUP addresses risk factors and constructs validates the system architecture.
3.2. RQ2: When to plan in agile methods

A plan can be succeeded when it is initiated at the right time. XP plan must be done once an iteration. The LEAN plan must be done when making any decision. A scrum plan must be done at the start of any Sprint. Crystal planning requires to be done before goals defining and task allocation. FDD planning must be done after building a feature list. DSDM planning must be done with the timeline schedule. MSF requires planning to be done after the Vision and Scope approved. In AUP, planning is done before the construction phase, and in ASD, it is difficult to Speculate without Collaborating/Learning. In all Agile Methodologies, planning must be initiated right after the requirements have been gathered, analyzed, and explained to the development team. This is needful for the allocation of resources and scheduling of timelines. LEAN helps in clarifying ambiguous situations by suggesting to plan before any decision making. Alike LEAN, ASD methodology also does not have any fixed phase of planning as it suggests to explore any hypothesis first.

3.3. RQ3: Planning outcomes of agile methods

We get different types of documents/business models when using different Agile Methodologies from the planning phase. Using XP, we get User story cards, task cards. LEAN provides a LEAN Plan. Scrum provides a Sprint Backlog, Crystal planning gives us a Frequent Delivery Plan. We get Feature Sets, Class Diagram by FDD planning. DSDM planning provides us the Outline Prototyping Plan. MSF planning gives us a Master Project Plan, Risk Exposure Rating Form. By AUP Planning, we get Interface prototypes, Project Plan, Use cases, Class/Package Diagrams, and ASD planning provides us with Outline Plan, Delivery Plan.

We obtain a "plan" by every Agile Methodology's planning phase. This output helps the development team from resource allocating to the maintenance phase of the software. We also get visual models with the plan, which demonstrates the construction and delivery plans. These include prototypes, UML diagrams, etc. Scrum provides us a Product Backlog, which lists the tasks that would be delivered at the end of that sprint. The overall review of the planning process in agile methods is given in Table 4.

| Agile Methods | Planning Goals | When to plan? | Planning outcomes |
|---------------|----------------|--------------|-------------------|
| XP            | Guide product in delivery | Once an iteration | User story cards, task cards |
| Lean          | Clarifying confusing situations | When taking any decision | Lean Plan |
| Scrum         | Details, sprint backlog production, Resource allocation | Start of sprint | Sprint backlog |
| Crystal       | Focusing on initial tasks and allocation of goals | Before goals defining and task allocation | Frequent Delivery Plan |
| FDD           | Constructing initial schedules and assigning initial responsibilities | After building a feature list | Feature Sets, Class Diagram |
| DSDM          | Plan in detail for the next phase and to plan in outline for the phases that are further away | Timeline scheduling | Outline Prototyping Plan |
| MSF           | Product vision is met through the requirements and the acceptance tests are developed | After Vision and Scope approved | Master Project Plan, Risk Exposure Rating Form |
| AUP           | Address known risk factors and to establish and validate the system architecture | Before construction phase | Interface prototypes, Project Plan, Use Cases, Class/Package Diagrams |
| ASD           | Acknowledgments the reality of uncertainty in complex problems | It is challenging to Speculate without Collaborating/Learning and vice versa | Outline Plan, Delivery Plan |

As per our analysis of the various Agile Methodologies, Scrum seems to be the best option for planning:

- The product owner creates a wishlist of high-priority tasks as per business values, which is proposed as a product backlog.
- Scrum teams plan a sprint planning session where the wishlist is broken down into smaller, more manageable tasks.
- The Scrum team produces and plans the implementation of the sprint backlog.
- The team decides the sprint's time duration (usually two weeks long).
- Daily standup meetings are held to discuss bottlenecks, the status of previous tasks, and tasks due before the next meeting.

- The Scrum Master guides, facilitates, motivates, and brings focus to the team.
- Product owner reviews tasks at the end of a sprint.

4. Conclusion

The work has compared various agile methods in the context of planning processes with two objectives. One is to review planning processes and requirements in various agile development methods, and the second objective is to identify the best agile development method(s) with respect to planning processes. We found that Scrum is the best choice among various agile methods in the context of planning requirements and processes.
Compliance with ethical standards

Conflict of interest

The authors declare that they have no conflict of interest.

References

Abrahamsson P, Warsta J, Siponen MT, and Ronkainen J (2003). New directions on agile methods: A comparative analysis. In the 25th International Conference on Software Engineering, IEEE, Portland, USA: 244-254. https://doi.org/10.1109/ICSE.2003.1201204

Ahmad G, Soomro TR, and Brohi MN (2014). Agile methodologies: Comparative study and future direction. European Academic Research, 1(11): 3826-3841.

Anwer F, Aftab S, Shah SM, Ahmad G, Soomro TR, and Brohi MN (2014). Agile methodologies: A comparative analysis of two popular agile process models: Extreme programming and scrum. International Journal of Computer Science and Telecommunications, 8(2): 1-7.

Braun V and Clarke V (2006). Using thematic analysis in psychology. Qualitative Research in Psychology, 3(2): 77-101. https://doi.org/10.1191/1478088706qp063oa

Febbraro N and Rajlich V (2007). The role of incremental change in agile software processes. In the Agile 2007, IEEE, Washington, USA: 92-103. https://doi.org/10.1109/AGILE.2007.58

Fernandes JM and Almeida M (2010). Classification and comparison of agile methods. In the 7th International Conference on the Quality of Information and Communications Technology, IEEE, Porto, Portugal: 391-396. https://doi.org/10.1109/QUATIC.2010.71

Highsmith JA and Highsmith J (2002). Agile software development ecosystems. Addison-Wesley Professional, Boston, USA.

Kampker A, Meckelborg A, Burggraf P, and Netz T (2013). Factory planning scrum: Integrative factory planning with agile project management. In the International Conference on Competitive Manufacturing, Stellenbosch, South Africa.

Kanwal F, Bashir K, and Ali AH (2014). Documentation practices for offshore agile software development. Life Science Journal, 10(11): 70-73.

Kitchener B, Brereton OP, Budgen D, Turner M, Bailey I, and Linkman S (2009). Systematic literature reviews in software engineering: A systematic literature review. Information and Software Technology, 51(1): 7-15. https://doi.org/10.1016/j.infsof.2008.09.009

Kniberg H (2015). Scrum and XP from the trenches. 2nd Edition, Lulu Press, Morrisville, USA.

Matharu GS, Mishra A, Singh H, and Upadhyay P (2015). Empirical study of agile software development methodologies: A comparative analysis. ACM SIGSOFT Software Engineering Notes, 40(1): 1-6. https://doi.org/10.1145/2693208.2693233

Mendes TS, Farias MA, Mendonça M, Soares HF, Kalinowski M, and Spinola RO (2016). Impacts of agile requirements documentation debt on software projects: A retrospective study. In the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy: 1290-1295. https://doi.org/10.1145/2851613.2851761

Merzouk S, Elhadi S, Ennaji H, Marzak A, and Sael N (2017). A Comparative study of agile methods: Towards a new model-based method. International Journal of Web Applications, 9(4): 121-128.

Minarik M (2004). Analysis of the change management process in agile software development. Diploma Thesis, Masaryk University, Brno, Czechia.

Myklebust T, Ståhane T, Hansen GK, Wien T, and Hauge B (2014). Scrum, documentation and the IEC 61508-3: 2010 software standard. In the International Conference on Probabilistic Safety Assessment and Management, PSAM, Hawaii.

Shelly (2015). Comparative analysis of different agile methodologies. International Journal of Computer Science and Information Technology Research, 3(1): 199-203.

Stettina CJ and Heijstek W (2011). Necessary and neglected? An empirical study of internal documentation in agile software development teams. In the 29th ACM International Conference on Design of Communication, Association for Computing Machinery, Pisa, Italy: 159-166. https://doi.org/10.1145/2038476.2038509

Stettina CJ and Kroon E (2013). Is there an agile handover? An empirical study of documentation and project handover practices across agile software teams. In the International Conference on Engineering, Technology and Innovation (ICE) and IEEE International Technology Management Conference, IEEE, The Hague, Netherlands: 1-12. https://doi.org/10.1109/ITMC.2013.7352703

Tria J (2006). RUP vs. MSF: A comparative study. Available online at: https://bit.ly/3aG5ZSU

Waddell D and Sohal AS (1998). Resistance: A constructive tool for change management. Management Decision, 36: 543-548. https://doi.org/10.1108/00251749810232628

Wagenaar G, Overbeek S, Lucassen G, Brinkkemper S, and Schneider K (2018). Working software over comprehensive documentation—Rationales of agile teams for artefacts usage. Journal of Software Engineering Research and Development, 6: 7. https://doi.org/10.1186/s40411-018-0051-7