International Journal of Mechanical Engineering and Technology (IJMET)
Volume 10, Issue 02, February 2019, pp.671–687, Article ID: IJMET_10_02_070
Available online at http://www.iaeme.com/ijmet/issues.asp?JType=IJMET&VType=10&IType=02
ISSN Print: 0976-6340 and ISSN Online: 0976-6359

A SYSTEMATIC REVIEW OF REQUIREMENT ENGINEERING PRACTICES IN AGILE MODEL

Julius Okesola
Department of Computational Sciences, First Technical University, Ibadan, Nigeria

***Marion Adebiyi**
Department of Computer Sciences, Landmark University, Omu-Aran, Nigeria
Department of Computer Sciences, Covenant University, Ota, Nigeria

Kennedy Okokpujie
Department of Electrical and Electronic Engineering, Covenant University, Ota, Nigeria

David Odepitan, RowlandGoddy-Worlu, Olamma Iheanetu, Zacchaeus Omogbadegun
Department of Computer Sciences, Covenant University, Ota, Nigeria

Ariyo Adebiyi
Department of Computer Sciences, Landmark University, Omu-Aran, Nigeria
Department of Computer Sciences, Covenant University, Ota, Nigeria
* Corresponding Author

ABSTRACT

Requirement engineering practice (REP) are developed from requirement engineering processes to guide the engineers in requirement definition. Even though, the practices in agile present a few restrictions and open up several challenges to software industry, they help to address difficulties of conventional models and avert unnecessary cost. However, there are no clear-cut separation of these practices into their individual processes since the practices are closely used and the requirements are ever dynamic. Using Quasi-Gold standard to develop the search strategy and validating the same strategy by Snowballing, this study adapted Kitchenham guidelines to perform a systematic literature review towards identifying the common REP and the extent at which the REP are imbibed in software engineering. Requirement Management is the most popular practice while pre-testing is barely noticed. Eight of their associated challenges are identified and solutions discussed. REP is fully accepted in software engineering going by the annual rise in their discussions but future works is required on the detailed analysis of the root cause of these challenge.

Keywords: Engineering practice; Engineering process; Quasi-Gold; requirement engineering practice; requirement engineering process; requirement management; and snowballing;

Cite this Article: Julius Okesola, Marion Adebiyi, Kennedy Okokpujie, David Odepitan, RowlandGoddy-Worlu, Olamma Iheanetu, Zacchaeus Omogbadegun and Ariyo Adebiyi, A Systematic Review of Requirement Engineering Practices in Agile Model, International Journal of Mechanical Engineering and Technology, 10(02), 2019, pp. 671–687
http://www.iaeme.com/IJMET/issues.asp?JType=IJMET&VType=10&IType=02

1. INTRODUCTION

The traditional method for requirements engineering (RE) has always been challenged by the ever changing business environment and programmers are being compelled to manage requirement that have tendency to change before the project completion [1]. Hence, the call for agile especially now that needs of most organizations have drifted from being static to dynamic and progressive. However, requirement engineering in agile are guided by practices but comes with processes [2] which are fundamentally different from one organisation to another due to several factors including disciplinary involvement, technical maturity, organisational culture, and application domain [3]. Hence, no requirements engineering process is ideal [4],

A process is "an organised set of activities which transforms inputs to outputs" [5][6]; they are RE activity structured to elicitate, analyse, document and validate software requirements [7]. Practices however, are those activities emerged from requirement engineering process to help requirement engineers define good requirements [3][8]. They could be new or industry best practices to address the deficiencies in an elicited requirement [9]. Below are some that have been developed to identify and avert unnecessary cost [10]:

a) *Direct Communication (DC).* This is a face-to-face communication that calls for the clients or stakeholders and the engineers to communicate directly to each other. It is an efficient practice for requirement gathering as it produces an output even with minimal documentation [11].

b) *Customers Involvement (CI)*. This is where clients participate directly and interact with the software or requirement engineers. It is a good practice because customers direct involvements in the process ensures well-defined and prioritized requirements [12] as customers' involvement is a major success factor in software development [13][14].

c) *User Stories (US).* This practice in agile enables an end-user to describe the necessary features of the system from his own perspectives [15]. It promotes understanding and communication amongst stakeholders and aids requirement elicitation.

d) *Iterative Requirement (IR).* The changes in requirement calls for frequent discussions amongst the stakeholders to ensure that requirements remain clearly defined. Frequent meeting with stakeholders strengthens relationships and ensures the requirements remains in line with development [16].

e) *Requirement Prioritization (RP).* The elicited requirement are prioritized to determine what requirement should be given considered first [17]. It is important to consider high priority requirements before the low priority.

f) *Change Management (CM).* The main activity here is the addition and removal of features in the requirement document [18] as requested by the clients. It remains a distinguishing practices of Agile Software Development (ASD).

g) *Prototyping (PR).* With an initial set of requirement, a prototype is built in order to meet the immediate system requirements due to the fact that the stakeholders of this system have little or no idea of the final system requirements [19].

h) *Pre Testing (PT).* This is testing before coding and involves writing functional codes for the elicited requirements which to enhance feedbacks by test cases in the event of test failures. It also proposes another approach called automated acceptance test-driven development - ATDD [20].

i) *Requirement Management (RM).* It is a practice where features list or product backlog are maintained, monitored and tracked as in the case of scrum [18].

j) *Review Meetings (RE).* There is a need for review meetings and acceptance to bring the stakeholders together in order to access the current status of the development and check volatile requirements [21].

k) *Code refactoring (CR).* This is an important practice especially for volatile requirements. It is the revisiting of the code structures used in development for possible modification towards improvement [22].

2. RELATED WORKS

The traditional requirement software development process is faced with so many problems among which is the ‘traditional requirement engineering processes’, thereby paving ways for a global adoption of ASD practices. Based on systematic mapping therefore, [23] highlighted major differences between the traditional and agile practices, as well as challenges and solutions of using ASD. To address this, [24] introduced the best guidelines for agile requirement engineering towards enhancing the quality of the requirement elicited, and [25] presented a unique model - FlexREQ - to handle RE approaches.

Meanwhile, the need for the adoption of ASD becomes apparent when traditional approaches can no longer handle requirement activities [25] due to the rapid and dynamic changes in requirement as related to business environment. [26] Investigated the cause for this shift by comparing studies between the traditional requirement and ASD approaches, and discover that the accommodating features of Agile to rapidly changing requirement in the industry was the main factor. [27] presented an overview of project management for small companies and reasons behind companies preferring Scrum, an ASD. Similarly, [28] collaborated the needs for adopting agile methodologies but views requirements from the points of stakeholders goal. They proposed an Agile Technique for Agent Based Goal Elicitation (ATABGE) which was based on the mechanisms of agile practices for the extraction of goals from the user.

Agile methodologies are based on small development cycles with continuous communication and need for stakeholders’ involvement without any formality and theoretical modelling. However, since requirement modelling is formerly required in some projects, [29] presented a logical architecture to address this problem by including a formal model of delivering requirements as inputs which are derived from user stories and delivered to multiple scrum teams. [8] also proposed gamification of requirements where requirements practices - user stories and acceptance tests - were introduced.

While [30] carried out data analysis on 16 software development companies, [15] conducted a systematic literature review on 21 papers towards identifying Agile practices commonly in use. Although, these studies independently identified various challenges faced by these practices, [31] already proposed a conceptual framework in order to make the practices more efficient and effective. This framework provides guidelines and easy approaches to software development.

3. RESEARCH METHODOLOGY

Being guided by our research goal, this study adapted Kitchenham [32] guidelines to perform empirical study, and Quasi-Gold standard (QGS) as proposed by Zhang et al. [33] to develop the search strategy. Kitchenham et al. [32] standard has proved to be a set of concrete guidelines, good enough to assist researchers in performing empirical studies while QGS is a thorough search strategy approach effective in SLRs to improve the reliability and validity of search processes [34]. Five researchers performed this review out of which two who are experts played supervisory roles and dealt with divergences in-between others. The research questions were formulated by the experts while the comprehensive literature search was performed by the other three who are postgraduate students.

3.1. Research questions

The primary questions formulated to address the research objective of this tertiary study is *"To what extent are REP imbibed in software engineering?"* This question is however decomposed into the following secondary research statements to address the principal issues required towards achieving our research goal:

RQ1. Which REP is common in agile?

RQ2. What are the general challenges in REP?

RQ3. How are RE issues resolved in Agile Method?

3.2. Search strategy

Since no search is capable of collecting all relevant studies in total [35][36], it is important to devise a relatively objective approach to improve the quality of a search strategy. This study therefore adopts Quasi-Gold standard (QGS), which rigorously integrates the automated and manual search strategies together for sensitivity and precision. Hence, our search is in three stages - manual study search, automatic search, and snowballing. The research strategy was developed by one of the two supervisors and reviewed by the other. However, the manual search, automated search, and snowballing were independently handled by the three research students and their results compared to resolve any possible discrepancies. The studies were also checked independently by the two supervisors to avoid potential bias.

3.2.1. Manual Search

We guided against a study limitation of [34] where books, technical reports, thesis and dissertations were not searched because of publication bias based on an assumption that authors hardly publish negative results. Being motivated by [37] that some authors are already publishing negative results in software engineering domain, we did not limit our search to only journal and conference proceedings but to all other publications where quality reports may be obtained. Hence, the authors had to visit some Nigerian universities and professional libraries recognised as highly specific to Software Engineering (SE), Empirical Software Engineering (ESE) and Evidence-Based Software Engineering (EBSE) since many quality books are not available online just as many study thesis and dissertations are not enlisted on their university's websites. The manual search was done in September 2017 on the four selected locations looking for highly reputed generic publications. Only [38] was obtained.

Since [34] was free of quality issues, this work adopted their study list which ranges between 2004 and 2017 as the starting set. Bearing in mind that "the purpose of manual search is to establish an effective quasi-gold standard for improving the follow-up automated search rather than striving to capture as many SLRs as possible" [35], the search was limited to selected venues slightly different from that of [34]. This did not in any way affect the originality of our search as only searched venues that are inexplicit to SE, ESE, and EBSE were ignored.

3.2.2. Automated Search

The auto search was conducted over eight digital libraries and broad indexes namely: Citeseer, ACM, IEE Computer society digital library, EBSCO, Springer Link, Scopus, and Science Direct. The search was limited to the document title, abstract and keywords as quality papers hardly miss their main research subject out of their metadata. Our automated search was made simple as only one set of search string (SS) - "Requirement engineering practice" AND agile - was employed over the selected indexes (through Article title, abstract and keyword). Requirement Engineering practices are many (section 2.1) and some literatures may have used them as terminologies to describe their studies. However, such a work that fails to recognise its main subject –RE - in its basic metadata cannot be adjudged to be quality and should not be recognised as one [34].

Using the same string of [34], we noticed that four publications were missing, all of which were traced to mapping issues. Since the search string was already validated for quality, we decided to modify and recode it into the following equivalent forms that matches the syntax of each of the selected digital libraries:

3.2.3. TITLE-ABS-KEY("Requirement engineering practice" *AND* agile*)*

We acknowledge sensitivity as an important metric to measure the quality of a search strategy. Since it is a percentage of related studies covered [35] aiming at evaluating and validating the search string for any possible adjustment, it was mathematically expressed as:

$$Sensitivity = \frac{Number\ of\ relevant\ studies\ retrieved}{Total\ number\ of\ relevant\ studies} 100\%$$

A few studies were actually omitted in error during the evaluation of this search query. However, since the corresponding quasi-sensitivity was as high as 88.3% and quasi-precision was 1.59%, our updated search string is adjudged valid and suitable for the automated search.

3.2.4. Snowballing

Our search covered the major publishers as the digital libraries searched are the most important to SE studies. The process was also validated by QGS, which is “an approach that devises a rigorous search strategy for improving the validity and reliability of a search phase” [35]. Notwithstanding, there is still a slim possibility of some studies being missed out. Hence, snowballing strategy was employed to further identify as many qualified studies as possible. In which case, Google Scholar was used to find the papers associated with some (recent??) REP seminal papers on an assumption that they should be cited by quality studies. This study list were then reviewed for possible qualified articles.

3.3. Inclusion/ Exclusion

The flow diagram of [34] was adapted for this study selection as depicted in fig 1

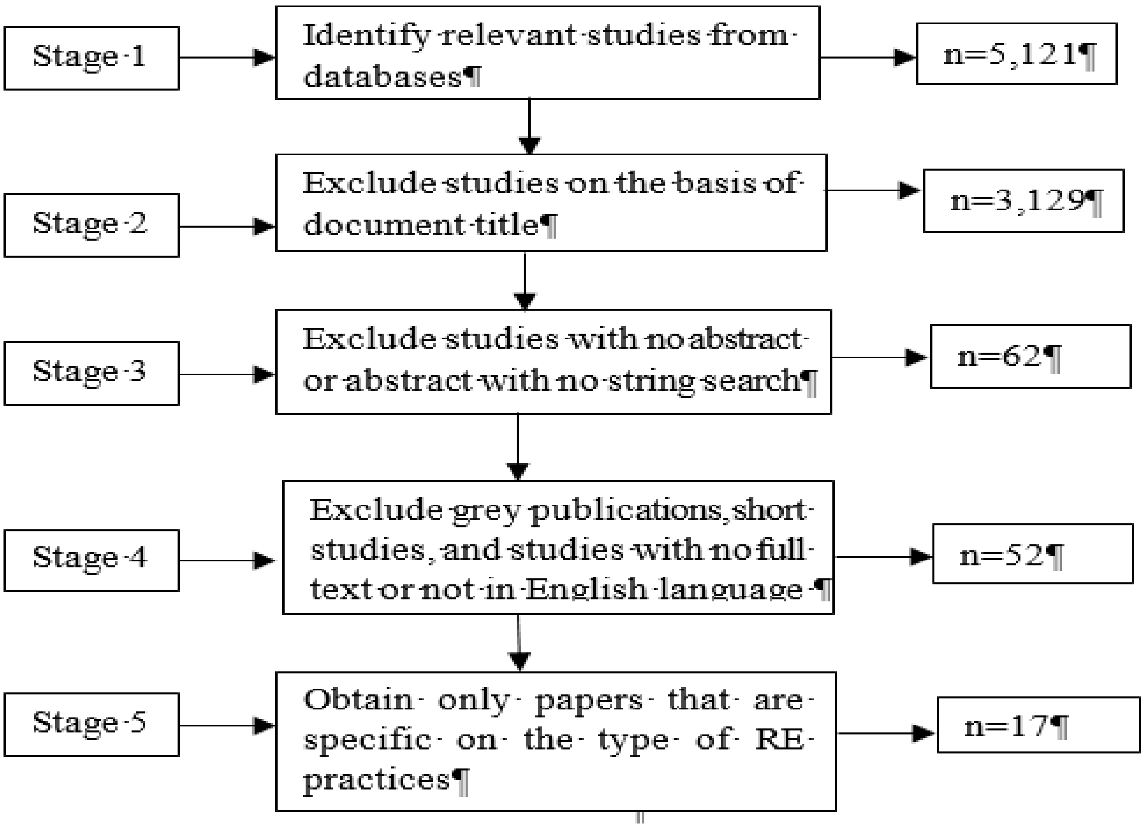

Figure. 1. Flow diagram of the study selection [34]

Searches from all the selected digital libraries and broad indexes totalling 5,121 are collated and subjected to initial screening paying special attention to the study title and abstract. Exporting the total list to Microsoft excel and sorting on the document title, the total papers were reduced to 3,129 as many are multiple indexed. Since abstract statement of any quality paper is not expected to be silent on its main subject [34], we went through the keywords and abstract columns to *remove* studies that have none, or did not include the search string (SS) - *Requirement engineering practice,* thereby reducing the total number to 62. We observed that most papers found in other libraries are contained in Scopus except 58 which have all been excluded as they failed the initial screening exercise. Hence the entire 62 studies subjected for further assessment are Scopus index, and were downloaded for the two independent teams of researchers to screen for exclusion going by the following criteria:

- The paper is not written in English language
- The full-text is copyright and therefore not available
- The paper is a technical report or other grey publication without peer-review.
- The study was generally on SDLC and not on a particular agile methodology
- The study was on agile methodologies but not REP
- The paper too short (like 5 pages) to convey quality

This process dropped three papers whose full texts were not available. Another two papers which were not written in English language was identified but only one was dropped when it could not be interpreted by the available French language translator. A total of four papers were excluded because they were not full-fleshed research submission but mere technical reports or grey publications with no peer-review. Two papers with respective four and five pages were considered too short to convey sufficient details and was therefore excluded leaving the study list at 52. However, majority of these documents, though discussed agile types and associated challenges, are not being specific on REP types. These were all backed out to have a final study list of 17 articles depicted on table 1. Notes from each research team were compared against each other and discrepancies resolved to avoid false rejection of relevant papers.

3.4. Data Extraction

We followed similar procedure used in study selection except that the process here was purely manual. The following attributes were extracted from each of the 17 publications.

- The publication year (to show how current the study)
- The type of the document (journal, work-shop, conference, book chapter).
- The type of Agile model(s) discussed
- Whether challenges in REP are stated. This is to help in addressing RQ2

Table 1. Challenges to REP

Study No	Year	Study type	Agile type	Problem raised?	DC	CI	US	IR	RP	CM	PR	PT	RM	RE	CR	**Yearly TOTAL**
[39]	2002	Conf	XP	Y	1		1	1			1		1			**5**
[30]	2008	Conf	Scrum	Y	1			1	1		1		1			**6**
[40]	2008	Conf	XP	Y			1									
[38]	2010	Book		Y		1	1						1			**8**
[16]	2010	Article	XP/Scrum	Y	1			1	1		1		1			
[41]	2012	Article	Scrum	Y	1		1	1			1					**9**
[20]	2012	Conf	ATDD	Y	1			1	1		1		1			
[42]	2014	Conf		N			1						1	1		**10**
[43]	2014	Conf		N	1				1	1		1		1		
[44]	2014	Article	FDD	Y		1							1			
[15]	2015	Article	Scrum	Y	1	1	1	1	1	1	1	1	1	1	1	**24**
[12]	2015	Conf		Y	1	1	1	1	1	1	1	1	1	1	1	

[45]	2015	Article	XRE	Y		1				1						
[13]	2016	Article	Scrum	N		1				1		1		1		**4**
[46]	2017	Conf		Y	1	1		1	1		1		1	1	1	**14**
[47]	2017	Conf		Y	1	1			1	1	1		1			
[48]	2018	Article	Scrum	Y	1	1	1	1	1	1	1	1	1	1		**10**
Total	**17**				**11**	**9**	**8**	**9**	**9**	**7**	**10**	**5**	**12**	**7**	**3**	
Rank					2nd	4th	7th	4th	4th	8th	3rd	10th	1st	8th	11th	

4. RESULTS

Various observations and results of our research process are presented in this section following the response to the research questions

4.1. Selected Studies

The study list of publications summarized on table 1 was later used to answer research questions raised in this work with the selected primary studies being elicited by their publication year, document type, and agile model discussed.

Table 1 infers that aside year 2016 where most articles were not being specific, discussions on REP have been on increase since year 2002 with the maximum being reported in 2015. The decline in 2017 and 2018 (Fig 1a) may have been a result of our search date which did not consider articles after September 2017 except [48] that was already accepted in 2017 but slated for publication in 2018. The table also shows that studies are hardly reported in books as all but [38] were found in journal and conference proceedings. Similarly and for reasons yet to be investigated, many authors prefer to work on Scrum making it the most popular agile model (fig 1b), and globally used especially in distributed projects.

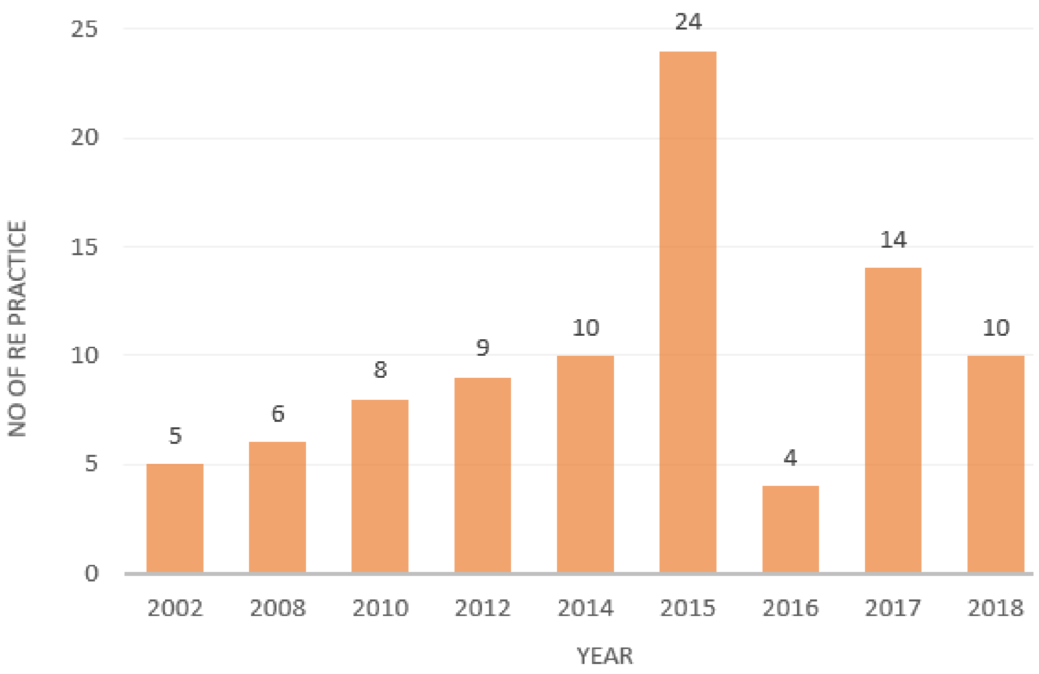

Figure. 1a: REP annual growth **(BAR CHART)**

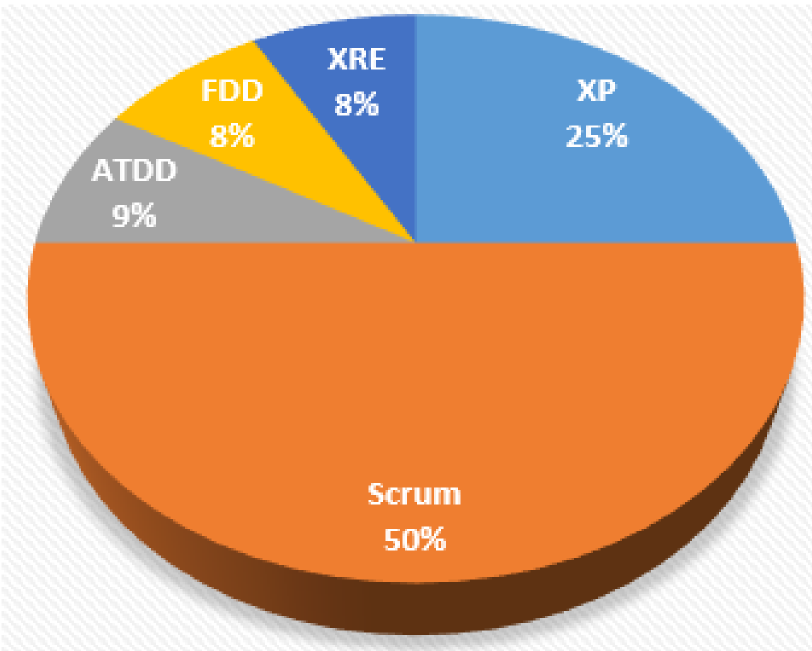

Figure. 1b: Scrum Popularity (**PIE CHART**)

4.2. Study Distribution/RE Practice

Although there are several REP in Agile, only 11 are employed by our primary studies. The practices are as described in section 2.1 but the number of studies that embraced them are depicted on table 1 and fig 2

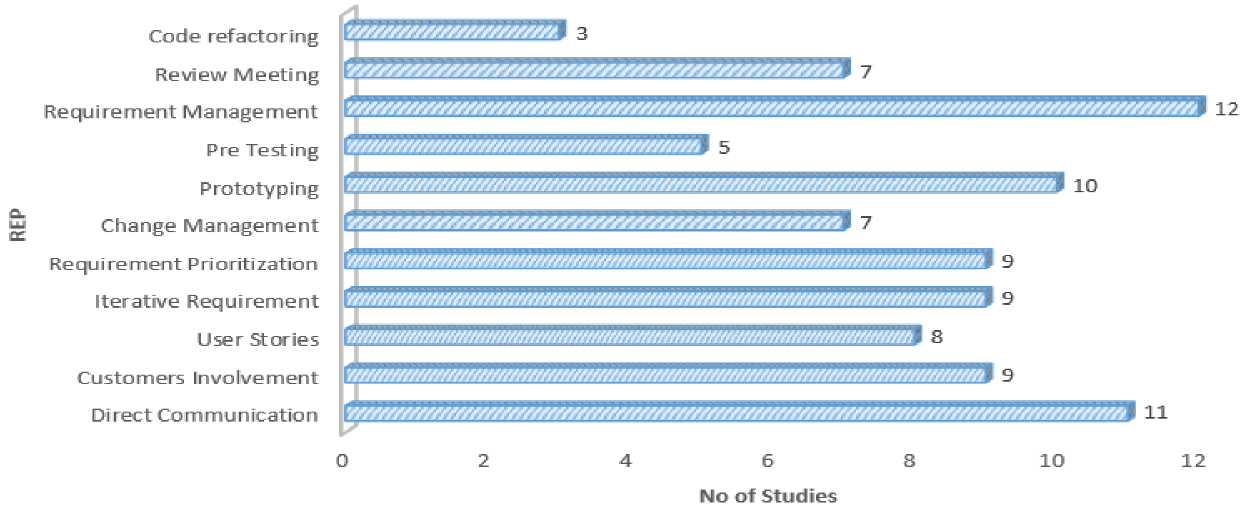

Figure. 2: Study Distribution

4.3. Solutions to challenges in REP

While RE practices in agile helps to address difficulties of conventional models, they present a few restrictions and open up several challenges to software industry. As contained in its primary studies, this study summarises on table 2 a list of notable REP challenges, their impacts and potential solutions.

Table 2. Challenges to REP

S/n	Challenges	Impact	Solutions
1	Minimal documentation [18][16][12]	Traceability issues [49]	Use of user stories with detailed artefacts [12].
2	Customer availability [50][12][18]	Increase in rework [8][12].	Surrogate customers to represent a genuine client [16]
3	Budget and time estimation [16][2]	Increase in cost (budget)	Project initial valuation
4	Inappropriate architecture [30][16]	Project delay in commencement or continuation [16].	Frequent communication; discourage Code refactoring [12]
5	Neglecting non-functional requirements [14][51][52]	Systems insecurity; usability and performance are at stake [51].	Accurate modelling of agile processes [13].
6	Customer inability and agreement [12][16]	Project delay; increase in rework [12].	Frequent communication; Creation of delivery stories [16]
7	Contractual limitations [12]	Project failure; and budget overrun [53].	Adoption of lawful measures [12].

8	Requirements change and change evaluation [54][55].	Increase in work delay [23].	Usage of RE KOMBINE framework for change evaluation [55].

5. ANALYSIS AND DISCUSSION

Answers to research questions are discussed in this section.

5.1. RQ1: Which REP is common in agile?

Requirement Management (RM) is ranked first amongst the 11 practices under review (table 1). For reasons yet to be investigated, it remains the most embraced followed by Direct Communication (DC) and Prototyping (PR) as depicted in Fig 2.

5.2. RQ2: What are the general challenges in REP?

RQ3: How are RE issues resolved in Agile?

The systematic literature review conducted in this study have identified quite a number of challenges in RE practices and how they are being addressed in agile models. They are as highlighted on table 2 and iterated in this section:

5.2.1. Minimal documentation

Minimal documentation is always due to rapid or sudden changes in requirements and it is a major challenge posed by agile methods [16] as it is often difficult for development team to debug software with inadequate documentation. It also results to Traceability issues where developers are unable to scale the software, evolve the application over time and induct new members into their team [49]. This challenge is bigger in large projects with large team members where for instance, workspace (room or office) is not large enough to accommodate members thereby making requirements verbal communication insufficient and documentation inadequate.

Essential communication may help to improve on requirement documentation [53][23] since the former is essential to the latter [15]. However, an important solution is the use of user stories complemented with delivery stories (detailed artefacts) which helps developers to make right implementation choices and proper documentation at the coding stage [12].

5.2.2. Customer availability

Agile model calls for the presence of the business owners at every requirement change process. This advocacy is practically unrealistic [18]. as it poses some challenges in terms of time, cost and workload [3][50][50]

A way out of this problem is the usage of RE KOMBINE framework, where surrogates or intermediary clients are engaged when genuine clients are not available [12] to tackle inconsistencies in requirements change during evaluation [55].

5.2.3. Budget and schedule estimation

The use of agile enables the initial valuations of a project [16] but the extent of a project is based on known user stories some of which may be discarded in forthcoming iterations. This is a challenge because dynamic planning and design makes it impossible to give requirements estimates. Hence, cost estimations could be significantly affected [16].

A solution proffered by Ramesh et al., [16] and Alam et al., [1] is for agile method to adopt project initial valuations approach where developers will have right requirements and planning to give an appropriate cost estimate.

5.2.4. Inappropriate architecture

New requirements may make project architecture inadequate at the later stage of development [30] as it encourages refactoring - an ongoing activity amongst agile teams where codes are changed whenever there is a change in requirement. This delays project completion as the unexpected occurrences impedes project's commencement or continuation [16].

Although, this challenge may be minimised when user story is accurately modelled and communication amongst the project members is enhanced, code refactoring (a process of restructuring existing computer code-changing factoring without changing its external behaviour) must be avoided to prevent changes that may be cumbersome to handle [16].

5.2.5. Neglecting non-functional requirements

Non-functional requirements (NRFs) focus on system quality including maintainability, testability, usability, and security [52]. However, these requirements are often neglected in agile model thereby posing major challenges that may result to massive reworks.

Domah & Mitropoulos [56] and Farid & Mitropoulos [14] have proposed novel and slight artefacts such as agile use cases, agile loose cases and agile choose cases by using NFRs Modelling to effectively model agile processes.

5.2.6. Customer inability and agreement

Customers' incompetence due to their knowledge gap, and inability to agree reach consensus is a challenge next to 'customers' availability' [12]. Hence, members differences may slow the project implementation pace thereby negatively affecting the group performance [16].

Challenges such as customer incompetence in decision making, adequate knowledge can be resolved by the creation of delivery stories to accompany user stories; and frequent communication [16]

5.2.7. Contractual limitations and requirements volatility

Contract is legally binding as changes are not usually permitted for the fear of project failure and budget overrun. While rework may be avoided by customers' involvement and awareness, the contractual limitation could be a challenge especially when some changes are inevitable. Lawful measures should therefore be taken to maintain a strategic distance from such a circumstance and fittingly handle the adaptable idea of coordinated RE.

This challenge can be addressed by adopting the 'settled instalment per discharge' to secure speculations and averts unpredictability of prerequisites [12].

5.2.8. Requirements change and change evaluation

The flexibility nature of agile models naturally induces requirement changes which causes work delay [23] and extremely difficult to evaluate in most cases.

JIRA [57] is an agile RE tool recommended for usage in challenging projects [54], and RE KOMBINE framework may be adopted for easy evaluation to tackle inconsistencies in requirements changes [55]. These approaches ensure the requirements are formally specified but flexible enough to accommodate the changes.

6. STUDY LIMITATION

Systematic reviews do have limitations of bias in study selection and the possible imprecision in data extraction. When developing its research strategy therefore, this study treated its search-string-building process as a learning process that included experimentation and then followed its research questions to define keywords for auto-search in electronic databases. Notwithstanding, since searches in software engineering are based on the search strings which are language dependent [58], there is a possibility that some relevant studies which may have used alternative terminologies (user stories, features, tasks, etc.) to describe their keywords were missed out.

When attempting to ensure quality in their study sample, the authors considered publications below six pages as too short to covey quality and therefore excluded them from their list. Although this limitation may not be material in this study as only two articles were affected, there is a possibility of a false rejection error as length of publications may not be a true test of quality reporting.

This study identified challenges in REP and their associated solutions but failed to detail the underlying causes. This is because none of the primary studies that describe the challenges hardly discuss the underlying causes in details as the authors of those respective papers must have found it unnecessary. Again, this limitation is of little importance here as it is out of this research scope.

7. CONCLUSION

The ever changing business environment facilitates the general adoption of agile methodologies when traditional approaches can no longer handle requirement activities/**practices??**). REP are therefore developed to manage the changes and unnecessary cost in agile models. This work has identified the major REPs in agile and addressed the principal issues in order to establish the extent at which REP is imbibed in software engineering.

This study presents Requirement Management as the most popular REP and confirm that the adoption of REPs has been on increase since 2002. Meanwhile, the associated studies are hardly published in books but in conference proceedings and journals. Some challenges posed by REP in agile are also identified as well as their potential solution.

Most researchers prefer to be silent on the detailed analysis of the underlying causes of challenges in REPs. This work appreciate the need for such a detailed study and therefore recommend for future works that studies on challenges in REPs should be encouraged and must be interested in the root causes towards offering a more feasible solution.

REFERENCES

[1] S. Alam, S. Nazir, S. Asim, and D. Amr, “Impact and Challenges of Requirement Engineering in Agile Methodologies: A Systematic Review,” Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 4, pp. 411–420, 2017.

[2] V. Tripathi and A. Goyal, “Agile Requirement Engineer: Roles and Responsibilities,” IJISET - Int. J. Innov. Sci. Eng. Technol., vol. 1, no. 3, pp. 213–219, 2014.

[3] R. K. Kandt, “Software Requirements Engineering : Practices and Techniques,” in Jet Propulsion Laboratory, Califonia Institute of Technology, 2003.

[4] O. J. Okesola and O. S. Oluwafemi, “Meta-heuristics Based Multi-Layer Access Control Technique (MBMAC),” Ann. Comput. Sci. Ser., vol. 9, no. 1, pp. 399–404, 2011.

[5] G. Kotonya and I. Sommerville, “Requirements Engineering Processes,” 1998.

[6] O. J. Okesola, K. O. Okokpujie, A. A. Adewale, S. N. John, and O. Omoruyi, “An Improved Bank Credit Scoring Model: A Naïve Bayesian Approach,” in 2017 International Conference on Computational Science and Computational Intelligence (CSCI), 2017, pp. 228–233.

[7] S. Amber, N. Shawoo, and S. Begum, “Determination of risk during requirement engineering process,” J. Emerg. Trends Comput. Inf. Sci., vol. 3, no. 3, pp. 358–364, 2012.

[8] P. Lombriser, F. Dalpiaz, G. Lucassen, and S. Brinkkemper, “Gamified requirements engineering: Model and experimentation,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9619, pp. 171–187, 2016.

[9] O. Folorunso, O. S. Ogunseye, J. O. Okesola, and O. M. Olaniyan, “Visualizing e-voting results,” J. Theor. Appl. Inf. Technol., vol. 16, no. 1, 2010.

[10] A. Tahir and R. Ahmad, “Requirement Engineering Practices - An Empirical Study,” 2010 Int. Conf. Comput. Intell. Softw. Eng., pp. 1–5, 2010.

[11] S. S. Johnson, “Requirements Engineering in an Agile Environment,” Northgroup, no. October, 2009.

[12] M. Daneva et al., “Agile requirements prioritization in large-scale outsourced system projects: An empirical study,” J. Syst. Softw., vol. 86, no. 5, pp. 1333–1353, 2013.

[13] W. M. Farid and F. J. Mitropoulos, “NORMATIC: A visual tool for modeling non-functional requirements in agile processes,” in Conference Proceedings - IEEE SOUTHEASTCON, 2012.

[14] W. M. Farid and F. J. Mitropoulos, “Novel lightweight engineering artifacts for modeling non-functional requirements in agile processes,” Conf. Proc. - IEEE SOUTHEASTCON, 2012.

[15] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, “A systematic literature review on agile requirements engineering practices and challenges,” Comput. Human Behav., vol. 51, pp. 915–929, 2015.

[16] B. Ramesh, L. Cao, and R. Baskerville, “Agile requirements engineering practices and challenges: an empirical study,” Inf. Syst. J., vol. 20, no. 5, pp. 449–480, 2010.

[17] M. Batra and A. Bhatnagar, “Requirements Prioritization: A Review,” Int. J. Adv. Res. Sci. Eng. Technol., vol. 3, no. 11, pp. 2899–2904, 2016.

[18] K. Elghariani and N. Kama, “Review on Agile requirements engineering challenges,” 2016 3rd Int. Conf. Comput. Inf. Sci. ICCOINS 2016 - Proc., pp. 507–512, 2016.

[19] R. Eckert, Business Model Prototyping. 2014.

[20] B. Haugset and T. Stålhane, “Automated acceptance testing as an agile requirements engineering practice,” in Proceedings of the Annual Hawaii International Conference on System Sciences, 2012, pp. 5289–5298.

[21] B. P. Douglass, Agile Systems Engineering. 2015.

[22] T. Javdani Gandomani, H. Zulzalil, A. A. A. Ghani, and A. B. M. Sultan, “A Systematic Literature Review on relationship between agile methods and Open Source Software Development methodology,” ArXiv e-prints, vol. 1302, p. 2748, 2013.

[23] I. Inayat, L. Moraes, M. Daneva, and S. S. Salim, “A Reflection on Agile Requirements Engineering: Solutions Brought and Challenges Posed,” Sci. Work. Proc. XP2015, p. 1, 2015.

[24] R. Grau, “Requirements Engineering in Agile Software Development,” Softw. People Fundam. Trends Best Pract., pp. 97–119, 2012.

[25] S. M. Butt and W. F. W. Ahmad, “Handling requirements using FlexREQ model,” in ICSESS 2012 - Proceedings of 2012 IEEE 3rd International Conference on Software Engineering and Service Science, 2012, pp. 661–664.

[26] a Batool et al., “Comparative study of traditional requirement engineering and Agile requirement engineering,” Adv. Commun. Technol. (ICACT), 2013 15th Int. Conf., pp. 1006–1014, 2013.

[27] E. Caballero and J. A. Calvo-Manzano, “A Practical Approach to Project Management in a Very Small Company,” Syst. Softw. Serv. Process Improv. (Eurospi 2012), vol. 301, pp. 319–329, 2012.

[28] A. M. Sen and K. Hemachandran, “Elicitation of goals in requirements engineering using agile methods,” in Proceedings - International Computer Software and Applications Conference, 2010, pp. 263–268.

[29] N. Costa, N. Santos, N. Ferreira, and R. J. Machado, “Delivering user stories for implementing logical software architectures by multiple scrum teams,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, vol. 8581 LNCS, no. PART 3, pp. 747–762.

[30] L. Cao and B. Ramesh, “Agile requirements engineering practices: An empirical study,” IEEE Softw., vol. 25, no. 1, pp. 60–67, 2008.

[31] A. Batool, Y. Hafeez, S. Asghar, M. A. Abbas, and M. S. Hassan, “A scrum framework for requirement engineering practices,” Proc. Pakistan Acad. Sci., vol. 50, no. 4, pp. 263–270, 2013.

[32] B. A. Kitchenham et al., “Preliminary Guidelines for Empirical Research in Software Engineering,” Main, vol. 28, no. 8, pp. 721–734, 2002.

[33] H. Zhang, M. A. Babar, and P. Tell, “Identifying relevant studies in software engineering,” Inf. Softw. Technol., vol. 53, no. 6, pp. 625–637, 2011.

[34] J. O. Okesola, K. O. Okokpujie, A. A. Adebiyi, and C. K. Ayo, “QUALITATIVE ASSESSMENT OF SYSTEMATIC LITERATURES IN SOFTWARE ENGINEERING,” J. Theor. Appl. Inf. Technol., vol. 96, no. 18, pp. 6018–6027, 2018.

[35] Y. Zhou, H. Zhang, X. Huang, S. Yang, M. A. Babar, and H. Tang, “Quality assessment of systematic reviews in software engineering: A tertiary study,” in ACM International Conference Proceeding Series, 2015, vol. 27–29–Apri.

[36] O. J. Okesola, K. Okokpuji, P. R. O. Oyom, O. Kalesanwo, and O. Awodele, “Structuring Challenges in Requirement Engineering Techniques,” in International MultiConference of Engineers and Computer Scientists, IMECS2018, London, 2018.

[37] B. Kitchenham et al., “Systematic literature reviews in software engineering-A tertiary study,” Inf. Softw. Technol., vol. 52, no. 8, 2010.

[38] C. Patel and M. Ramachandran, Best Practices Guidlines for Agile Requirements Engineering Practices. 2010.

[39] J. Nawrocki, M. Jasiński, B. Walter, and A. Wojciechowski, “Extreme programming modified: Embrace requirements engineering practices,” Proc. IEEE Int. Conf. Requir. Eng., vol. 2002–Janua, pp. 303–310, 2002.

[40] C. Patel and M. Ramachandran, "INSERT: An Improved Story card Based Requirement Engineering Practice for Extreme Programming," in Proceedings of the 2008 International Conference on Software Engineering Research and Practice, SERP 2008, 2008, pp. 267–271.

[41] E. Bjarnason, K. Wnuk, and B. Regnell, "Are you biting off more than you can chew? A case study on causes and effects of overscoping in large-scale software engineering," Inf. Softw. Technol., vol. 54, no. 10, pp. 1107–1124, 2012.

[42] M. Kassab, "An Empirical Study on the Requirements Engineering Practices for Agile Software Development," 2014 40th EUROMICRO Conf. Softw. Eng. Adv. Appl., pp. 254–261, 2014.

[43] M. Bourimi and R. Tesoriero, "Non-Functional Requirements for Distributable User Interfaces in Agile Processes," Proc. 2014 Work. Distrib. User Interfaces Multimodal Interact. - DUI '14, pp. 1–13, 2014.

[44] J. Radhika, C. Lan, M. Kannan, and R. Balasubramaniam, "Situated Boundary Spanning : An Empirical Investigation of Requirements Engineering Practices in Product Family Development," ACM Trans. Manag. Infomation Syst., vol. 5, no. 3, pp. 1–29, 2014.

[45] N. Ikram and S. Naz, "Extreme requirements engineering (XRE)," in Communications in Computer and Information Science, 2015, vol. 558, pp. 95–108.

[46] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality Requirements in Large-Scale Distributed Agile Projects - A Systematic Literature Review," in 23rd International Working Conference on Requirements Engineering: Foundation for Software Quality, 2017, pp. 219–234.

[47] S. Wagner, D. M. Fernández, M. Felderer, and M. Kalinowski, "Requirements engineering practice and problems in agile projects: Results from an international survey," in CIbSE 2017 - XX Ibero-American Conference on Software Engineering, 2017.

[48] M. Ochodek and S. Kopczyńska, "Perceived Importance of Agile Requirements Engineering Practices – A Survey," J. Syst. Softw., 2018.

[49] M. Huo, J. Verner, L. Zhu, and M. a Babar, "Software quality and agile methods," Comput. Softw. Appl. Conf. 2004. COMPSAC 2004. Proc. 28th Annu. Int., pp. 520–525 vol.1, 2004.

[50] Z. Racheva, M. Daneva, and A. Herrmann, "A Conceptual Model of Client-driven Agile Requirements Prioritization : Results of a Case Study."

[51] N. B. Moe, A. Aurum, and T. Dybå, "Challenges of shared decision-making: A multiple case study of agile software development," Information and Software Technology, vol. 54, no. 8. pp. 853–865, 2012.

[52] M. Cardinal, "Addressing Non-Functional Requirements with Agile Practices Agile is Like Teen Sex Because …."

[53] L. Zamudio, A. A. B, and C. Tripp, "in Agile Software Development Methods," vol. 2, pp. 683–698.

[54] H. M. Sarkan, T. P. S. Ahmad, and A. A. Bakar, "Using JIRA and redmine in requirement development for Agile methodology," 2011 5th Malaysian Conf. Softw. Eng. MySEC 2011, pp. 408–413, 2011.

[55] N. A. Ernst, A. Borgida, I. J. Jureta, and J. Mylopoulos, “Agile requirements engineering via paraconsistent reasoning,” Inf. Syst., vol. 43, pp. 100–116, 2014.

[56] D. Domah and F. J. Mitropoulos, “The NERV methodology: A lightweight process for addressing non-functional requirements in agile software development,” in Conference Proceedings - IEEE SOUTHEASTCON, 2015, vol. 2015–June, no. June.

[57] J. Fisher, D. Koning, and A. P. Ludwigsen, “Utilizing Atlassian Jira For Large-Scale Software Development Management,” Proc. 14th Int. Conf. Accel. Large Exp. Phys. Control Syst., pp. 1–7, 2013.

[58] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, “A decade of agile methodologies: Towards explaining agile software development,” J. Syst. Softw., vol. 85, no. 6, pp. 1213–1221, 2012.