المجلة الدولية للحاسبات والمعلوماتية

Vol. (2), No. (4) August 2023

الإصدار (2)، العدد (4) أغسطس 2023

International Journal of Computers and Informatics (IJCI)

مجلة علمية دولية محكمة

تصدرها دار النشر

رؤية للبحوث العلمية والنشر

Vision for Scientific Research and Publishing, London, UK

المجلة الدولية للحاسبات والمعلوماتية

International Journal of Computers and Informatics

(IJCI)

مجلة علمية دولية محكمة

المجلة حاصلة على رقم تسلسلي معياري دولي: ISSN 2976-9361 (Online)

رقم Doi المجلة: https://doi.org/10.59992/IJCI.ISSN.2976-9361

موقع المجلة: https://ijci.vsrp.co.uk

البريد الإلكتروني: ijci@vsrp.co.uk

رقم التليفون (واتس): 442039115546+

تصدرها دار النشر **رؤية للبحوث العلمية والنشر**، لندن، المملكة المتحدة

Vision for Scientific Research and Publishing, London, UK

71-75 Shelton Street, Covent Garden, London, WC2H 9JQ

جميع حقوق النشر محفوظة لدار النشر رؤية للبحوث العلمية والنشر

International Journal of Computers and Informatics (IJCI)
Vol. (2), No. (4)

المجلة الدولية للحاسبات والمعلوماتية
الإصدار (2)، العدد (4)

August 2023

تقديم

عزيزي الباحث

يسعدنا في دار النشر **رؤية للبحوث العلمية والنشر** أن نقدم لكم المجلة الدولية للحاسبات والمعلوماتية IJCI وهي مجلة علمية دولية محكمة متخصصة، تهدف إلى أن تكون عوناً للباحثين العرب لتساعدهم على نشر إنتاجهم العلمي من الأبحاث، والدراسات العلمية. وتهتم المجلة بنشر الأبحاث العلمية التي يتوافر فيها الأصالة والحداثة والمنهجية العلمية والتي تشكل إضافة علمية في جميع التخصصات والعلوم باللغتين العربية والإنجليزية. وتخضع البحوث المنشورة في المجلة للتحكيم على يد نخبة من الأساتذة الأكاديميين المتخصصين من العديد من دول العالم.

تنشر المجلة الدولية للحاسبات والمعلوماتية IJCI الإنتاج العلمي في العديد من المجالات والتخصصات العلمية لإتاحة الفرصة أمام الباحثين وطلاب الدراسات العليا لنشر بحوثهم وأوراقهم العلمية. ومن أهم هذه التخصصات على سبيل المثال (وليس الحصر):

- الذكاء الاصطناعي Artificial Intelligence
- نظم التشغيل Operating Systems
- مترجمات لغات البرمجة Programming Languages Compilers
- النظم الضبابية (الفازية) Fuzzy Systems
- الشبكات العصبية Neural Network
- منهجيات هندسة البرمجيات Methodologies of Software Engineering
- هندسة المتطلبات Requirements Engineering
- المنهجيات الرشيقة لتطوير البرمجيات
  Agile Methodologies for Software Development

- البرمجة الشيئية Object-Oriented Programming
- اختبار البرمجيات Software Testing
- توكيد جودة البرمجيات Software Quality Assurance
- إدارة مشروعات البرمجيات Software Project Management
- تحليل وتصميم النظم Systems Analysis and Design
- منهجيات تطوير النظم Methodologies of Systems Development
- مشروعات نظم المعلومات Information System Projects
- قواعد البيانات Database
- أمن المعلومات Information Security
- الأمن السيبراني Cyber Security
- تنقيب البيانات Data Mining
- شبكات الحاسب Computer Network
- معالجة الصور Image Processing
- أمن الشبكات Network Security
- التعرف على الأشكال Pattern Recognition
- الرياضيات والإحصاء في مجال علوم البيانات

Mathematics and Statistics of Data Science

- طرق البرمجة في مجال علوم البيانات
- Programming Methods for Data Science
- تعلم الآلة Machine Learning
- التعلم العميق Deep Learning
- مجالات تطبيق علوم البيانات Application Domains of Data Science
- نظم المعلومات الإدارية Management Information Systems
- نظم دعم اتخاذ القرار Decision Support Systems

- نظم تخطيط موارد المؤسسة ERP
- التجارة الإلكترونية E-commerce
- التسويق الإلكتروني E-Marketing
- الحكومة الإلكترونية E-government
- التحول الرقمي Digital Transformation
- ذكاء الأعمال Business Intelligence

كما تشجع المجلة الدولية للحاسبات والمعلوماتية IJCI نشر الإنتاج العلمي في العلوم والموضوعات المتداخلة ذات الفائدة العلمية أو التطبيقية الواضحة. وهذه النوعية من الأبحاث تشمل موضوعين أو أكثر من الموضوعات المذكورة سابقاً.

نظراً لأهمية الوقت لجميع الباحثين، تتعاون المجلة الدولية للحاسبات والمعلوماتية IJCI مع مجموعة من المحررين المتميزين والمراجعين النظراء الذين لديهم الخبرة الكافية والمهارات الفنية والأدوات لتسريع عملية المراجعة والنشر قدر الإمكان. وغالباً ما تستغرق هذه العملية فترة زمنية من أسبوع إلى 3 أسابيع على الأكثر.

رئيس التحرير

International Journal of Computers and Informatics (IJCI)
Vol. (2), No. (4)

المجلة الدولية للحاسبات والمعلوماتية
الإصدار (2)، العدد (4)

August 2023

قائمة الأبحاث المنشورة بالعدد

م	عنوان البحث	اسم الباحث الجامعة، الدولة	تخصص البحث	الصفحة
1	Software Requirement Engineering: Traceability Techniques and Tools	Tarek Mohamed Nour University of Tabouk, Kingdom of Suadi Arabia Noura Abd Elrahman Albaladi University of Tabouk, Kingdom of Suadi Arabia	Software Engineering	23-8
2	Activities and Practices of Requirements Engineering in Agile Software Environments	Ahmed Jasem Kahar University of Mosul, Iraq	Software Engineering	43-24

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk

E-ISSN 2976-9361

International Journal of Computers and Informatics (IJCI)

Vol. (2), No. (4)

المجلة الدولية للحاسبات والمعلوماتية

الإصدار (2)، العدد (4)

August 2023

"Software Requirement Engineering: Traceability Techniques and Tools"

Tarek Mohamed Nour
Lecturer of Information Systems, University of Tabouk, Kingdom of Suadi Arabia
tarekmnour@yahoo.com

Noura Abd Elrahman Albaladi
Master of Information Systems, University of Tabouk, Kingdom of Suadi Arabia
albaladi_noura2@hotmail.com

Abstract:

Requirement Traceability is one of the activities in managing requirements. It is important for software projects and is affecting the quality of software products. Requirement Traceability is a method to analyze the effect of changes among various software development lifecycle parts. Agile methodologies have been presented as an alternative to traditional software engineering methodologies. The transformation between traditional and agile methodologies is a hard task so the need for traceability grows. This paper introduces traceability research at the requirement engineering on the traceability literature published during the last years. It also investigates and discusses the requirements for traceability issues. It finally presents several requirement traceability techniques and tools to support traceability.

Keywords: Requirements Traceability, Agile Software System, Requirements Traceability Tools.

1- Introduction

This research has focused on requirements traceability, which aims to study how to describe and follow the life of a requirement, in both forward and backward directions [1][2]. Many researchers have participated in the area for the last two decades [3][4][5], to provide solutions in the form of methods, tools, and a better understanding of traceability needs and challenges. The purpose of this paper is to review the development in traceability research of requirement engineering (RE) and the tools used to enhance traceability [6].

The traceability of software artifacts is considered an important factor in supporting various activities in the development process of a software system. In general, the objective of traceability is to improve the quality of software systems. More specially, traceability of information can be used to support various activities such as the change impact analysis, software maintenance, and evolution, and the reuse of software artifacts by identifying and comparing the requirements of the new system with those of the existing system.

The goal of software traceability is to discover relationships between software artifacts to facilitate the efficient retrieval of relevant information, which is necessary for many software engineering tasks.

Implementing traceability helps in conducting important tasks, such as the evaluation of how changes in an element may impact other parts of the system. Software maintenance is one of the core goals of implementing traceability in a software system.

This paper is organized as follows: Section (2) presents the definition of requirement traceability along with its types. Section (3) presents the related works through three parts (a) review Traceability in traditional software, (b) review Traceability in Agile

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

software systems, and(c) tools used for traceability. Section (4) concludes the paper and outlines future research directions to enhance RS traceability.

2- Requirement Traceability

Gotel and Finkelstein stated that "Requirements traceability refers to the ability to describe and follow the life of a requirement, in both forward and backward directions within software development process (i.e., from requirements, design, implementation, to testing and maintenance)" [2].

Pinheiro and Goguen state that "Requirements traceability refers to the ability to define, capture and follow the traces left by requirements on other elements of the software development environment and the traces left by those elements on requirements" [7]. Wieringa divided traceability into two groups in 1995 Forward and Backward Traceability [8] [9].

Forward traceability is the ability to trace a requirement to components of a design or implementation. Backward traceability is the ability to trace a requirement to its source, i.e., to a person, institution, law, argument, etc. While Lindval and Sandah decomposed traceability into two groups Vertical and Horizontal traceability as in [10].

Vertical traceability is tracing dependent items within a model. Horizontal traceability is tracing correspondent items between different models.

Backward and forward traceability are like horizontal traceability. Most of the research on requirements traceability was concentrated on horizontal traceability because it is more accurate than vertical traceability.

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

Pinheiro divides traceability into two groups [9]:

- Inter-requirements traceability refers to the relationships between requirements. Inter-requirements traceability is important for requirements change and evaluation. For example, when extracting all requirements derived from a specific requirement or its chain for refinement.
- Extra-requirements traceability refers to the relationships between requirements and other artifacts.

Inter-requirements traceability is like vertical traceability. There are three features that should be covered by a traceability model [9]: First Definition: The definition is related to the specification of the traces and traceable objects. Second Production: The production is related to the capture of traces, usually by means of an explicit registration of the objects and their relationships. Third Extraction: The extraction is related to the actual process of tracing, i.e., the retrieval of registered traces [7].

3- Related Work

3-1 Traceability in Traditional Software Development Process

In this section, all the manual traceability techniques are mentioned such as cross reference, document, and structured centered. Traceability is very hard since the requirements keep on changing throughout the lifecycle of software. The basic techniques to handle this process are mentioned with their small description in the table given below.

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

Table1. Manual traceability techniques

Technique	Description
1. Cross-reference centered.	In this technique documents which keep online forms are supported automatically by linking ends of cross-reference hyper textually. For example: forms of explicit requirements such as (tagging, numbering, or indexing) [2].
2. Document centered.	In this technique, requirements can be traced by describing either all or part of the content of the project documentation.
3. Structure centered.	In this technique helps in completing the requirement traceability by restructuring the document in the form of a graph or network.

The above techniques provide early feedback from the customers by which the analyst team gets more time to respond to the changes in requirements. Another important benefit is that the verification process becomes easier to implement. The graph or network view of a particular project which makes the requirement as a graphical representation is easy to implement.

3-2 Traceability in Agile Information Systems

Agile methodology enables the organization to deliver quickly. Change quickly. Change often [18]. While agile techniques vary in practices and emphasis, they follow the same principles behind the agile manifesto [36]:

- Working software is delivered frequently (weeks rather than months).
- Working software is the principal measure of progress.
- Customer satisfaction with rapid, continuous delivery of useful software.
- Even late changes in requirements are welcomed.
- Close daily cooperation between business people and developers.
- Face-to-face conversation is the best form of communication.

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

- Projects are built around motivated individuals, who should be trusted.
- Continuous attention to technical excellence and good design.
- Simplicity.
- Self-organizing teams.
- Regular adaptation to changing circumstances.

Agile development methods have been designed to solve the problem of delivering high-quality software on time under constantly and rapidly changing requirements and business environments.

The initial problem was how to add traceability in agile methods such as Scrum and Extreme Programming (XP). In Scrum [20], [21], a single person in the role of a product owner (PO) is responsible for requirements elicitation and requirements prioritization. Requirements in Scrum reside in a product backlog, which is a prioritized list of all work items imagined for the software, which also can include technical improvements. The work items in the product backlog are called backlog items. Only the product owner can add new items to the backlog. The product owner works with a development team of five to nine cross-functional software developers. The PO and the development team conduct requirements analysis, requirements specification, and requirements validation informally and in collaboration. At the beginning of each two to four-week development iteration, based on the development team's previous performance, the PO and the development team decide which backlog items are implemented during the iteration. At the end of the iteration, the system validation is done by the PO, who reviews the new system behavior. Extreme Programming (XP) [38] concentrates on software construction and there is little guidance for requirements engineering. The actors in XP are an on-site customer and a team of three to twenty developers. The main RE practice in XP is the planning

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

game, which begins with on-site customer writing requirements. Thereafter, the on-site customer and developers decide which of the requirements are to be implemented during the following two-week development iteration. The implemented requirements are also validated by the on-site customer.

Next, we present a summary of traceability modeling:

Cleland-Huang et al. [14] propose a Traceability Information Model (TIM). It creates traces between acceptance tests and user stories. RT is constructed by inserting a cross-reference to one or more user stories into each acceptance test. When the test cases are executed and passed, RT links are automatically created between the source code and test cases.

Taromirad et al. [15] proposed Domain-Specific Requirements Traceability (DSML) Used to build a traceability scheme for a specific domain or project.

Badreddin et al. [16] proposed Requirement Oriented Modeling and Programming Language (ROMPL) that supports the automatic generation and maintenance of the RT links between requirements, models, and code.

Ratanotayanon et al. [17] used a tool, namely Zelda Work with an agile software development process that captures and maintains links between high-level information and source code.

Espinoza et al. [18] proposed a traceability meta-model (TmM) that Supports creating RT by allowing user-defined RT links, well-defined roles, and linkage rules and specifying what kinds of RT links can and must be created.

Cleland-Huang et al. [19] used the Traceability (JITT) tool Which Provides an interactive domain for retrieving relevant code. The tool returns a list of elected classes and shows the relation between the retrieved class and the user story.

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

Duraisamy et al. [20] used RTM as a two-dimensional array that shows items in rows and columns. It is a technique to create RT links of requirements between product backlog and sprint backlog where the information is retrieved by using keyword searching functionality.

3-3 Tools for Traceability Management

Innoslate[1] [27] is a requirements management tool that allows import of requirements from other tools. By underlying model-based database, these relationships are automatically generated into Hierarchy Charts, Traceability Spider Diagrams, or 3D Traceability Diagrams in which requirements can edit requirements, or new requirements are created directly in these diagrams.

Trace Maintainer [27] [28] [29] uses a rule engine and modifier for maintenance of links. It accepts change events in the case tool and provides element properties to the rule engine to update the traceability relations. The disadvantage of Trace Maintainer is that it does not support different types of traceability links, so the modifier should be written for the integration of this tool with any specific case tool.

Doors from IBM [21] [26] [27]is a requirements management solution that allows changing the attributes of the traceability links, gives the possibility to produce different types of traceability links between the artifacts, and stores all the documents that can be excel sheets, word files, and other rough documents into a centralized database.

ADAMS Re-Trace [27] [30]: is a Traceability Recovery Tool that compares the links retrieved by Latent Semantic Indexing (LSI) Information Retrieval with the links

[1] https://www.innoslate.com/requirements-management/

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

traced by the software engineer. If there are any contradictions the links are highlighted and built within the Advanced Artefact Management System (ADAMS).

Trust Analyzer tool [27] [31] establishes the traceability between scenarios which can exist in the form of plain English or diagrams and the source code. When applying the scenarios the internal activities of the system can be observed and recorded.

Rational Requisite Pro[2][27] is another tool from IBM that helps the management of requirements and allows traceability of one requirement to another. It can also be integrated with Rational ClearQuest Test Manager which manages all the test cases.

TraceM [27] [32]is a framework for automating the management of traceability relationships. It provides registration, integration, evolution and querying services. Information Integration and Open Hypermedia services are utilized by TraceM for Traceability Management. Open Hypermedia allows the storing of relationships separately from the artifacts. Information integration provides the services for creating and maintaining evolving relationships between artifacts. Confluence[3] is open and collaborative, helping you create, manage, and collaborate on anything from product launch plans to marketing campaigns. Find work easily with dedicated and organized spaces, connect across teams, and integrate seamlessly with the Atlassian suite or customize with apps from our Marketplace.

REquirements Tracing On target (RETRO) [27] [33] is a tool for tracing requirements. It uses Information Retrieval methods (Vector Space Retrieval, Latent semantic indexing, and Keyword word extraction methods). The IR method is

[2] http://www-01.ibm.com/support /knowledgecenter/SSRTLW_7.5.5/com.ibm.xtools.reqpro.doc/topics/c_trace.html

[3] https://www.capterra.com/p/136446/Confluence/

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

executed using reweighted query vectors. The methods are continuously enhanced with user feedback processing.

CRADLE[4][27] is a requirements management tool that enables traceability of items. Requirements can be imported from various sources like Excel and Word files and the traceability among different items can be established using the tool. The traceability and coverage of information can be analyzed using tables, matrices, and graphical hierarchy diagrams.

Trustrace proposed by Ali, Nawazish [27] [34], is a traceability recovery approach between requirements and source code using data mining. Trustrace uses a combination of both the Information Retrieval Method and Data mining to establish traceability links between requirements and source code. Trustrace is found to have more precision when compared to tools that use only Information retrieval (IR) methods.

End to End Software Traceability tool is proposed by Asuncion, Hazeline [27] [35]. The tool follows three tiers architecture. This is the only tool that deals with end-to-end traceability of artifacts and was implemented for an organization.

Many tools are designed to help companies to test and manage their projects (https://www.softwaretestinghelp.com/software-testing-trends).

4- Conclusion and Future Work

The main goal of this paper is to investigate what is meant by traceability and how to trace the requirement. The topic highlighted is the study of many techniques used by different researchers. This research also draws some attention to the required traceability tools that are used to manipulate it.

[4] https://www.threesl.com/cradle/index.php

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

The idea of tracing requirements presented in this paper has many opportunities for further expansion and research. The researchers in the traceability community established a roadmap [22] [23], and identified several challenges for traceability, including the Grand Challenge [24] to achieve Ubiquitous Traceability. These challenges are summarized as goals in Table (2). Each goal represents a required quality of traceability and is transformed to a set of research topics (described in detail in [22] [23]).

Table2. A Goal-Oriented Perspective [22] [37]

Goal Identified	Goal Description
Goal 1: Purposed	Traceability fit for purpose and should support stakeholder needs. So, it must be defined clearly for systems engineering tasks.
Goal 2: Cost-effective	Develop techniques for computing the return on investment (ROI) of traceability in a project. Supporting the effect of traceability decisions at different stages of the system life cycle.
Goal3: Trusted	Develop techniques for evaluating the state of traceability in a project so all the stakeholders can depend on the provided traceability.
Goal 4: Configurable	Develop techniques for dynamically generating and maintaining trace links that are configured according to project needs.
Goal 5: Scalable	Varying types of artifacts can be traced. develop techniques for scaling up traceability and for supporting multi-grained traceability across a variety of artifact types.
Goal 6: Portable	Traceability is changed and reused across projects and organizations so policies, standards, and formats must be developed for change and integrate traceability information.
Goal 7: Valued	Develop supporting techniques that bridge the technical and business domains of a project, so the benefits of traceability are visible and accessible to all stakeholders.
Goal 8: Dataset	availability of traceability datasets and benchmarks
Goal 9: Applications	real-world applications of traceability

Today, we have over 16 datasets available for community use from our CoEST.org website. The researchers designed an online Research Directions forum at "CoEST.org." this website has over 16 datasets available for community use and provides downloadable, executable, baselined experiments, developed in Trace-Lab

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

[22] [25] [37]. Trace Lab is a different experimental environment which allows researchers to reuse, reproduce, and/or modify previous experiments, compose new experiments from a combination of existing and user-defined components, use publicly available datasets, exchange components, and relatively evaluate results against previous benchmarks. They provide links to these experiments from the Research Directions forum to encourage evaluation and generate the experimental results [22] [25].

Trace Lab is designed to enable future traceability research, by facilitating collaboration and creativity between researchers, decreasing the startup costs and effort of new traceability research projects, and encouraging technology transfer. Trace Lab has been released via CoEST.org in the summer of 2012. In addition, by late 2012 Trace Lab's source code was released as open-source software, licensed under GPL. Trace Lab currently runs on Windows but is designed to port to Linux and Mac environments [25].

Industrial and governmental organizations still work with tasks that could be reduced by more ubiquitous use of traceability – and industrial practice wants more tools and techniques from the research community [37].

The researchers should share in the ongoing discussion, to keep the community informed of important advances and new challenges as they grow, and where possible use existing experimental baselines or create new ones for others to use. This collaboration works will assist in achieving the ubiquitous traceability goal [22] [25].

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

References

[1] O. Gotel & Anthony" An Analysis of the Requirements Traceability Problem", C. W. Finkelstein,1993.
[2] O. Gotel and A. Finkelstein, "An Analysis of the Requirements Traceability Problem," in Proceedings Of 1st International Conference on Requirement Engineering, 1994, pp. 94-101.
[3] G. Spanoudakis and A. Zisman, "Software traceability: a roadmap", in Handbook of Software Eng. and Knowledge Engineering, 2005.
[4] R. Torkar, et al., "Requirements traceability: a systematic literature review and industry case study", International Journal of Software Engineering and Knowledge Engineering, vol. 22, no. 3, pp. 1-49, 2012.
[5] O. Gotel, et al., "The Quest for Ubiquity: A Roadmap for Software and Systems Traceability Research", RE 2012, pp.71-80.
[6] M. Narmanli, "A Business Rule Approach to Requirements Traceability", Sept. 2010.
[7] F. Pinheiro and J. Goguen, "An Object-Oriented Tool for Tracing Requirements," IEEE Software, vol. 13, no. 2, pp. 52-64, March 1996.
[8] R.J. Wieringa, "An Introduction to Requirements Traceability," Faculty of Mathematics and Computer Science, University of Vrije, Amsterdam, September 1995.
[9] Francisco A. C. Pinheiro, "Requirements Traceability" in Perspectives on software requirements, Jorge Horacio Doorn, Ed.: Springer, 2003, Ch. 5, pp. 91-113.
[10] M. Lindval and K. Sandahl, "Practical Implications of Traceability," Software Practice and Experience, vol. 26, no. 10, pp. 1161-1180, 1996.
[11] B. Ramesh and M. Jarke, "Towards Reference Models for Requirements Traceability," IEEE Transactions in Software Engineering, vol. 27, no. 1, pp. 58-93, 2001.
[12] A. Ghazarian, 2008, "Traceability Patterns: An Approach to Requirement-Component Traceability in Agile Software Development", Proceedings of the 8th WSEAS International Conference on Applied Computer Science, pg.: 236-241.
[13] M. Jacobsson "Implementing Traceability in Agile Software Development", 2009-02-02.
[14] J. Cleland-Huang, O. Gotel, and A. Zisman, "Software and systems traceability". Springer,2012, vol. 2, no. 3.

E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

[15] M. Taromirad and R. F. Paige, “Agile requirements traceability using domain-specific modelling languages,” in Proceedings of the 2012 Extreme Modeling Workshop, 2012, pp.45-50.

[16] O. Badreddin, A. Sturm, and T. C. Lethbridge, “Requirement traceability: A model-based approach," in Model-Driven Requirements Engineering Workshop (MoDRE), 2014 IEEE 4th International. IEEE, 2014, pp. 87-91.

[17] S. Ratanotayanon, S. E. Sim, and R. Gallardo-Valencia, “Supporting program comprehension in agile with links to user stories," in Agile Conference, 2009. AGILE'09. IEEE, 2009, pp.26-32.

[18] B. Arbain, A. Firdaus, I. Ghani, W. Kadir, and W. M. Nasir, “Agile non-functional requirements (NFR) traceability metamodel," in Software Engineering Conference (MySEC),2014 8th Malaysian. IEEE, 2014, pp. 228-233.

[19] J. Cleland-Huang, B. Berenbach, S. Clark, R. Settimi, and E. Romanova, “Best practices for automated traceability," Computer, no. 6, pp. 27-35, 2007.

[20] G. Duraisamy and R. Atan,” Requirement traceability matrix through documentation for scrum methodology." Journal of Theoretical & Applied Information Technology, vol. 52, no. 2, pp. 154-159, 2013.

[21] M. Omar and J. Dhar,” A Systematic Literature review of traceability Practices for Managing Software Requirements”, Journal of Engineering and Applied Science 12 (Special Issue 4), Medwell Journals 2017

[22] J. Cleland-Huang, O. Gotel, P. Mäder, A. Zisman, and J. Huffman Hayes “Software Traceability: Trends and Future Directions, ICSE ’14 Hyderabad, India Copyright 2014 ACM 978-1-4503-2865-4/14/05.

[23] O. Gotel, J. Cleland-Huang, J. Huffman Hayes, A. Zisman, A. Egyed, P. Grunbacher, and G. Antoniol. The quest for ubiquity: A roadmap for software and systems traceability research. In 21st IEEE International Requirements Engineering Conference (RE), pages 71 -80, 2012.

[24] O. Gotel, J. Cleland-Huang, J. Huffman Hayes, A. Zisman, A. Egyed, P. Grunbacher, A. Dekhtyar, G. Antoniol, and J. Maletic. The grand challenge of traceability (v1.0). In J. Cleland-Huang, O. Gotel, and A. Zisman, editors, Software and Systems Traceability, pages 343{409. Springer, 2012.

E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

[25] E. Keenan, A. Czauderna, G. Leach, J. Cleland-Huang, Y. Shin, E. Moritz, M. Gethers, D. Poshyvanyk, J. Maletic, J. Huffman Hayes, A. Dekhtyar, D. Manukian, S. Hossein, and D. Hearn. Trace lab: An experimental workbench for equipping researchers to innovate, synthesize, and comparatively evaluate traceability solutions. In Tool Demo,34th International Conference on Software Engineering (ICSE), pages 1375-1378, 2012.

[26] M. Taromirad, and R. F. Paige. "Agile Requirements Traceability Using Domain-Specific Modelling Languages". Proceedings of the 2012 Extreme Modeling Workshop on –XM '12 (2012).

[27] Satish C J, Anand M, and Thendral Puyalnithi," A Review of Tools for Traceability Management in Software Projects ", International Journal for Research in Emerging Science and Technology, Volume-3, Issue-3, Mar-2016.

[28] Mäder, Patrick, Orlena Gotel, and Ilka Philippow."Enabling automated traceability maintenance through the upkeep of traceability relations." Model Driven Architecture-Foundations and Applications. Springer Berlin Heidelberg, 2009.

[29] Mäder, Patrick, et al. "trace Maintainer-Automated Traceability Maintenance." International Requirements Engineering, 2008. RE'08. 16th IEEE. IEEE, 2008.

[30] Lucia, Andrea D., et al. "Adams re-trace: A traceability recovery tool. "Software Maintenance and Reengineering, 2005. CSMR 2005. Ninth European Conference on. IEEE, 2005.

[31] Alexander E. "scenario-driven approach to traceability." Proceedingsof the 23rd international conference on Software engineering. IEEE Computer Society, 2001.

[32] Sherba, Susanne A., Kenneth M. Anderson, and Maha Faisal."A framework for mapping traceability relationships." Proceedings of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering. 2003.

[33] Sundaram, Senthil Karthikeyan, et al. "Assessing Traceability of software engineering artifacts." Requirements engineering 15.3 (2010): 313-335.

[34] Ali, Nawazish, Yann-Gael Gueneuc, and Giuliano Antoniol. "Trustrace: Mining software repositories to improve the accuracy of requirement traceability links." Software Engineering, IEEE Transactions on 39.5 (2013):725-741.

[35] Asuncion, Hazeline U., Frédéric François, and Richard N. Taylor. "An end-to-end industrial software traceability tool." Proceedings of the 6th joint meeting of the European software

E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering. ACM, 2007.

[36] Appleton, B. ACME Blog: Traceability and TRUST-ability. http://bradapp.blogspot.com/2005/03/traceability-and-trust-ability.html (2005, Tuesday, 15 March). Accessed June 2011.

[37] G. Antonio, J. Cleland- Huang, J. Hayes, M. Vierhauser," Grand Challenges of Traceability 2017", Cornel University.

[38] K. Beck and C. Andres. Extreme programming explained: embrace change. Addison-Wesley, Boston, MA, USA, 2nd edition, 2004.

E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p1

International Journal of Computers and Informatics (IJCI)

Vol. (2), No. (4)

المجلة الدولية للحاسبات والمعلوماتية

الإصدار (2)، العدد (4)

August 2023

"Activities and Practices of Requirements Engineering in Agile Software Environments"

Ahmed Jasem Kahar

Master of Information Systems, University of Mosul, Iraq

ahmed.kahar@uomosul.edu.iq

Abstract:

Requirements are one of the important factors for software success. However, Requirements Engineering (RE) activities, in the Waterfall process model, are done sequentially in the analysis phase, which makes it difficult when RE practitioners think and reason about them in Agile Software Development (ASD) process model. RE practitioners need to find the established RE activities conventions in the ASD process model, especially with an increase of software companies, which transform to the ASD process model, in order to foster their transition to the new model. The objective of this study is to provide RE activities in ASD to allow RE practitioners to utilize the appropriate activity for specific ASD methodology. RE activities in ASD are handled repetitively and on a small scale, which makes them embedded in the development life cycle. This paper focuses on highlighting them in different ASD methodologies and discusses the practices that resolve the traditional Waterfall model issues. Although the ASD model has resolved some traditional RE issues, it introduced other issues, such as a consequence of trying to achieve an adequate balance between agility and stability. In addition, there is a lack of practices that target non-functional requirements.

Keywords: Requirements Engineering, Requirements Elicitation, Requirements Documentation, Requirements Validation, Agile Requirements.

E-ISSN 2976-9361

https://doi.org/10.59992/IJCI.2023.v2n4p2

1- Introduction

It is widely acknowledged in the literature that requirements are critical for the success of software projects. Requirements Engineering (RE) is an essential part of the software development process, especially in the system analysis phase. It is concerned with improving knowledge acquisition and knowledge sharing that allows a more complete understanding of the application domain constraints and stakeholder needs [1]. However, RE activities are done differently in Waterfall against Agile Software Development (ASD) process models.

RE activities in the Waterfall model are done sequentially in the analysis phase. Thus, it is difficult when RE practitioners think and reason about them in ASD; as a separated phase, especially with an increase of software companies, which transform to ASD process model [2]. On the other hand, in ASD, RE activities are integrated at the whole development process and done iteratively. We mention RE practitioners instead of system analysts, project managers, or requirements engineers, as in ASD, the multifunctional team, with one person acting in more than one role. The need for agile RE has been raised in the literature as [1] mentioned, in order to better improve RE "software engineering community will likely implement three distinct sub processes as RE is conducted: (1) improved knowledge acquisition and knowledge sharing that allows a more complete understanding of application domain constraints and stakeholder needs. (2) Greater emphasis on iteration as requirements are defined, and (3) more effective communication and coordination tools that enable all stakeholders to collaborate effectively." [1].

ASD practices have resolved some of RE challenges that were faced in the Waterfall model [3]. These challenges are (1) communication issues, (2) overscoping, (3) requirements validation, (4) requirements documentation, and (5) rare customer involvement [3]. Communication issues are solved using practices such as, face-to-

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

face communication, collocated teams, on-site customer or alternate customer representations, and integrated RE process in ASD. Overscoping is reduced by gradual detailing of requirements and participation of cross-functional teams. Requirements validation is met by prototyping, which helps in providing the customer with a blueprint of the product, and therefore helps in requirements validation. Requirements documentation, which is characterized by long Software Requirements Specification (SRS) documents, is reduced by face-to-face communication and user stories. User stories are precise and provide a to-the-point explanation of user needs. Finally, a rare customer involvement challenge is met by a requirements prioritization practice by the customer for all iterations which ensures that the customer goals will be met [3].

RE practitioners need to find the established RE activities in the ASD process model, in order to foster their transition to the new model. The objective of this study is to provide an RE activities mapping in ASD to allow RE practitioners to utilize the appropriate activity for specific ASD methodology. Section II introduces background and RE activities. Section III gives relevant work. Section IV illustrates the RE activities mapping in ASD methodologies. Section V gives a discussion on the impact of ASD on RE activities. Finally, section VI gives the conclusion.

2- Background

This section introduces a background for our research. In the Waterfall process model, RE activities include a set of ordered processes for capturing, gathering, documenting, and validating requirements regarding the users' needs and demands. Next, we introduce them and the goal of each one. We start with requirements elicitation, analysis, documentation, validation, and finally, management. They are well-established in the traditional Waterfall model, as have been revolutionized in

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

ASD model in a manner that achieves the manifesto for agile; values and principles [4].

2-1 Requirements Elicitation

It is the first activity in the RE process. Through which the requirements of a system are discovered and elaborated through consultation with stakeholders, from previous documents, and domain knowledge. During this activity, the boundary for the proposed system is defined. Also referred to in the literature as requirements acquisition. Requirements elicitation is the process through which the requirements specification is derived [5].

There are different requirements elicitation techniques, which could be organized into 7 categories. They are traditional, collaborative, prototyping, modeling, cognitive, contextual, and agile techniques [6]. Table 1 illustrates each category with its techniques.

Table 1. Requirements Elicitation techniques categorized.

Technique Category	Technique's Name
Traditional	Interviews, surveys, task analysis, and questionnaires.
Collaborative	Focus groups, workshops, and brainstorming.
Prototyping	Prototyping.
Modelling	Scenarios, goal-based approaches, business process models, and use cases.
Cognitive	Ontology, card sorting, and repertory grid.
Contextual	Ethnography and ethnomethodology.
Agile	Mind mapping, user stories, and group storytelling.

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

2-2 Requirements Analysis

It is the activity which is concerned with reaching a richer and more precise understanding of each requirement and representing them in multiple ways. It is used to get a better understanding of the whole business and to check if the elicited requirements are consistent, complete, and feasible. Sometimes, during these activities, the requirements can be modeled to make them clearer for the developers. It consists of analyzing the information which elicited from users to identify their task goals and classify them to functional and non-functional requirements [7]. Analysis techniques such as Joint Application Development (JAD), requirements prioritization, and modeling.

2-3 Requirements Documentation

It is the activity that results in producing the requirements specification, which is the output of the RE process. There is a wide variety of ways for expressing a requirements specification, ranging from informal natural language to more formal graphical and mathematical notations [8].

2-4 Requirements Validation

It is the activity through which possible problems in the requirements specification are detected before the specification is used for development. The validation checks if the requirements statements are valid, not contradictory, and if they satisfy the customer's needs. Test cases are used in this phase to discover the ambiguities and vagueness of written requirements. The requirements specification is validated to ensure its accuracy, consistency, and relevance.

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

2-5 Requirements Management

Requirements management supports all RE activities. It is an activity which is concerned with requirements versioning and control, traceability, and change. The traceability is between requirements, or between requirements and other software elements, such as features. In addition to, tracking requirement status [7].

3- Related Work

This section presents the related studies for agile RE. Schön et al. [9] introduced a Systematic Literature Review (SLR) for agile RE. The study focuses on approaches that target the stakeholder involvement in the process. The authors concluded that there is a lack of building a shared understanding of the user perspective in ASD. They identified four methodologies that were integrated into ASD to increase the understanding of user needs. These methodologies are Human-Centered Design, Design Thinking, Contextual Inquiry, and Participatory Design.

Liskin et al. [10] aim to understand how requirements artifacts are used for daily work in Kanban. They concluded that the communication with stakeholders enhanced as artifacts helps in mitigating the misunderstandings between the participants. In addition, the study mentioned collaboration challenges that arise when artifacts are too detailed. However, it is difficult to generalize their findings as their study was a qualitative study that may reflect subjective opinions.

4- RE activities in agile methodologies

RE activities, described above, have different techniques to achieve their objectives. RE practitioners used to handle them in order in the Waterfall model. However, in the ASD model RE activities have two important characteristics; they are done in a small scale and iteratively. Following is our survey of the RE activities in ASD

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

methodologies. First, we list the most adapted ASD methodologies in companies, according to the annual State of Agile report [2].

Secondly, we mention RE activities in each one. Figure 1 illustrates a survey results for the most adapted ASD methodologies, as mentioned Scrum is the most adapted methodology, then ScrumBan, which is the combination of Scrum and Kanban (a lean methodology), then Kanban, and eXtreme Programming (XP). XP practices partially involved in the hybrid part of the report, as well as in other methodologies, which is the reason we specify it here.

4-1 Scrum

Scrum is a wide-spread, adapted methodology, due to the fact that it is light weight and could be adapted easily. The Scrum is based on a set of values, which is achieved by a set of principles and practices. These principles and practices supply the foundation to engineering approaches for the Scrum practices implementation [11]. Figure 2. shows the practices. Requirements elicitation in Scrum, which concerned capturing domain knowledge and user needs, has been spread throughout the whole development process.

First in Product backlog preparation, when the Product Owner understands the needs and priorities of the organizational stakeholders, the customers, and the users well enough to act as their voice. Second, during sprint planning and after development during the sprint review/release review with Product Owner or the end users. The used elicitation techniques could be varied or mixed between traditional, collaborative, and agile techniques [6]. Scrum does not restrict specific techniques; however, it establishes a framework to work in.

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

International Journal of Computers and Informatics (IJCI)

Vol. (2), No. (4)

المجلة الدولية للحاسبات والمعلوماتية

الإصدار (2)، العدد (4)

August 2023

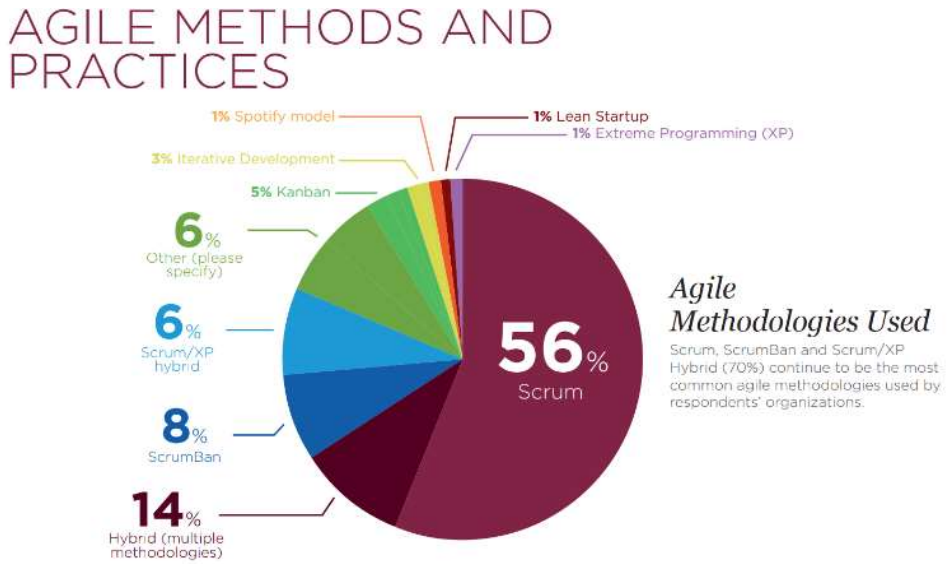

Figure 1. Agile methodologies usage survey [2].

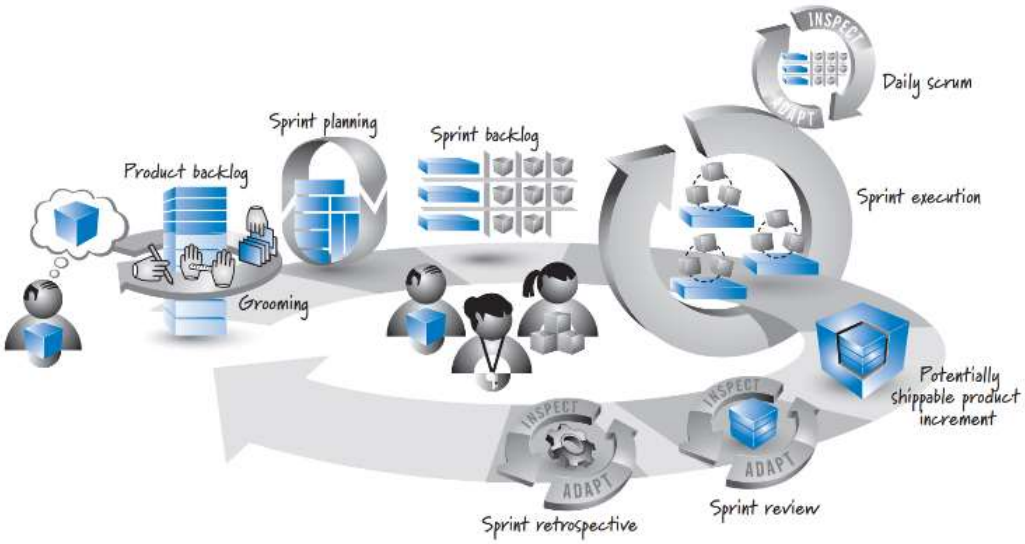

Figure 2. Scrum methodology [11].

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk

E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

Requirements analysis activities in Scrum could be found in Product Owner analyses the feasibility of the requirements, Backlog Refinement meeting, and Product Owner prioritizes the Product Backlog. Requirements Documentation is handled by face-to-face communication, as well as by writing user stories. Requirements Validation is done by Review meetings, which is more effective validation by working software.

Requirements Management activities will be found in iterative RE, short releases, and customer-feedback [12]. This could especially be elaborated via Product Backlog tracking (by changing the requirements (added/deleted) to/from Product Backlog), and Sprint Planning meetings. Table 2 summarizes RE implementation in Scrum [13].

4-2 Extreme Programming

XP is one of the earliest agile methodologies, which focuses on technical practices, and well-documented methodologies [14]. It aims to produce higher quality software, as well as quality of life for the development team. XP has twelve rules, namely: Planning Game, Small Releases, Metaphor, Simple Design, Tests, Refactoring, Pair Programming, Continuous Integration, Collective Ownership, Onsite Customer, 40-Hour Weeks, and Open Workspace. In addition to these practices, XP suggests a development life cycle as shown in Figure 3 [15]. XP project [15]. One important rule in XP is On-Site Customer, which recommends that the development team must have someone from the customer side [14].

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

Table 2. RE Implementation in Scrum.

RE Activity	Scrum Implementation
Requirements Elicitation	Product Owner formulates the Product Backlog, stakeholders' participation in the Product Backlog preparing and Sprint Review meeting.
Requirements Analysis	Backlog Refinement meeting (grooming), Product Owner prioritizes the Product Backlog, Product Owner analyses the feasibility of requirements.
Requirements Documentation	Face to face communication.
Requirements Validation	Review meetings.
Requirements Management	Sprint Planning meeting, Product Backlog tracking by change requirements (added/deleted) to/from Product Backlog.

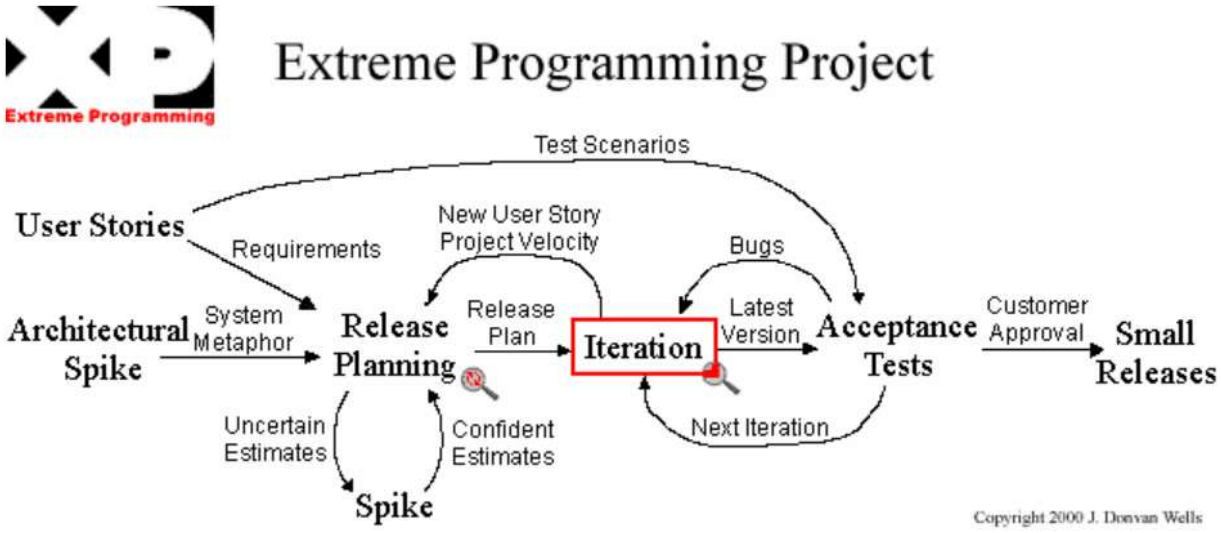

Figure 3.Extreme Programming Project [15].

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

Requirements elicitation and analysis in XP happen iteratively as there is a customer representative in the team, On-Site Customer. First in the Planning Game rule, where at the beginning of each iteration the User Stories' writing activity and Release Planning (see figure 3) occur. The planning starts to define, estimate and prioritize the User Stories, as a requirement elicitation artifact, for the next release [14]. Requirements Documentation realized through face-to-face communication and User Stories. User Stories in XP is for facilitation rather than documentation purposes. However, requirements management are not specified, as the User Story cards are destroyed after implementation [16].

4-3 Kanban

Kanban is an agile software development methodology based on Lean software development, which aims to minimize waste. Lean software development focuses on seven principles. They are: 1) eliminate waste, 2) amplify learning, 3) decide as late as possible, 4) deliver as fast as possible, 5) empower the team, 6) build integrity in, and 7) see the whole. Kanban emphasizes on "just-in-time" delivery [17]. It prioritizes tasks and defining workflow as well as required time to delivery [18]. The word "Kanban" is a Japanese word means visual work. Kanban has main principles: visualize workflow, limit work in progress, and measure and manage Flow [19].

Kanban board provides a visualization of the requirements' progress through the development workflow, which could be used as a requirement management tool. Figure 4 shows an example of User Stories visualized in Kanban [20]. Requirement management, especially requirement tracing was done through the Kanban board. However, there are no specific artifacts or activities for other RE activities.

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

International Journal of Computers and Informatics (IJCI)

Vol. (2), No. (4)

IJCI

August 2023

المجلة الدولية للحاسبات والمعلوماتية

الإصدار (2)، العدد (4)

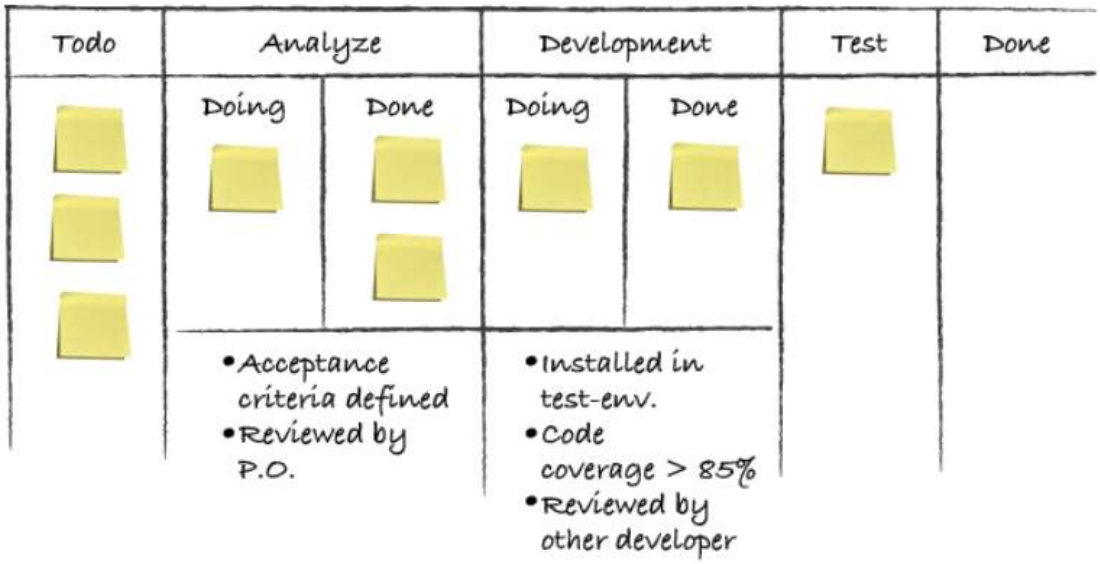

Figure 4. Kanban board visualizes the requirement's progress [20].

In spite of Kanban and Scrum are different ASD methodologies, the development teams may combine activities or techniques from Kanban into Scrum. For instance, Kanban board is used by Scrum team to track and monitor User Stories' progression. In addition, Kanban board helps in minimizing the work in progress, which allows the development team to focus on Sprint items and deliver the working software as planned in the Sprint Planning Meeting. However, the development teams acknowledge the difference between the two methodologies, such as the delivery cadence/rate. Scrum teams commit to deliver working software at the end of each Sprint, while Kanban teams deliver software continuously [21].

5- Discussion

RE activities are crucial for the software requirement development. They formalize and organize the required work to produce good requirements. However, they are

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk

E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

challenging, especially when RE practitioners transformed from Waterfall to ASD model. ASD practices have resolved some of RE challenges in the Waterfall model such as communication issues, and rare customer involvement. The solution of the previous challenges has been done by adapting the list of practices in different ASD methodologies.

These practices were mentioned in [3], they are:

- Face-to-face communication.
- Customer involvement.
- User stories.
- Iterative requirements.
- Requirements prioritization.
- Change management.
- Cross-functional teams.
- Prototyping.
- Testing before coding.
- Requirements modelling.
- Review meetings and acceptance tests.
- Code refactoring.
- Shared conceptualizations.
- Pairing for requirements analysis.
- Retrospectives.
- Continuous planning.

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

ASD methodologies are varying regards to the implementation of RE activities. Scrum methodology is the one that most covered RE activities. XP emphasis on technical practices, and there is not a clear requirements management established practices. Kanban focuses on visualizing work, as this is a good approach for requirement management and tracking.

Requirements elicitation in Scrum methodology is done during the Product Backlog preparation, or Review meeting. Product Owner collaborates with Business Analyst/Developer, to elicit and elaborate the requirements. As well during the sprint, Developer could request clarification for more details from Product Owner, as onsite customer reprehensive. While in XP, Planning Game practice, where elicitation techniques like interviews, brainstorming and prioritization are used. Kanban, as following Lean principles, postpone elicitation and analysis until the last responsible moment, i.e., before requirement development/realization to working software. Table 3 summarizes and compares requirements elicitation in agile methodologies.

Requirements analysis in Scrum could be applied during prioritizing Backlog items, and Backlog Grooming, Where Product Owner prioritizes the Product Backlog, and analyses the feasibility of requirements. While in XP, the Planning Game practice, where analysis techniques like JAD, brainstorming, and prioritization are used. Together software developers and onsite customers move the user story cards around on a large table to create a set of stories to be implemented as the first (or next) release. Table 4 summarizes and compares requirements analysis in agile methodologies.

Requirements documentation in ASD methodologies has been eliminated. Scrum, XP, and Kanban use User Stories for facilitation rather than documentation. They depend on an onsite customer and emphasis on face-to-face communication. Table 5

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

summarizes these points.

Table 4. Requirements elicitation in agile methodologies.

Agile Methodology	Requirements Elicitation
Scrum	During the Product Backlog preparation, or Review meeting Product Owner with collaboration with Business Analyst/Developer, they elicit the requirements. As well during the sprint, Developer could ask question to Product Owner, as onsite customer reprehensive.
XP	Planning Game practice, where elicitation techniques like interviews, brainstorming and prioritization are used. Onsite customer.
Kanban	Onsite customer. Postpone elicitation and analysis until the last responsible moment, i.e. before requirement development/realization to working software.

Table 5. Requirements analysis in agile methodologies.

Agile Methodology	Requirements Analysis
Scrum	Prioritized Backlog items, Backlog Grooming, Product Owner prioritizes the Product Backlog, and analyses the feasibility of requirements.
XP	Planning Game practice, where analysis techniques like JAD, brainstorming and prioritization are used. Together software developers and onsite customers move the user story cards around on a large table to create a set of stories to be implemented as the first (or next) release.
Kanban	Onsite customer

Requirements validation in ASD may be handled differently, as a result of applying two agile principle, “Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale” and “Working

E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

software is the primary measure of progress [4].” Scrum validates the requirements through the Review meeting and receives immediate feedback from the Product Owner. In addition to using the acceptance criteria in the User Story to validate the requirement. While in XP, in addition to using a small release iteration, Stories are also translated into acceptance tests during the iteration. These acceptance tests are run during this iteration [15]. Table 6 summarizes requirements validation in agile methodologies.

Table 6. Requirements documentation in agile methodologies.

Agile Methodology	Requirements Documentation
Scrum	User stories, and face to face communication.
XP	User Stories in XP is for facilitation rather than documentation. Onsite customer is an on-time reference for instant elaboration or clarification.
Kanban	User stories, and face to face communication.

Requirements management in Scrum is represented in Sprint Planning meeting, Product Backlog tracking by change requirements (added/deleted) to/from Product Backlog. While in XP, Requirements management are not specified, as the User Story cards are destroyed after implementation. Kanban provides an efficient requirement tracing tool, which is Kanban board. This board visualizes the state of a User Story during the development. Table 7 summarizes requirements management in agile methodologies.

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

Table 7. Requirements validation in agile methodologies.

Agile Methodology	Requirements Validation
Scrum	Review meetings. Small iterations/sprints.
XP	Test Driven Development, prototyping, working software used by Onsite Customer
Kanban	Onsite Customer

Although the ASD model has resolved some traditional RE issues, it introduces other issues, as a consequence of trying to achieve an adequate balance between agility and stability. These issues are lack of practices that target non-functional requirements such as security and scalability, minimal documentation that raises traceability issues. In addition, the lack of customer availability for requirements clarification and feedback, which increases the rework, and a lack of harmony among customers [3].

Table 8. Requirements management in agile methodologies.

Agile Methodology	Requirements Management
Scrum	Sprint Planning meeting, Product Backlog tracking by change requirements (added/deleted) to/from Product Backlog.
XP	Requirements management are not specified, as the User Story cards are destroyed after implementation
Kanban	Tracing requirements via Kanban board, such figure 4 depicts.

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

6- Conclusion

RE activities, in ASD, have two main characteristics. First, they are handled on a small scale, which makes them embedded in the development life cycle, rather than isolated in a separate phase. Second, they are repeated frequently, under the idea of iteration. This paper presented a survey for RE activities in ASD methodologies. In order to help RE practitioners, who transform from the Waterfall model, to think and reason about RE activities in ASD, especially with an increase of software companies, which transform to ASD process model. ASD methodologies introduce various practices regards to RE activities implementation, such as face-to-face communication, customer involvement, and user story. However, there is still a need for more practices that deal with the issues that introduced by ASD methodologies. These issues are a result of finding a balance between agility and stability. Agility to deliver fast software products, as well as responding to change, and the minimal documentation that raises traceability issues. Requirement stability is required during an adequate amount of time, during development. In addition, there is a need for practices that target non-functional requirements such as security, usability, and scalability. Our future work will focus on studying how non-functional requirement elicitation is done in ASD. Our ultimate objective is to provide the RE practitioners' community with a complete guide to RE activities in ASD. This guide will include different RE activities, which handle either functional or non-functional requirements.

IJCI, VSRP Publishing, UK
https://ijci.vsrp.co.uk
E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

References

[1] R. S. Pressman, Software engineering : a practitioner's approach, 7th ed., New York: McGraw-Hill, 2010.

[2] "VersionOne 12th Annual State of Agile Report," VersionOne, 2018.

[3] I. Inayat, S. S. Salim, S. Marczak, M. Daneva and S. Shamshirband, "A Systematic Literature Review on Agile Requirements Engineering Practices and Challenges," *Computers in Human Behavior,* vol. 51, pp. 915-929, 2015.

[4] K. Beck, M. Beedle, A. v. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland and D. Thomas, "Manifesto for Agile Software Development," 2001. [Online]. Available: http://agilemanifesto.org/. [Accessed 16 January 2019].

[5] E. Hull, K. Jackson and J. Dick, Requirements Engineering, 3rd ed., London: Springer-Verlag, 2011.

[6] C. Pacheco, I. García and M. Reyes, "Requirements Elicitation Techniques: A Systematic Literature Review Based on The Maturity of The Techniques," *IET Software,* vol. 12, no. 4, pp. 365-378, 2018.

[7] K. Wiegers and J. Beatty, Software Requirements, 3rd ed., Redmond, Washington: Microsoft Press, 2013.

[8] I. Sommerville, Software Engineering, 9th ed., New York: Addison-Wesley, 2011.

[9] E.-M. Schön, J. Thomaschewski and M. J. Escalona, "Agile Requirements Engineering: A Systematic Literature Review," *Computer Standards & Interfaces,* vol. 49, pp. 79-91, 2017.

[10] O. Liskin, , K. Schneider, F. Fagerholm and J. Münch, "Understanding the Role of Requirements Artifacts in Kanban," in *Proceedings Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering*, 2014.

[11] K. S. Rubin, Essential Scrum A Practical Guide to The Most Popular Agile Process, Pearson Education, Inc, 2013.

[12] N. R. Darwish and S. Megahed, "Requirements Engineering in Scrum Framework," *International Journal of Computer Applications,* vol. 149, no. 8, pp. 24-29, 2016.

E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

[13] V. N. Vithana, "Scrum Requirements Engineering Practices and Challenges in Offshore Software Development," *International Journal of Computer Applications,* vol. 116, no. 22, pp. 43-49, 2015.

[14] M. Al-Zewairi, M. Biltawi and W. Etaiwi, "Agile Software Development Methodologies: Survey of Surveys," *Journal of Computer and Communications,* vol. 5, no. 5, pp. 74-97, 2017.

[15] D. Wells, "Extreme Programming: A gentle introduction," 8 10 2013. [Online]. Available: http://www.extremeprogramming.org/map/project.html. [Accessed 23 12 2018].

[16] N. Baruah, "Requirement Management in Agile Software Environment," in *The 2015 International Conference on Soft Computing and Software Engineering*, 2015.

[17] M. Poppendieck and T. Poppendieck, Lean Software Development: An Agile Toolkit, Addison Wesley, 2003.

[18] H. Lei, F. Ganjeizadeh, P. K. Jayachandran and P. Ozcan, "A Statistical Analysis of The Effects of Scrum And Kanban on Software Development Projects," *Robotics and Computer-Integrated Manufacturing,* vol. 43, pp. 59-67, 2017.

[19] M. O. Ahmad, D. Dennehy, K. Conboy and M. Oivo, "Kanban in Software Engineering: A Systematic Mapping Study," *The Journal of Systems and Software,* vol. 137, pp. 96-113, 2018.

[20] M. Hammarberg and J. Sundén, Kanban in Action, Shelter Island: Manning Publications Co., 2014.

[21] M. Rehkopf, "Kanban vs. Scrum," [Online]. Available: https://www.atlassian.com/agile/kanban/kanban-vs-scrum. [Accessed 09 02 2019].

E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n4p2

International Journal of Computers and Informatics (IJCI)

Vol. (2), No. (4)

المجلة الدولية للحاسبات والمعلوماتية

الإصدار (2)، العدد (4)

August 2023

انتظروا العدد القادم

المجلة الدولية للحاسبات والمعلوماتية

International Journal of Computers and Informatics (IJCI)

موقع المجلة: https://ijci.vsrp.co.uk
البريد الإلكتروني: ijci@vsrp.co.uk
رقم التليفون (واتس): +442039115546

دار النشر **رؤية للبحوث العلمية والنشر**، لندن، المملكة المتحدة

Vision for Scientific Research and Publishing, London, UK

71-75 Shelton Street, Covent Garden, London, WC2H 9JQ