Key Artefacts in the Initial Phases of IT Project Management: Systematic Mapping Study

Oksana Nikiforova[1][a], Kristaps Babris[1][b], Megija Krista Miļūne[1][c], Navyasri Tanguturi[1][d] and Óscar Pastor[2][e]

[1]*Riga Technical University, Riga, Latvia*

[2]

{oksana.nikiforova, kristaps.babris}@rtu.lv, {megija-krista.milune, navyasri.tanguturi}@edu.rtu.lv, opastor@dsic.upv.es

Keywords: IT Project Artefacts (Artifacts), IT Project Management, IT Project Initiating, IT Project Planning, Systematic Mapping Study.

Abstract: Mistakes made during the initial phases of an IT project are often critical as they can have cascading effect that impact every following phase of the project, especially implementation. These mistakes can lead to increased costs, delays and potential project failure. The initial phases of IT project, such as planning, requirements gathering, and design, set the foundation for the entire project defining project objectives, requirements and scope and setting the direction for the entire project. The paper demonstrates the results of the systematic mapping study performed on the definition of the types of artefacts created during IT project management before the implementation, as it lays the foundation for effective project planning, avoiding common pitfalls and ensuring alignment with industry best practices.

1 INTRODUCTION

Projects in the field of IT are organized into unique development phases. These phases assist teams from the starting point to finish, guaranteeing that complicated solutions are delivered successfully (Helmlinger, 2023). Usually these stages have initiation, planning, execution, monitoring and closure as their parts - each plays a vital role in maintaining concentration and gaining desired results. The beginning stages where specifically the initiation phase comes first followed by planning stage hold much importance because they state project goals and budgetary funds while assigning resources properly thus offering an understandable guide for teams (Omonije, 2024). Activities that involve multiple functions like quality checking and communication become crucial to fill spaces between teams and make sure they align with project objectives. The responsibility of overseeing these tasks falls upon project managers who balance technical needs against monetary limits and time factors to keep forward movement.

Recently, the COVID-19 pandemic has transformed company working styles. The "agile-style work environment" factor highlights the importance of conveying information efficiently and effectively using appropriate methods and tools in a remote setting (Binboga and Gumussoy, 2024).

Studying prior research and established frameworks like PMBOK (Project Management Institute, 2013), PRINCE2 (Simonaitis et al., 2023), or Agile (Agile manifesto 2001) methodologies provides access to best practices and standards for project management artefacts. By aligning with these practices, the project can meet industry standards and ensure consistency in documentation quality. This insight ensures that necessary documents are prepared at each phase, supporting both compliance and project coherence. A literature survey performed on understanding project documentation needs reveals the variety of artefacts typically required,

[a] https://orcid.org/0000-0001-7983-3088
[b] https://orcid.org/0000-0003-3855-6963
[c] https://orcid.org/0009-0002-2417-4518
[d] https://orcid.org/0009-0004-2203-6976
[e] https://orcid.org/0000-0002-1320-8471

Nikiforova, O., Babris, K., Miļūne, M. K., Tanguturi, N. and Pastor, Ó.
Key Artefacts in the Initial Phases of IT Project Management: Systematic Mapping Study.
DOI: 10.5220/0013471000003928

In *Proceedings of the 20th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2025)*, pages 773-781
ISBN: 978-989-758-742-9; ISSN: 2184-4895

such as project charters, requirement specifications, design documents, risk assessments, and testing plans. It helps to set clear expectations for documentation, making sure that important artefacts aren't overlooked, which could otherwise cause gaps in requirements, design, or quality.

Studying industry best practices can help IT managers to understand how other successful projects have approached artefact creation and can provide templates or guidelines that streamline work and reduce project ambiguity. Many artefacts (such as risk logs, compliance checklists, and quality assurance plans) play a crucial role in risk management and regulatory compliance (Schön et al, 2020). By understanding these through a literature survey, project teams can proactively address potential legal and security requirements and mitigate risks associated with non-compliance. Recognizing common risks and mitigation strategies found in literature reduces the likelihood of issues during the project and builds confidence among stakeholders. Different project management methodologies (e.g., Waterfall, Agile, DevOps) require specific types of artefacts. A literature survey clarifies the most used documentation needs associated with each methodology, helping project managers choose a documentation strategy aligned with their project needs. All these insights have strong traditions and are used over the years, but the pandemic situation changed the approach to IT project management and put corrections on the ability to perform certain processes and to support creation of particular artefacts turning focus on total digitalization of the IT project communication channels.

The goal of this paper is to go through the last five years scientific publications addressed to the artefacts used and created during IT projects corresponding to software development and to identify the most used and mentioned activities and their outputs during IT project management before software implementation. Thus, performing such a survey on artefacts created during IT project initiating and planning provides a roadmap to follow, helping IT project managers to prepare for each phase with the right tools and documentation. As a result, it can help to promote alignment with best practices and its compliance and to ensure efficient use of resources up to modern trends within IT projects.

The paper is structured as follows: Section 2 provides a discussion on related work, Section 3 outlines the research methodology applied for study collection, Section 4 presents and discusses the research findings, and Section 5 offers concluding remark.

2 RELATED WORK

To ensure that a solid foundation is established for the effective implementation of IT projects, the initiation phase is crucial in outlining objectives and aligning stakeholder expectations. In recent years, researchers have focused on the significance of artefacts and models, particularly regarding their relevance during the initiation phase for managing requirements, planning, and potential execution. This section reviews the current literature concerning artefacts in the initiation phase, model transformations, and the challenges encountered in this process.

An empirical study of Greer & Conradi (2009) highlights the variation in documentation practices among organizations and points out the trade-off between the resources devoted to thorough requirements engineering and the quality of the resulting plans. It examines how requirements frequently lack completeness or stability at the beginning of a project, which can influence the predictability and quality of initial planning, including cost-value assessment and scope definition.

In a similar context Wiegers & Beatty (n.d.), argue that textual descriptions increase the likelihood of misunderstanding when used as the sole medium for communicating requirements.

A most recent study performed by Kim et al. (2024) further discusses this issue by comparing text-based and model-based approaches within the domain of knowledge representation. They identified that both approaches are essential to realize genuine reflective representations of complex information but argue that model-based approaches provide clearer and more structured depictions of knowledge in situations, such as with multi-dimensional data. Their study underlines that text-based representations are very likely to be ambiguous, while model-based approaches, such as process models or use case diagrams, reduce ambiguity because they allow for a better-structured form of communication for complex ideas.

Building on this, a significant study performed by Sànchez-Ferreres et al. (2018) compares textual documentation with model-based representations. While it stresses the fact that such model-based descriptions are much clearer and more concise, it underscores the importance of process models. In particular, the use of Business Process Modelling Notation (BPMN) can provide more ordered and less vague ways of modelling project elements that can better express communication between various stakeholders.

Model transformation is critical in the initiation phase of IT projects, where textual requirements need to be translated into structured, visual models to reduce ambiguity and ensure clearer communication. As explained by Sendall & Kozaczynski, (2003), an effective model transformation language must be both expressive and efficient in handling this complexity of transforming as diversified textual descriptions, such as project charters and business cases, into structured models, such as UML diagrams, BPMN, or ER diagrams. This transforms project scope, objectives, and stakeholder requirements clearly. Additionally, Sendall & Kozaczynski (2003) further describes the importance of specifying conditions about when the transformation is applicable or valid. Transformations applied for IT project initialization should only happen if there exist certain conditions such as when the project scope or goal is clear and well defined; or after taking approval from other stakeholders. Again, it happens to align in line with controlling project risks with the aim to ensure models effectively capture changes made in the process.

Authors have been performed a systematic literature survey on last ten years solutions, where model transformations are used for IT project artefacts development during project initial stages (Nikiforova et al., 2025). The results of this survey identified artefacts, where model transformations have been successfully applied, and areas where they have not been utilized. While existing studies emphasize the advantages of model-based representations in IT project initiation, there is a notable lack of understanding regarding which artefacts are most transformed into models and the specific methods or frameworks used for these transformations. This gap includes limited insight into the types of artefacts frequently utilized during this phase, the systematic processes for deriving models from these artefacts, and the challenges encountered during these transformations. Addressing this gap is critical for establishing effective practices in IT project initiation, as it can enhance clarity, alignment, and communication among stakeholders.

The commonly studied user story often provides an insufficient description of software requirements. Numerous studies address challenges in requirements specification and propose various solutions. However, with the diversity of agile methods – each incorporating distinct practices – solutions must align with the specific ceremonies of each method. Few studies examine practices for requirements specification development within agile methods, and none offer a comparative analysis of these practices across different agile approaches (Herdika and Budiardjo, 2020).

Traditional metrics for software quality, such as defect density and mean time to failure, do not fully align with Agile iterative and sprint-based processes, prompting the need for new metrics like sprint velocity and burn-down charts (Chakravarty and Singh, 2021). Another research performed by Jarzębowicz and Weichbroth (2021) investigates the part of non-functional requirements in Agile Software Development projects, concentrating on existing methods and gathering techniques. A methodical review of literature and ten interviews with specialists from the industry unveiled a lack of agreement about when non-functional requirements should be recognized during the running cycle of a project. However, most experts give priority to early recognition along with constant improvement (Jarzębowicz and Sitko, 2020).

The related work in the area of machine learning for guessing effort in Scrum projects shows the difficulties and progressions in precisely predicting project effort inside Agile frameworks. Usual estimation methods, like expert opinion and regression analysis, have frequently not been enough in Agile environments because requirements and iterations can change often in Scrum. To handle these problems, researchers have more turned to ML models - multiple studies prove that ML ways usually do better than traditional strategies (Arora et al., 2020). In response to practical challenges in Agile estimation, studies have focused on identifying key project factors, such as complexity and team experience, that affect estimation accuracy.

Effective communication, training, and documentation are essential for successful agile requirement gathering. Collaboration and continuous improvement are also crucial, and feedback from stakeholders should be used to refine the approaches (Simhadri and Shameem, 2023).

Not having enough documents or good quality documents can make it hard for new members of the project. They might struggle to understand systems they are not familiar with and could make mistakes because of misunderstanding things. It is helpful when documentation is done at the end of a project cycle, after decisions have been made about how to implement everything (Nolan et al., 2022). That helps people maintain and change the system in future without needing constant updates as changes happen in systems. Also making strict rules around storing electronic document where people can easily find them may help avoid problems related to lost or unfindable information while also reducing unnecessary extra documents that no one uses.

Recent work on scaling agility in organizations has led to the development of taxonomies to systematically categorize Agile frameworks. As companies increasingly adopt frameworks for scaling Agile practices, research aims to establish a standardized understanding of the key dimensions and characteristics of these frameworks (Turhan et al., 2024). Wróbel et al. (2023) identified "Unfinished Tasks" as the most common anti-pattern, underscoring the critical role of effective planning and task management within sprints. Wróbel et al., (2023) also identified several other common anti-patterns, such as daily scrums exceeding the recommended duration, user stories lacking full refinement, and the sprint goal not being established during the sprint planning meeting. Among the various factors, customer-related and agile process factors are stronger predictors of process efficiency, sustainable software quality, and stakeholder satisfaction than other factors (Binboga and Gumussoy, 2024).

Based on comprehensive analysis of existing literature reviews in the area, the authors arrived to the following conclusions:

1) Further research could be conducted on the impact of effective requirement gathering on project outcomes (Simhadri and Shameem, 2023).

2) There is need for standardization of terminology, as semantically similar factors are often labelled differently across instrument (Santos et al., 2023).

3) When agile methods are implemented inappropriately, projects risk delayed or defective software, and overall decreased productivity (Nolan et al., 2022).

4) Currently, the procedure and practices of agile requirements engineering are still in the grey area (Herdika and Budiardjo, 2020).

Consequently, the systematic mapping study focusing on most published challenges in IT project management during initial stages of software development is not performed before and is quite required in modern situation with the rapid technologies and approaches changes. Moreover, such literature survey can help to determine whether which artefacts are more appropriate, guiding teams to develop a process that best supports the project's scope and timeline.

3 RESEARCH METHODOLOGY

The primary objective of this systematic mapping study research is to identify existing research on artefacts used for IT project management at the initial stages before the software implementation. In order to provide a focused direction for the corresponding papers collection the following research questions are formulated:

1) Which artefacts are mentioned in scientific papers as used for IT project management at the initial stages of the project?

2) Which artefacts in the initial stage of projects are obtained from which other artefacts on the same stage?

The collection of the corresponding studies is performed comprehensively in correspondence with the approach described by Kitchenham and Brereton (2013). An initial literature pool is constructed by examining Scopus, IEEE, ACM, ScienceDirect, IEEExplore databases of scientific papers. Firstly, the pool is filtered by reviewing study titles and abstracts. Secondly, a full-text assessment is performed for each remaining study to identify its relevance to the research scope. Subsequently, a snowballing technique is applied to identify additional relevant studies that may have been missed due to not being found with the search query. The following criteria are applied to select the initial pool of studies:

- Year of publication: 2020–2024.
- Language: English.
- Subject area: Computer science.

To identify potentially relevant studies, a systematic search was conducted using a predefined search query designed to capture relevant research within the paper scope. The search query employed the following keywords and logical operators:

("software" OR "information system") AND ("software development" OR "software project management") AND ("software requirement specification" OR "user story" OR "user stories") AND ("scrum" OR "kanban" OR "waterfall" OR "iterative" OR "incremental").

The initial search across these databases yielded a total of 304 studies. After excluding duplicate entries across databases, the remaining unique studies were consolidated into a single dataset. A manual screening process was subsequently conducted to evaluate the relevance of each study. This process involved assessing the titles, abstracts, and keywords of the papers against the scope of the study. The following inclusion criteria were used in the selection process:

1. Studies that explicitly address the IT project management artefacts at the initial stages of projects.
2. Research focusing on specific methodologies or frameworks such as Scrum, Kanban, Waterfall, Iterative, or Incremental models.
3. Papers discussing software requirement specification techniques, including user stories or similar representations.

Exclusion criteria:

1. Studies unrelated to software project management or development processes.
2. Papers focusing on later stages of IT projects.
3. Duplicate entries identified during the consolidation of datasets across databases.

This rigorous selection process ensured that the final pool of 118 studies was comprehensive, relevant, and aligned with the research scope. 24 studies published in 2020, 29 - 2021, 25 – 2022, 20 – 2023, 20 – 2024.

Consequently, all the artefacts, notations used for that artefacts and their types are registered in the spreadsheet. In turn to perform systematic mapping study on IT project artefact transformations, it is essential to identify the artefact (-s) discussed in the papers and to depict all mentioned transformations among them in one scheme. The mapping results are shown and discussed in the next section.

4 RESULTS AND DISCUSSION

The initiation phase of a project is generally seen as the basis on which the whole project rests. According to Russell (2018), during the initiation objectives are clearly stated, the scope of the project is defined, and a structure is created to align with the organizational goals and stakeholder expectations, which makes or breaks the success of the project. Initiating the project is important because this is the stage which defines the problems that need to be faced, specifies the parameters of success, and defines the necessary resources required for the project. Initiation of a project refers not only to the launching of it but also to laying the proper foundation of getting the project executed properly.

In the initiation phase, textual descriptions are used as the communicating key element of a project, such as objectives, scope, and requirements from the stakeholders. These text-based documents like project charters, business cases, and requirement specifications are often used at the onset of projects for planning, thus ensuring all stakeholders have an agreement on what the objectives of the project are. However, textual descriptions in the initiation phase are not without their challenges. Textual documentation can allow requirements to be ambiguous, and requirements of different types and perspectives are in danger of being unintentionally mixed-up during documentation. In that case, it is difficult to isolate information pertaining to a certain perspective amidst all the requirements in natural language (Pohl, 2016).

Textual descriptions often cause ambiguity and miscommunication among stakeholders, particularly in complex IT projects where requirements must be very well aligned. According to Wiegers & Beatty (n.d.), the likelihood of misunderstanding is increased when textual descriptions are used as the sole medium for communicating project requirements.

In turn to overcome these limitations, many organizations are looking at alternative approaches, such as using models instead of textual descriptions. Models are structured and visual ways of representing information, which can reduce ambiguity and enhance the clarity of the requirements (Pastor & Molina, 2007). For example, process models, use case diagrams, and entity-relationship diagrams are accurate ways of expressing project elements and thus enable easier identification of dependencies and communicate complex concepts to diverse stakeholders. These approaches, therefore, not only enhance communication but also support more in better project planning and execution.

The systematic mapping study explores transformations among artefacts used during the initial stages of IT projects, focusing on how the transformations cover all the activities performed at the beginning of the projects. The results provide valuable insights into the processes and practices adopted in the early stages of IT project development. The study revealed a wide range of artefacts used at the initial stages, including business models and requirements, user stories, as well as product and sprint backlog items. These artefacts vary in abstraction levels, reflecting different stakeholder perspectives and project needs.

Artefacts such as customer initial documentation and high-level requirements documents serve as transformations inputs, while user stories estimation and prioritization as well as product and sprint backlog planning and revision act as intermediate representations for transitioning from project ideation to elaboration before the development.

The analysis of studies collected by the research methodology identifies common transformation mechanisms, including manual translation, tool-supported transformations, and model-driven engineering practices (Nikiforova et al., 2009). For example, business requirements are often translated into system specifications using standardized templates or through stakeholder workshops. Similarly, models are converted into detailed specifications using modelling tools.

Artefact transformations in IT projects encompass a series of structured activities designed to ensure that information flows accurately from one stage to the next. These activities often require multidisciplinary collaboration, domain expertise, and the integration of tools to achieve seamless transitions. At the beginning of IT projects, raw business needs are collected through interviews, workshops, or surveys. These needs are often vague and require refinement into structured formats such as user stories or use cases. Activities here include defining priorities, identifying dependencies, and verifying requirements with stakeholders to ensure clarity and alignment with organizational goals. Figure 1 shows how these activities are covered with transformations offered in the studies collected for the mapping. Numbered references to the artefact's transformations used by authors in the survey are decoded in Table 1, giving the numbered reference in square bracket and the DOI of the corresponding study.

The transformation of requirements into system models is a critical step. Activities in this stage involve creating process flows, data models, and architecture diagrams to represent the intended solution. These models serve as blueprints for development teams, translating abstract requirements into actionable designs. Misinterpretation of requirements during transformations is especially evident when it is performed from informal artefacts (e.g., user stories) to structured ones (e.g., prioritized and estimated product or sprint backlog).

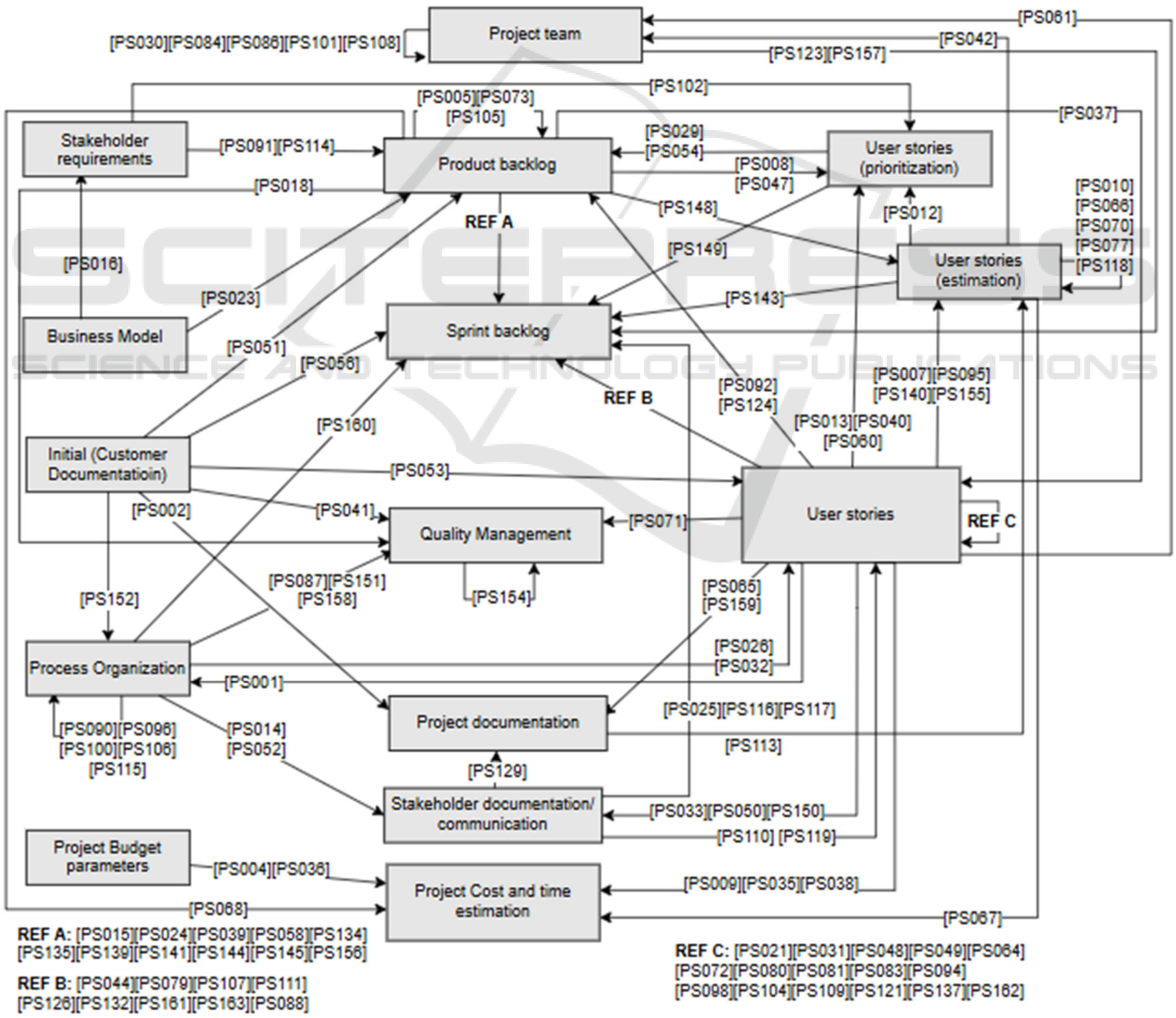

Figure 1: Transformations among IT project artefacts at the initial project stages offered in the collected studies.

Table 1: DOI of the studies offered the solutions for transformations shown in Figure 1.

ID	DOI	ID	DOI
PS001	10.1007/978-3-030-63329-5_2.	PS084	10.1111/isj.12282
PS002	10.1007/978-3-030-77474-5_2.	PS086	10.1007/s10664-020-09876-x
PS004	10.1007/978-981-15-1081-6_53.	PS087	10.1007/s10664-022-10208-4
PS005	10.1145/3493244.3493257	PS088	10.48550/arXiv.2008.02502.
PS007	10.29007/6vwh	PS090	10.1007/978-3-030-36674-2_30
PS008	10.1002/smr.2247.	PS091	10.1145/3403746.3403902
PS009	10.1016/j.advengsoft.2022.103159.	PS092	10.1109/ICSE-SEET55299.2022.9794220
PS010	10.1007/978-981-15-1451-7_59	PS094	10.11591/eei.v9i6.2484
PS012	10.1109/OCIT59427.2023.10430672.	PS095	10.1145/3468264.3473106
PS013	10.1109/ICOCO56118.2022.10031863.	PS096	https://ceur-ws.org/Vol-3776/paper02.pdf
PS014	10.1109/SERA51205.2021.9509045.	PS098	10.1049/sfw2.12037
PS015	10.1109/IBSSC53889.2021.9673243	PS100	10.1109/ICBATS54253.2022.9759013
PS016	10.1109/ACCESS.2021.3064424.	PS101	10.1016/j.infsof.2022.107079
PS018	10.11591/ijece.v11i6.pp5342-5350.	PS102	10.1109/ICNC47757.2020.9049681
PS021	10.1007/978-3-030-81242-3_10.	PS104	10.1109/ASE51524.2021.9678939
PS023	10.1007/978-981-19-9888-1_32	PS105	DOI:10.5381/jot.2022.21.3.a3
PS024	10.1007/978-3-031-35251-5_29	PS106	DOI:10.1145/3387940.3392241
PS025	10.1016/j.procs.2020.09.052.	PS107	10.1007/978-3-030-89817-5_30
PS026	10.1007/978-3-031-71142-8_21	PS108	10.1007/s11219-024-09688-y
PS029	10.1177/1063293X20958541	PS109	10.1016/j.jss.2022.111479
PS030	10.1109/APSEC57359.2022.00058.	PS110	10.5753/cibse.2024.28454
PS031	10.1007/978-3-031-43126-5_10	PS111	https://ceur-ws.org/Vol-3414/paper-1-preface.pdf
PS032	10.1145/3605098.3635901	PS113	10.1109/ICCSAI59793.2023.10421235.
PS033	10.1007/978-981-16-0404-1_24	PS114	10.1145/3524614.3528633.
PS035	10.1109/ACCESS.2020.3010968	PS115	10.3390/info14060327
PS036	10.1109/APCIT62007.2024.10673601	PS116	10.11591/ijeecs.v21.i1.pp360-366.
PS037	10.1155/2021/6611407.	PS117	10.1145/3377812.338216
PS038	10.1155/2022/3556809	PS118	10.1109/ICoDSE56892.2022.9972012
PS039	10.1109/RE57278.2023.00034	PS119	10.1007/978-3-030-94238-0_12
PS040	10.1142/S0218194023430015	PS121	10.1109/CONISOFT58849.2023.00017
PS041	10.1109/ICIC53490.2021.9693024.	PS123	https://web.archive.org/web/20201105065450id
PS042	10.1145/3328778.3366948	PS124	10.1007/978-3-031-21388-5_24
PS044	10.1109/ACCESS.2023.3305249	PS126	10.1109/RE57278.2023.00041
PS047	10.1007/978-3-030-63329-5_2	PS129	10.1007/978-3-031-60227-6_11.
PS048	10.1145/3555776.3577696	PS132	10.1007/s11219-022-09593-2
PS049	10.1145/3419604.3419793	PS134	10.1109/CONISOFT52520.2021.00023
PS050	10.18517/ijaseit.10.1.10176	PS135	10.1186/s13173-021-00114-w
PS051	10.3390/informatics11010012	PS137	10.1109/ACCESS.2024.3393831
PS052	10.1109/SEAI62072.2024.10674233	PS139	10.1007/978-3-031-03884-6_39
PS053	https://ceur-ws.org/Vol-3672/PT-paper2.pdf	PS140	10.1109/ENC56672.2022.9882947
PS054	10.1109/ICITSI56531.2022.9970965	PS141	10.3390/educsci11020073
PS056	10.1016/j.infoandorg.2020.100288.	PS143	10.1007/s10664-022-10192-9.
PS058	10.1109/INCOFT60753.2023.10425234	PS144	10.1007/978-3-030-96308-8_107
PS060	10.1007/s00766-022-00384-6	PS145	10.1109/ICT4S55073.2022.00013
PS061	10.1109/KI55792.2022.9925969.	PS148	10.1109/ESEM56168.2023.10304859.
PS064	10.1145/3383219.3383245	PS149	10.1080/20421338.2021.1955431
PS065	10.1007/978-3-030-67445-8_11	PS150	10.14569/IJACSA.2023.0140788
PS066	10.1007/978-981-19-7663-6_67	PS151	10.1109/CONISOFT55708.2022.00016.
PS067	10.1109/CIMPS61323.2023.10528839	PS152	10.1109/ATSIP62566.2024.10639040.
PS068	10.3390/math11061477	PS154	10.1007/978-3-030-89912-7_36
PS070	10.1016/j.infsof.2024.107447	PS155	10.1109/ICSME52107.2021.00017
PS071	10.1109/ACCESS.2024.3414614.	PS156	10.14569/IJACSA.2021.0121225
PS072	https://api.semanticscholar.org/CorpusID:270069156	PS157	10.1016/j.jss.2021.111013
PS073	10.1007/978-3-031-64576-1_17	PS158	10.1109/TELE58910.2023.10184341
PS077	10.1109/ICITACEE50144.2020.9239165	PS159	10.1109/ICIDM51048.2020.9339668
PS079	10.3390/app14198991	PS160	10.1007/978-3-030-88304-1_9
PS080	10.1007/978-3-030-79976-2_6	PS161	10.1109/IC2IE50715.2020.9274564
PS081	10.1109/CBI52690.2021.10066	PS162	https://www.researchgate.net/publication/381164049
PS083	10.1007/978-3-031-70245-7_19	PS163	10.1007/978-3-031-19968-4_5

The lack of standardized transformation processes often leads to inconsistencies and misalignments between artefacts, which can propagate errors to later stages. Tools and frameworks supporting transformations significantly improve the accuracy and efficiency of artefact obtaining from some source information in a form of well-structure transformation rules. Model-driven tools, for example, automate certain aspects of design creation, ensuring consistency across artefacts. Many IT projects use specialized tools to support artefact transformations. For example, requirements management tools may generate traceability matrices, while e.g. UML modelling tools can create technical diagrams (Nikiforova and Pavlova, 2008). Tool-assisted activities often include importing, exporting, and refining artefact formats to maintain compatibility and ensure information consistency across platforms. However, over-reliance on tools without proper customization or stakeholder input can lead to generic solutions that fail to address specific project contexts. As well as transformations are rarely one-direction and linear.

5 CONCLUSIONS

This study examined the studies on creation of IT project artefacts used in the initial stages of project management, with a focus on methodologies and frameworks such as Scrum, Kanban, Waterfall, Iterative, and Incremental models. The "agile-style work environment" factor emphasizes the critical need for efficient and effective communication, leveraging suitable methods and tools to ensure seamless information exchange in a remote work context. This highlights the critical role of well-defined IT project artefact, especially at the initial stages of the projects and underscores the need for structured approaches, collaborative practices, and technological support to optimize obtaining of these artefacts as complete and consistent.

Key findings were identified such as the role of artefacts, different transformation mechanisms, text description issues, technological support and the impact of flexible and remote working environments. While the study provides valuable insights, its limitations include industry differences, reliance on secondary sources and contextual differences. The study offers practical guidelines for improving project initiation and forms the basis for future research on the optimisation of artefact transformation in IT projects.

The study has been validated through a systematic selection process that ensures the reliability and relevance of the data collected. A comprehensive literature review identified 80 relevant studies from an initial 304 publications across multiple academic databases. The inclusion criteria ensure that only studies directly addressing the research objectives are considered. However, while the study analysed a wide range of studies, it may not have covered all the nuances of artefact transformation across different sectors and organisational structures. The reliance on secondary data sources means that some contextual details and case-specific insights may be overlooked, and differences in project complexity, stakeholder involvement and technological support further affect the applicability of the findings.

The added value of this study is the identification of structured transformation methods, highlighting the importance of model-based tools and collaborative practices. While this lays the foundations for improving the accuracy and consistency of artefacts, further empirical research and case studies are needed to validate these findings in real IT project environments.

ACKNOWLEDGEMENTS

This research has been supported by Research and Development grant No RTU-PA-2024/1-0015 under the EU Recovery and Resilience Facility funded project No. 5.2.1.1.i.0/2/24/I/CFLA/003 “Implementation of consolidation and management changes at Riga Technical University, Liepaja University, Rezekne Academy of Technology, Latvian Maritime Academy and Liepaja Maritime College for the progress towards excellence in higher education, science, and innovation”.

REFERENCES

Agile Manifesto (2001) https://agilemanifesto.org/

Binboga, B., Gumussoy, C. (2024). Factors Affecting Agile Software Project Success, IEEE Access, vol. 12, 95613-95633, DOI: 10.1109/ACCESS.2024.3384410

Chakravarty, K., Singh, J. (2021). A Study of Quality Metrics in Agile Software Development. In: Machine Learning and Information Processing. Advances in Intelligent Systems and Computing, vol 1311. Springer. DOI: 10.1007/978-981-33-4859-2_26

Greer, D., & Conradi, R. (2009). Software project initiation and planning – an empirical study. IET Software, 3(5), 356–368. DOI: 10.1049/iet-sen.2008.0093

Helmlinger, P. (2023). Agile transformation: A case study on early stage of agile adoption, Our Economy, Sciendo, Warsaw, Vol. 69, Iss. 1, 56-67, DOI: 10.2478/ngoe-2023-0006

Jarzębowicz, A., Sitko, N. (2020). Agile Requirements Prioritization in Practice: Results of an Industrial Survey, Procedia Computer Science, v. 176, pp 3446-3455, DOI: 10.1016/j.procs.2020.09.052

Jarzębowicz, A., Weichbroth, P. (2021). A Qualitative Study on Non-Functional Requirements in Agile Software Development, IEEE Access, v. 9, 40458-40475, DOI: 10.1109/ACCESS.2021.3064424

Josep Sànchez-Ferreres, Han, Carmona, J., & Lluís Padró. (2018). Aligning textual and model-based process descriptions. Data & Knowledge Engineering, 118, 25–40. DOI: 10.1016/j.datak.2018.09.001

Kim, M. K., Kim, J., & Heidari, A. (2024). Exploring the multi-dimensional human mind: Model-based and text-based approaches. Assessing Writing, 61, 100878–100878. DOI: 10.1016/j.asw.2024.100878

Kitchenham, B.; Brereton, P. (2013) A systematic review of systematic review process research in software engineering, Inf Softw Technol, vol. 55, no. 12, 2049–2075, DOI: 10.1016/j.infsof.2013.07.010

Nikiforova, O., Babris, K., Karlovs-Karlovskis, U., Narigina, M., Romanovs, A., Jansone, A., Grabis, J., & Pastor, O. (2025). Model Transformations Used in IT Project Initial Phases: Systematic Literature Review. Computers, 14(2), 40. https://doi.org/10.3390/computers14020040

Nikiforova, O., Nikulsins, V., Sukovskis U. (2009) Integration of MDA framework into the model of traditional software development, Frontiers in Artificial Intelligence and Applications, 187 (1), 229 - 239, DOI: 10.3233/978-1-58603-939-4-229

Nikiforova, O., Pavlova, N. (2008) Development of the tool for generation of UML class diagram from two-hemisphere model, 3rd International Conference on Software Engineering Advances, 105 - 112, DOI: 10.1109/ICSEA.2008.37

Nolan, A., Strickland, B., Quinn, A., et al. (2022). Exploring Aspects of Agile Software Development Risk – Results from a MLR. In: Yilmaz, M., et al. (eds) Systems, Software and Services Process Improvement. Communications in Computer and Information Science, vol 1646. Springer. DOI: 10.1007/978-3-031-15559-8_35

Omonije, A. (2024). Agile Methodology: A Comprehensive Impact on Modern Business Operations. International Journal of Science and Research, 13. DOI: 10.21275/SR24130104148

Pastor, O.; Molina, J. (2007). Model-driven architecture in practice: A software production environment based on conceptual modelling. Springer, DOI: 10.1007/978-3-540-71868-0.

Pastor, O., Nöel, R., Panach, I. (2021) From Strategy to Code: Achieving Strategical Alignment in Software Development Projects Through Conceptual Modelling. Transactions on Large Scale Data Knowledge Centred Systems. 48: 145-164, DOI: 10.1007/978-3-662-63519-3_7

Pasuksmit, J., Thongtanunam, P., & Karunasekera, S. (2021). Towards Just-Enough Documentation for Agile Effort Estimation: What Information Should Be Documented? IEEE International Conference on Software Maintenance and Evolution, 114–125. DOI: 10.1109/ICSME52107.2021.00017

Pohl, K. (2016). Requirements Engineering Fundamentals, 2nd Edition: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB compliant. United States: Rocky Nook.

Project Management Institute (2021). A Guide to the Project Management Body of Knowledge: PMBOK(R) Guide (7th. ed.). Project Management Institute. ISBN:978-1-935589-67-9

Russell, J. S., Pferdehirt, W. P., & Nelson, J. S. (2018). Project Initiation, Scope, and Structure. Unizin.org; University of Wisconsin-Madison. https://wisc.pb.unizin.org/technicalpm/chapter/project-initiation-scope-and-structure/

Santos, R., Cunha, F., Rique, T., et al. (2023). Evolution of Teamwork Quality Instruments in Agile Software Development: A Systematic Literature Review, 216-225. DOI: 10.1145/3613372.3613404

Schön, EM., Radtke, D., Jordan, C. (2020). Improving Risk Management in a Scaled Agile Environment. In: Stray, V., et al. (eds) Agile Processes in Software Engineering and Extreme Programming. XP 2020. Lecture Notes in Business Information Processing, vol 383. Springer. DOI: 10.1007/978-3-030-49392-9_9

Sendall, S.; Kozaczynski, W. (2003). Model transformation: the heart and soul of model-driven software development. IEEE Software, 20(5), 42–45. DOI: 10.1109/ms.2003.1231150

Simhadri, R., Shameem, M. (2023). Challenges in Requirements Gathering for Agile Software Development. 27th International Conference on Evaluation and Assessment in Software Engineering ACM, 406–413. DOI: 10.1145/3593434.3594237

Simonaitis, A., Daukšys, M., Mockienė, J. (2023). A Comparison of the Project Management Methodologies PRINCE2 and PMBOK in Managing Repetitive Construction Projects. Buildings 2023, 13, 1796. DOI: 10.3390/buildings13071796

Turhan, Y., Buehrle, D., Herzwurm, G. (2024). Developing a Taxonomy for Agile Scaling Frameworks. 7th ACM International Workshop of Software-intensive Business: Software Business in the era of generative artificial intelligence, ACM, p. 8. DOI: 10.1145/3643690.3648239

Wiegers, K., & Beatty, J. (n.d.). Software Requirements, Third Edition. https://thuvienso.hoasen.edu.vn/bitstream/handle/123456789/9059/Contents.pdf?sequence=5&isAllowed=y

Wróbel, M., Przała, D., Weichbroth, P. (2023). Exploring the Prevalence of Anti-patterns in the Application of Scrum in Software Development Organizations, DOI: 10.15439/2023F9562