The Impact of Requirements Artifacts on Efficiency in Agile Development: A Case Study

Sabine Molenaar
Dept. of Information and Computing Sciences
Utrecht University
Utrecht, The Netherlands
s.molenaar@uu.nl

Fabiano Dalpiaz
Dept. of Information and Computing Sciences
Utrecht University
Utrecht, The Netherlands
f.dalpiaz@uu.nl

***Abstract*—[Context] In agile software development, requirements are frequently represented in lightweight artifacts such as user stories (USs). To support their validation, USs may be complemented with additional artifacts such as acceptance criteria and test cases. [Problem] Existing studies propose frameworks for US writing, or investigate high-level factors that contribute to productivity, such as teamwork and support from management. However, empirical evidence on the impact of requirements artifacts on team efficiency is scarce. [Method] Through a mixed-methods case study at a large Dutch organization, we investigate how the use of requirements artifacts affects efficiency. We collected data from eight agile teams: 1,345 USs from nineteen sprints, US quality indicators, the quantity of associated acceptance criteria and test cases, and cycle time. We statistically analyzed the existence of correlations between these variables, and we triangulated our results through interviews with product owners and scrum masters from the involved teams. [Findings] Our employed indicators for US quality do not seem to have an effect on on-time completion, with our interviewees highlighting the trade-off between investing on writing quality and short-term efficiency. We found, instead, that the existence of acceptance criteria may be a relevant factor for on-time completion, as confirmed both by statistical tests and the interviewees.**

***Index Terms*—Agile Development, Requirements Artifacts, Requirements Engineering, Software Development.**

I. Introduction

Agile software development (ASD) gained rapid and widespread adoption across the past two decades [1]. Scrum, in its many variations [2], [3], is by far the most popular ASD method [4]. The definition of requirements in such projects is done incrementally and iteratively [5], and customer needs are often captured by writing user stories (USs) [6], [7]. These are most frequently expressed [7] via the Connextra format [8]: *As a [role], I want [action], so that [benefit].*

The systematic literature review by Inayat *et al.* [9] revealed that minimal documentation is a key challenge [10] for Requirements Engineering (RE) in ASD. The lack of details may limit traceability [9] and lead to challenges in accurately estimating USs [11], [12]. A likely consequence is rework [6], which results in diminished team efficiency.

Using high-quality RE artifacts is even more important when considering that team members predominantly rely on information contained in tickets and Scrum boards, rather than communicating with the customer on a regular basis [6]. This led to the creation of several US quality frameworks [13], either emerging from industry (e.g., INVEST [14]) or proposed from academia (e.g., Quality User Story [15]).

While studies at the intersection of ASD and RE exist (see Sect. II-B), they do not focus on the use and impact of requirements artifacts; the systematic literature mapping by Amna & Poels [16] confirmed the need of empirical studies on the use of USs and their effects on ASD processes.

We make a step toward bridging this gap. Specifically, we analyze the use of requirements artifacts of eight Agile teams in a large organization. We study the impact of requirements artifacts on efficiency—the time needed to fulfill USs—by considering the writing quality of 1,345 USs as well as the use of acceptance criteria and of test cases.

As a research method, we conduct an evaluative case study [17] and we adopt a mixed-methods approach to triangulate the quantitative results—obtained from the collected USs and related artifacts—with qualitative interviews with product owners (POs) and scrum masters (SMs) of the analyzed teams.

Our quantitative findings do not indicate a link between US quality and efficiency; however, they suggest a weak correlation between the use of acceptance criteria and the on-time completion of a US. The qualitative analysis provides additional explanations; for example, it reveals that developers perceive their efficiency being highly dependent on requirements that clearly express the desired functionality.

The rest of the paper is structured as follows. In Sect. II, we discuss related work. We present research design and questions in Sect. III. In Sect. IV, we describe the case study and provide demographics of the Agile teams. Data gathering and preprocessing are explained in Sect. V. After presenting descriptive statistics in Sect. VI, we report qualitative and quantitative results in Sect. VII. We discuss our results in Sect. VIII and conclude in Sect. IX.

II. Related Work

We introduce the relevant theoretical background (Sect. II-A), and then discuss empirical studies at the intersection of ASD and RE (Sect. II-B).

A. Theoretical background

According to Verwijs & Russo: "*Scrum teams are indeed more effective when they understand the needs of their stakeholders*" [4]. Stakeholder needs are described in requirements,

for instance in USs [8]. Creating USs, however, is challenging in Agile RE, as it takes time to define USs with the appropriate level of detail and team members need to be taught how to write them [18]. Cohn goes further on detailing USs, stating that "*we want detail that is just right*". While too little detail may lead to follow-up questions, incorrect features, and inaccurate estimates, too much detail requires excessive time and may restrict developers in their solution [19]. The study by Kasauli *et al.* highlights that while requirements decomposition is necessary, it is harder in ASD settings [20].

Lucassen *et al*. [7] investigated the perceived effectiveness of USs in practice and concluded that most practitioners participating in the study are largely positive about their experience with USs. However, they also found that most of these practitioners, unintentionally, do not employ any quality guidelines. Consistent requirements quality across granularity levels is also still a challenge in ASD [20].

The Quality User Story (QUS) framework [15] is a possible mitigation that offers thirteen quality criteria for US writing. These apply to either individual USs (well-formed, atomic, minimal, conceptually sound, problem-oriented, unambiguous, full sentence, estimatable) and to sets of USs (conflict-free, unique, uniform, independent, complete). While the authors did assess the quality of USs from eighteen software companies, they did not study the effect of a lack of quality. Their analysis reveals that, among automatically detectable defects, uniform, minimal and atomic violations are most frequent [15].

B. Empirical studies on Agile RE

Van Can & Dalpiaz [21] found that a single backlog item often includes different requirement types, as well as requirements at different levels of granularity. They also found that backlog item labels are often inconsistent with the content, e.g., an issue with a US may be labeled as 'task'. Too large backlog items may lead to estimation issues [11]. Sedano *et al.* [22] confirm this, stating that their observed backlog items were not always "*estimated nor fully fleshed out*", that several backlog items could not be considered requirements, that product managers groom the backlog frequently, and that prioritization may not be present nor set in stone.

Many papers explore factors that affect Agile team productivity, including management support [4], [23], employee commitment [24], teamwork [25], [26], team maturity [27], team autonomy [4], team design [23], member turnover [23], continuous improvement [4], and the adoption of Agile practices [28]. The Agile Team Effectiveness Model [29] proposes five components for effective Agile teamwork: shared leadership, peer feedback, redundancy, adaptability and team orientation. However, research that studies the effects of requirements artifact use and quality on team effectiveness is scarce.

Researchers often conduct surveys or interviews to gauge the (perceived) effect of certain practices and artifacts on productivity [7], [25], [28], [30]–[32]. The Scrumlity Framework [33] strives to include quality assessments in the ASD process. In their study, the researchers asked 78 undergraduate students whether using the QUS framework would benefit their projects and sixty participants said it would. However, this research focused on perception and did not involve practitioners.

We complement these efforts with a an in-vivo analysis of the relationship between the quality of USs and the use of requirements artifacts, acceptance criteria and test cases in particular, on the efficiency of fulfilling USs. Since the terms productivity and efficacy can also be interpreted as the output quantity (e.g., [23]) or the quality of delivered work (e.g., [4]), we intentionally use "efficiency" to emphasize that we focus on the time spent and that quantity and quality of deliverables are beyond the scope of this study.

III. Research Method

We conducted a mixed-methods evaluative case study [34] to answer the main research question (**MRQ**): *To what extent do requirements artifacts affect how efficiently USs are fulfilled?* Interviews with POs and SMs were held to gather qualitative findings, while data regarding USs were collected from a repository to identify quantitative findings. The research design is visualized in Fig. 1.

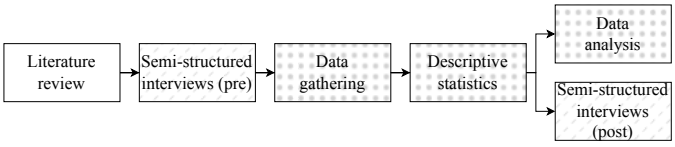


Fig. 1: Mixed-methods research design with qualitative (dashed) and quantitative (dotted) methods.

A. Conceptual model and research questions

We refine the MRQ into sub-questions on the basis of the conceptual model shown in Fig. 2. The model operationalizes *requirements artifacts* (in the MRQ) by considering *user stories* as the central artifact in ASD, characterized by their *quality* (IV), which we measure using part of the QUS framework. Moreover, we include two additional artifacts: the *acceptance criteria* (MV_1) and the *test cases* (MV_2) associated with a US, which are measured by their quantity. *Efficiency* is measured by the *time in days* (DV) it takes for a US to progress across various states (see Fig. 3 later).

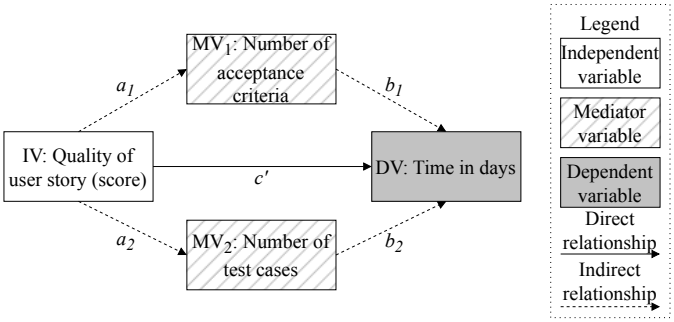


Fig. 2: Conceptual model of independent, mediator and dependent variables.

We consider acceptance criteria and test cases as means to validate a US. We introduce them as mediating variables, rather than independent variables, because we assume that the quality of a US may affect the number of acceptance criteria and test cases associated with that US. For example, if QUS' *atomic* criterion is violated, a US expresses a requirement for two or more features. Those features need to be validated, so acceptance criteria can be expected for each feature.

The conceptual model justifies our research questions. We first check if the effects of a_1 (RQ_1) and a_2 (RQ_2) are significant; if not, the mediator variables will be considered independent variables for the remaining analysis, thereby making b_1 (RQ_3) and b_2 (RQ_4) direct (non-mediated) relationships. Finally, c' (RQ_5) is considered a direct relationship between the main IV and the DV, regardless of other results.

First, we investigate how the quality of a US influences the existence and quantity of acceptance criteria:

RQ_1 *To what extent does the quality of a US affect the number of associated acceptance criteria?*

For RQ_1, we define a specific hypothesis, since we expect that USs with multiple actions, which express a need for many features, would have a higher number of acceptance criteria:

H_1 *USs that include multiple features have more associated acceptance criteria than USs including a single feature.*

We then define a similar research question to RQ_1, but for the other mediator (MV_2):

RQ_2 *To what extent does the quality of a US affect the number of associated test cases?*

RQ_3 investigates the relationship between US quality and the efficiency of its fulfillment:

RQ_3 *To what extent does the quality of a US affect the time spent on that US?*

RQ_4 focuses on the possible effects of using acceptance criteria associated with a specific US on how efficiently this US is fulfilled. This is justified by the observation that, while formulating acceptance criteria during refinement of a US takes time, it might reduce the time taken to develop it and have it accepted, thanks to the added clarity.

RQ_4 *To what extent does the use of acceptance criteria affect the time spent on a US?*

Finally, RQ_5 takes the same approach as RQ_4, but is instead aimed at the use of test cases for a specific US.

RQ_5 *To what extent does the use of test cases affect the time spent on a US?*

B. Qualitative method

In order to obtain a rich characterization of our case study context, we interviewed POs or SMs of the participating teams prior to and after data collection and analysis. All interviews were conducted by the same researcher (the first author of this paper) and lasted around an hour.

Pre-testing interviews. We asked questions on team demographics and on their use of requirements artifacts:

1) What is the size of the team in number of employees?
2) Are there any third-party team members?
3) Are there manuals, guidelines, or instructions for the team to use concerning how to document their work?
4) Do you use a US template?
5) Do you formulate acceptance criteria for USs?
6) If acceptance criteria are written, do you use a template?
7) Who is responsible for approving (*accepting*) the implemented USs?

Interviews were conducted before data collection to ensure that the data of the participating teams would be comparable.

Post-testing interviews. Six of the eight participating teams were invited to the post-testing interviews. Two teams (Echo and Foxtrot in Table I) were excluded due to the limited number of USs created in the data collection period. In each interview, the descriptive statistics of that team were shared and compared to average of the other seven teams, in order for the interviewee to compare their practices against those of the other teams, but without disclosing their identities.

The interviews were semi-structured and covered at least the following topics:

- US quality: most frequently violated criteria by the team;
- Number of acceptance criteria: comparison to other teams and average of all teams (see Fig. 4 later);
- Number of test cases: comparison to other teams and average of all teams (see Fig. 4 later);
- Time spent: average time USs spent in a state and the maximum time spent, compared to other teams;
- Qualitative findings: specific examples of quality criteria violations from the data gathered for the team in question.

Before asking questions about the quality criteria, we explained these to the interviewees. We decided to not share the results from the quantitative data analysis, as we intended to triangulate results and wanted to avoid confirmation bias.

In addition, all teams were asked the following questions:

1) What do you think is the most frequently made error in writing USs?
2) In which state does a US generally spend the most time?
3) What is the reason some USs spend much more time in a specific state?
4) What do you think improves the quick development of USs?

Team-specific questions were asked too. For example, one team used a template with a minimum number of acceptance criteria, but this was not met for every US. They were asked why they thought this rule was not (always) adhered to.

C. Quantitative method

We analyzed quantitative data regarding the requirements artifacts collected from a requirements documentation and sprint planning repository. For the quantitative analysis of the RQs, we gathered data regarding the time it took teams to fulfill USs, the number of associated acceptance criteria and test cases, and the quality of the USs. The US quality was not

available as a data item, so this was added. The relationships between these variables were studied using statistical tests.

We assessed US quality using the QUS framework [15]. We aimed to consistently and efficiently evaluate the USs, so we excluded quality criteria we considered infeasible to objectively assess or estimate. First, we excluded those criteria that need to be assessed on a collection (e.g., conflict-free), as those are too complex to reliably assess, since scopes and active USs keep changing due to iterative and incremental development. Also, this assessment is too time-consuming as it would require comparing all combinations of USs. We then excluded criteria in the 'semantic' category, as they are subject to interpretation, thereby requiring multiple taggers; this conflicted with our goal to gather a large number of USs. Finally, the 'estimatable' criterion would also require a contribution from the teams, taking into account that only they know what the size of the US is, even when using story points consistently, and if they consider it possible to plan and prioritize. In addition, US sizes between teams may not follow an interval scale, making comparisons subjective and inaccurate.

Given these exclusions, four QUS criteria remained, which we list below along with their descriptions adapted from [15]:

1) *Well-formed*: A US includes at least a role and an action.
2) *Atomic*: A US expresses a requirement for exactly one feature.
3) *Minimal*: A US contains only role, action and benefit.
4) *Full sentence*: A US is a well-formed full sentence.

IV. Case Study Description

Our case study was executed at a large organization based in the Netherlands. Due to the large number of developed and maintained applications, development teams are organized in Agile Release Trains (ARTs), each containing multiple Agile teams. At the time of writing, there are over 200 Agile teams. The organization was chosen for convenience; we randomly selected the teams from the organization's list of Agile teams, ensuring they belong to different ARTs. The only prerequisites were: use of the same project management system to record their Agile development artifacts and willingness to participate. All selected teams used Rally, a project management system by Broadcom for managing Agile processes and artifacts in large organizations. Rally also supports backlog management, (sprint) planning, release tracking, dashboarding, among others. Fig. 3 illustrates the cycle time for USs in Rally and in the ASD process at the case organization, showing the names of the states as obtained from Rally alongside the action moving an item to the next state (*A*), and the corresponding phases in the organization's ASD process (*B*).

The team demographics in Table I are based on the answers provided to the pre-testing interviews (see Sect. III). Note that the team names are fictional and used to protect the identities of the teams and their members. All representatives stated they use a template for USs, specifically, the Connextra format [8]: "*As a [role], I want to [action], so that [benefit].*" All interviewees also said they formulate acceptance

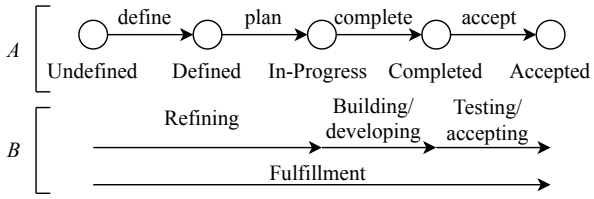


Fig. 3: Cycle time for USs: *A*: states for items in Rally. *B*: corresponding phases in the ASD process at the organization.

criteria. However, teams either did not use a template for acceptance criteria, did not do so consistently, or used different templates. Given this heterogeneity, we decided to only use the existence and number of the criteria. This is in line with the study conducted by Hess *et al.*, as they state that their interviewees considered USs the main RE artifacts and that acceptance criteria, when used, do not follow a homogeneous structure [6]. Three teams—Alpha, Delta and Echo—wrote their USs predominantly in Dutch, while the other teams used English as their main language.

TABLE I: Demographics of the teams included in the study.

Team name	Avg. size in #employees	Type of development	Use of manual	Dedicated reviewer	Ext. team members
Alpha	8 to 10	Low code	Guidelines	Requestor	No
Bravo	8 to 10	High code[1]	Yes	Yes	Yes
Charlie	12 to 18	High code	Yes	Yes	Yes
Delta	10 to 12	Low code	Guidelines	Yes	No
Echo	10 to 12	High code	Templates	Varies	Yes
Foxtrot	10 to 12	High code	Templates	Varies	Yes
Golf	8 to 10	Mixed	Yes	Yes	Yes
Hotel	10 to 12	High code	Guidelines	Yes	No

V. Data Preprocessing

In this section, we describe how data was gathered and any cleaning and transformation activities we performed. Throughout the remainder of this paper, data items and variables are stylized in `typewriter font`.

A. Data gathering and consolidation

We extracted from Rally data of nineteen sprints (January–September 2023) for each of the eight teams, for a total of 152 sprints. We only kept data items where the US went through the complete cycle (Undefined to Accepted in Fig. 3) in 2023, removing items that started earlier or finished later. The Rally data included: an `ID` for every item; the days the item spent in each state (decimal): `Undefined`, `Defined`, `In-Progress`, `Completed`; and the total `Cycle time` in days (decimal).

We manually assessed the four considered QUS framework criteria, each becoming a boolean variable. Wouters *et al.* reported high inter-rater reliability for the well-formed and atomic criteria [35], which, alongside confidentiality reasons, made us rely on a single tagger. For the `Well-formed` quality, we checked for the existence of a role and an action; for `Atomic`, we verified the inclusion of lists or options,

[1]We use "high code" to denote the opposite of low code.

for instance, through use of the words "and" or "or". The `Minimal` criterion is considered violated when additional text is included in a US, for instance in parentheses or after the period. The `Full sentence` criterion is violated when it lacks necessary words to be syntactically correct, e.g., subject or verb. We had to recover the number of associated acceptance criteria (`ACriteria`) and test cases (`TestCases`) by hand, due to the use of heterogeneous practices across the teams. While they often used Given-When-Then scenarios [36], a list with pre-conditions and steps to perform, or a checklist with items such as "*button x should be visible on page y*", they did not do so consistently for all USs. The use of different formats was observed within teams as well. Fortunately, all teams used a list, numbered or itemized, for these artifacts, so counting was a straightforward activity.

B. Data cleaning and transformation

First, the data was checked for duplicate items, by looking for non-unique `IDs`, but none were found. Second, Rally items that did not contain a US were removed (1,666 items). Third, USs with a value below 1 in the column `Cycle time` were removed, since it seems highly unlikely a US was defined, planned, completed and accepted within a single day. Fourth, USs with a value of less than 1 in the columns `Undefined`, `Defined`, `In-Progress` and/or `Completed` were removed from the dataset for that specific state (but retained for analyses of other states, if the time in those states was above 1).

A new categorical variable called `Quality` was added for the remaining (1,345) USs, counting how many QUS criteria were met: 0, 1, 2, 3 or 4. Scores of 0 and 1 were not observed.

This resulted in an Excel file with the variables shown in Table II. The data were analyzed both per `Team` and in `Total`. We did not analyze statistically cases with too limited data, e.g., fewer than five USs in the considered state.

TABLE II: Columns/variables in the final dataset.

Variable	Type	Comment
ID	String	Unique identifier for a US
Team	Categorical	Team name: Alpha, Bravo, ..., Hotel
Undefined	Decimal	Days spent in state *Undefined*
Defined	Decimal	Days spent in state *Defined*
Progress	Decimal	Days spent in state *In-Progress*
Completed	Decimal	Days spent in state *Completed*
Cycle time	Decimal	Days from *Undefined* to *Accepted*
Well-formed	Boolean	Is the US well formed?
Atomic	Boolean	Is the US atomic?
Minimal	Boolean	Is the US minimal?
Full sentence	Boolean	Is the US a full sentence?
Quality	Categorical	Number of met QUS criteria: 0...4
ACriteria	Non-negative Integer	Number of acceptance criteria
TestCases	Non-negative Integer	Number of tests cases

C. Selecting statistical tests

We chose to analyze the teams' data in a uniform way; if one team did not meet the assumptions for a specific test (e.g., linearity) all other teams' data would be studied via non-parametric, more conservative tests. All tests were run in Python using the Pandas and SciPy libraries.

First, we checked whether Alpha's dataset followed linear models for QUS and the state `Undefined`, QUS and the number of acceptance criteria, and QUS and the number of test cases. According to the Kolmogorov-Smirnov test, this assumption was not met for any of these three sets. In addition, we performed a Shapiro-Wilk test to check if the `Undefined` dataset was normally distributed, with a p-value of 1E–14, this was also not the case. Therefore, all datasets were tested using non-parametric tests. We used Spearman's rank order correlation test to test for a relationship between the variables, since this test allows the use of non-linear and non-normally distributed data, as well as a combination of categorical and continuous variables [37]. We opted for correlation analysis, rather than regression (which would be better suited for studying causality), due to the co-evolution of our variables and the presence of external, uncontrolled factors [38]. We interpret correlations, positive or negative, of 0.1 to 0.3 as weak, 0.4 to 0.6 as moderate (stylized in *italics*) and 0.7 to 0.9 as strong (in **bold**) [39]. The raw data used in this study cannot be shared due to confidentiality agreements.

VI. Descriptive Statistics

All the tables and figures shown from this point on contain preprocessed data items only. Table III shows the number of violated US quality criteria per team. The most frequently violated criterion is that of atomicity: 19.7% of the assessed USs expressed a requirement for more than one feature. Next, 12.6% of USs contained additional information and therefore violated the minimal criterion. Well-formedness and writing full sentences are almost always met by the teams. In Sect. VII-F, we provide some examples of typical violations.

TABLE III: Number of violated criteria per team. Labels: WForm = Well-formed; Atom = Atomic; Min = Minimal; FullS = Full Sentence.

	WForm	Atom	Min	FullS	*n*
Alpha	1	59	7	1	225
Bravo	0	11	4	0	121
Charlie	2	37	16	2	185
Delta	3	46	22	1	242
Echo	0	2	3	0	15
Foxtrot	3	14	26	2	75
Golf	0	46	15	2	195
Hotel	3	50	76	5	287
Total	12	265	169	13	1,345
Mean μ	1.5	33.1	21.1	1.6	-
Std-dev σ	1.3	19.8	22.1	1.5	-
Share (%)	0.9	19.7	12.6	1.0	-

In Table IV, we map the quality of USs (Q), i.e., the count of criteria that were met, into quality scores: low (two), medium (three), and high (four).

Most of the USs that were assessed (69.7%) contained no violations and can be considered of high quality according to the criteria we used. It is rare for a US to meet only two criteria, as this was observed in less than 4% of the USs.

Table V reports on the number of acceptance criteria and the number of test cases per US for each team. Even though

TABLE IV: Quality of USs, presented in categories, per team.

	Low	Medium	High	*n*
Alpha	3	62	160	225
Bravo	0	15	106	121
Charlie	5	47	133	185
Delta	11	50	181	242
Echo	0	5	10	15
Foxtrot	13	19	43	75
Golf	4	55	136	195
Hotel	16	102	169	287
Total	52	355	938	1,345
μ	6.5	44.4	117.3	-
σ	5.7	29.2	57.3	-
Share (%)	3.9	26.4	69.7	-

TABLE V: Number of acceptance criteria and test cases per team.

	Acceptance criteria			Test cases		
	n	μ	Mode	*n*	μ	Mode
Alpha	525	2.3	1	253	1.1	1
Bravo	175	1.4	0	98	0.8	0
Charlie	81	0.4	0	40	0.2	0
Delta	776	3.2	1	145	0.6	1
Echo	37	2.5	0	13	0.9	0
Foxtrot	79	1.1	0	17	0.2	0
Golf	463	2.4	1	157	0.8	0
Hotel	618	2.2	0	89	0.3	1
Total	2,754	1.9	0	812	0.6	0

acceptance criteria and test cases were often included in the used template, they were not always followed. Also, we found that acceptance criteria are more numerous than test cases. Fig. 4 shows the distribution of acceptance criteria and test cases per US. For example, 352 USs had no acceptance criteria, while 309 USs had one acceptance criterion.

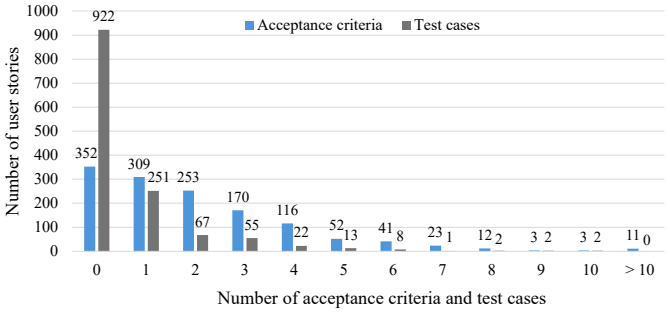


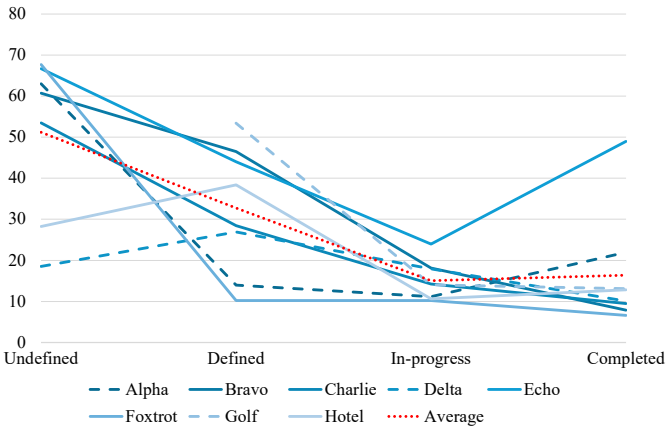
Fig. 4: Number of acceptance criteria and test cases per US.

According to the pre-testing interviews, most teams have a section for including acceptance criteria in their default template. Test cases, however, are not mandatory for most teams. This is also visible in Fig. 4; USs without acceptance criteria are less common than USs without test cases.

Table VI shows the time USs spent in each state, organized by team; team averages are plotted in Table VIa, details are in Table VIb. Visible differences across teams exist concerning the average time spent in states `Undefined` and in `Defined`, as per Table VIa. Possible explanations are that `In-Progress` and `Completed` are monitored more closely and given greater priority; teams often need to build functionality in time and are not allowed to release new functionality until it is accepted.

TABLE VI: Time spent (in days) per development phase of the US, per team.

(a) Average time per state (omitting the single US from Golf in `Undefined`); teams who use low code are shown in dashed lines.



(b) Details per team and the total of all teams.

		Alpha	*Bravo*	*Charlie*	*Delta*	*Echo*	*Foxtrot*	*Golf*	*Hotel*	*Total*
Undefined	μ	63.0	60.7	53.5	18.6	66.7	67.7	283.9	28.2	44.5
	Min	1.9	2.1	1.0	1.0	5.3	1.0	283.9	1.1	-
	Max	325.9	162.0	337.9	211.8	187.0	269.1	283.9	569.1	-
	n	189	11	150	193	12	56	1	111	723
Defined	μ	13.9	46.5	28.4	26.9	44.0	10.2	53.4	38.4	34.4
	Min	1.0	1.0	1.0	1.0	1.2	1.0	1.0	1.0	-
	Max	82.0	416.0	403.7	371.9	95.0	110.7	377.9	443.9	-
	n	129	88	151	121	13	26	149	257	934
In-**P**rogress	μ	11.2	18.1	14.3	17.9	24.0	10.2	14.0	10.6	13.7
	Min	1.0	1.0	1.0	1.0	2.3	1.1	1.0	1.0	-
	Max	58.9	282.3	97.3	124.8	80.2	35.3	93.2	91.0	-
	n	207	87	156	197	12	69	153	246	1,127
	% ontime	74.4	71.3	66.0	56.9	50.0	78.3	67.3	75.6	69.2
Completed	μ	21.9	7.8	9.5	10.1	49.0	6.6	13.1	12.8	13.9
	Min	1.0	1.8	1.0	1.0	5.9	1.0	1.0	1.0	-
	Max	82.1	34.2	55.1	85.0	199.2	40.0	132.8	105.7	-
	n	157	81	49	146	12	29	71	150	695

The time spent in `In-Progress` and `Completed` seems to align with sprint duration, as the averages for these states are around fourteen days, the number of days in a sprint for all of the eight teams. Thus, we also divided the USs included in the `In-Progress` state into two groups: on time ($\leq$ 14.00 days) and not on time ($>$ 14.00 days), using the two-week sprint target value. The percentage of USs that are completed on time are included in Table VIb (row "*% ontime*").

VII. Results

We analyze each RQ, presenting both the qualitative and quantitative results. We indicate which teams supported a qualitative finding by including the team name's first letter (full names in Table I). For instance, (AD) denotes that teams Alpha and Delta provided a given response. Significant quantitative results are **emphasized**. We denote the effect between two

variables with an arrow, e.g.: IV → DV. We use correlations to test if the variables are statistically associated, and qualitative data to learn more about the effect (e.g., how IV affects DV).

We evaluated if significant results were limited to teams with shared demographics, e.g., if all significant results could be associated with low-code teams (Table I). Since we found no indication, we did not perform statistical testing based on the demographic characteristics. If interviewees mentioned any team characteristic, those are included qualitatively. Key findings are **emphasized** and numbered (e.g., **F1**).

The interested reader can find tables showing the statistical test results of the effect of US quality on acceptance criteria (Q → A), on test cases (Q → T), and on time (Q → time), as well as the effect of acceptance criteria and test cases on time (A → time and T → time) in our online appendix [40].

A. RQ_1 & RQ_2: To what extent does the quality of a US affect the number of associated acceptance criteria (Q → A) and of associated test cases (Q → T)?

Tests on the correlation between US quality and number of acceptance criteria were significant only for 2 of the 8 teams, both of weak strength (ρ = 0.227 for Foxtrot, ρ = 0.172 for Hotel). So, the quantitative data does no support the correlation for RQ_1. We draw a similar conclusion for the relationship between US quality and the number of test cases (RQ_2). We only obtained one significant result on the total dataset for Q → T, but the correlation strength is negligible (ρ = 0.054).

We also tested hypothesis H_1 on whether atomic violations correlate with the number of acceptance criteria. The correlation was not significant (p = 0.083) and of negligible strength (ρ = –0.047), therefore we reject H_1.

The results show that **there is no reason to assume that US quality affects the number of associated acceptance criteria or test cases (F1)**; therefore, the two mediator variables MV_1 and MV_2 are treated as independent from here on.

B. RQ_3: To what extent does the quality of a US affect the time spent on that US? (Q → time)

When asked "*What do you think is the most frequently made error in writing USs?*", most POs/SMs assumed either atomic (ACDG) or minimal (CGH). Motivations are as follows:

- Well-formed: they want to write the USs quickly (H);
- Atomic: they like to be exhaustive in their USs (AG): if the required changes are quick to build, they prefer combining them as that takes less time than documenting multiple USs (BCD);
- Minimal: they like to include context, especially for third-party (external, see Table I) team members (CG);
- Full sentence: team members aim to work quickly and might not take the time to write a full sentence (BD).

Two teams (CD) mentioned they did not mind atomic violations, because they sometimes **combine multiple changes in one US to avoid double administrative work (F2)**. One team (B) was surprised by the absence of full sentence violations, as the team was very experienced and knew each other well, so they "did not need many words to communicate well".

We tested for a correlation between US quality and time spent in each state (U: Undefined, D: Defined, P: In-Progress, C: Completed) and in total CT: Cycle Time. Only four results were significant. Three are of weak strength: Q → U for Foxtrot, ρ = 0.329; Q → C and Q → CT for Hotel, with ρ = –0.256 and ρ = –0.345. One, with negligible strength, for Q → C on `Total`, with ρ = –0.079. Not only were the other combinations insignificant, but the results also show correlations in opposite directions, thereby making us conclude that **US quality does not affect the time variables (F3)**. We also tested for the effect of atomic violations on `In-Progress` and on-time completion, but found no significant correlation.

C. RQ_4: To what extent does the use of acceptance criteria affect the time spent on a US? (A → time)

Most teams are in favor of using acceptance criteria and consider them a key artifact in the development process. They represent the boxes that need to be checked to fulfill the US and serve as a guideline: "*do we understand what we need to do and what is expected of us?*" (ADGH). Acceptance criteria also allow the team to assess whether the US is testable (DG). Developers often include a technical solution in the acceptance criteria, which they can then build (H). Furthermore, they do not only specify a minimal solution, they also prevent developers from doing too much and spending time on inessential functionality (G). The only argument against their use is that they are unnecessary for simple USs; in those cases, writing them wastes time (CD). Bravo, for instance stated that acceptance criteria were not part of their template at the time and that, due to the team's experience, they usually did not 'need' acceptance criteria (B).

The added value of acceptance criteria for the team is widely considered to be ensuring developers understand what the business expects of them during refinement (ABDGH). Acceptance criteria are also 'proof' that you did what needed to be done (G). Finally, if used, they help make the requests explicit and increase the clarity and findability of this information, but mostly for less experienced team members (C). The participating teams are divided on when acceptance criteria play the largest role in the development process. For instance, during building and accepting (A), during refinement

TABLE VII: Correlation between acceptance criteria and whether USs are completed in time or not, α = 0.05.

		Alpha	*Bravo*	*Charlie*	*Delta*	*Echo*	*Foxtrot*	*Golf*	*Hotel*	*Total*
A → P*	*n*	207	87	156	197	12	69	153	246	1127
	ρ	0.305	0.106	0.171	0.239	-0.149	0.117	0.067	-0.063	0.128
	p-value	**7.94E-06**	0.327	**0.032**	**0.0007**	0.644	0.339	0.413	0.328	**1.61E-05**

and building (B) or during refinement only, as this is when all developers and testers come together to discuss the USs (C). However, half teams see a role for acceptance criteria during refinement and accepting (DGH). Questions from the team during the refinement sessions are often about the acceptance criteria (H). In addition, in the refinement phase, it is still possible to make changes based on acceptance criteria (G).

We tested for a correlation between the number of acceptance criteria and time spent, and we found some significant results. The most frequent case is the `In-Progress` state; although with a weak strength, four teams had a significant correlation for A $\rightarrow$ P. The correlation is confirmed on `Total`.

To test this relationship more rigorously, we examined if the use of acceptance criteria makes USs more likely to be completed on time, which we verified by selecting a target value of 14.00 (in days), as explained in Sect. VI. Table VII reports on the correlation between acceptance criteria and whether USs are completed on time or not (denoted as P*).

For the relationship between number of acceptance criteria and USs completed on time, 3/8 tests are significant, plus the test on `Total`. All significant results show a positive relationship with weak strength. We conclude that **our data shows a weak correlation between the number of acceptance criteria and whether USs are completed on time (F4).**

*D. RQ_5: To what extent does the use of test cases affect the time spent on a US? (*T $\rightarrow$ time*)*

Test cases are less popular than acceptance criteria among our teams. They are sometimes used as instructions for testers, in order for them to check if the acceptance criteria are met (AG). Teams occasionally refrain from formulating test cases, because they make use of test automation (H) or they believe all necessary information for testing is already included in the US itself and its acceptance criteria (CD). USs that contain maintenance tasks also do not require test cases (B). In summary, teams that work with dedicated testers write test cases, while teams that do their own testing are more likely not to include test cases, unless USs are particularly complex.

According to the team representatives, the added value of test cases is for testers to execute a specific test (ABH) and to help perform specific and complex tests such as chain and regression testing (D). They can also serve as documentation: if a bug is found after release, the team can check whether they missed this in a test or whether it has a different cause (C). Finally, they can help the team identify unclarity or complexity early in the process. During refinement, test cases help identify misunderstandings in the US and acceptance criteria. This allows them to make the USs clearer and easier to build and estimate (G). Most teams argued that test cases only play a role during the testing phase of development (ABDH).

The participating teams suggest that acceptance criteria are more commonly used than test cases because **acceptance criteria are needed to check whether all team members understand what needs to be done (F5)**; whether there is agreement between business and developers (ADH). Acceptance criteria are a quality check or a list of to-do items, which is relevant in any case, while tests are for the tester to do a specific task (B). The absence of dedicated testers is mentioned as the reason for not needing test cases (DG).

When testing for significant correlations between the number of acceptance criteria and time, we found some significant results (in 8/40 combinations of team and time state), but the results differ greatly per team. The correlation between the number of test cases and time spent in `In-Progress` (T $\rightarrow$ P) has the higher number of significant values (3/8 teams, plus `Total`). Like for acceptance criteria, we dug deeper by testing the correlation between the number of test cases and on-time completion, but this was the case only twice and all correlations were weak. Hence, **there is no reason to assume that there is a correlation between the number of test cases and the on-time completion of a US (F6)**.

E. Interviewees' answers: time USs spend in various states

Questions two through four from the post-testing interviews (see Sect. III) were concerned with the time spent on USs, but were not related to any of the three independent variables to avoid leading questions and confirmation bias.

For the question "*in which state does a US generally spend the most time?*", all participating teams agreed on the states `Undefined` and `Defined`, as refinement takes the most time and after that, depending on priorities, it may take a while for USs to get planned (ABCDGH). The majority of teams state that this is caused by USs remaining on the backlog due to lack of priority over other USs (ABCDG). In addition, during refinement, input is needed from multiple team members, and sometimes even multiple teams or departments. Questions are asked back and forth, which also takes time (H).

In response to "*what is the reason some USs spend much more time in a specific state?*", all teams said that the main reason is adding requests to their backlog immediately, but not necessarily refining right away, let alone planning (ABCDGH). Charlie was asked what contributes to quick acceptance of USs, as they have the fastest average in this phase. They explained they organize knowledge sharing sessions for team members to be up to speed; also, they group USs by topic in order to clarify what must be tested and accepted (C). Bravo also scored above average in this phase; they explained having a rule that a US must be accepted within the same sprint (B).

Knowing what is expected of you as a developer is the most important factor for on-time completion of USs (F7), according to the participating teams (ABCDG) when asked "*what do you think improves the quick development of USs?*". Among influential factors, the participants mentioned the inclusion of acceptance criteria (A), developers' experience and familiarity with the product (AGH), as well as writing small-enough USs (ABG). Bravo re-evaluates their velocity and story point capacity every spring, based on available developers, every sprint to plan work according to the resources; this requires estimable USs (B). Managing dependence on other teams or being able to work independently from other teams is a factor too (CH). Participating teams also argued about the difference between high and low code, stating that high code

requires more documentation, while in low code most of the documentation is contained within the product (D). On the other hand, in high code, there is more freedom in releasing and there is not always a need to wait until the end of a sprint (G). Finally, one team suggested that their goal of minimal documentation allows them to save time (D).

F. Other qualitative findings

Finally, all participating teams were shown qualitative findings gathered from their USs while assessing their quality. We list findings related to the USs of three or more teams.

Seven teams included maintenance work in USs. An example is "*As a [developer], I want to update [application x] to version [y.z].*" Other examples of described maintenance work include code refactoring and monitoring the performance of a system or application. Interviewees stated that they were unsure where else to include maintenance tasks in a way that makes them easy to find and track (CDGH). Moreover, Delta stated that every change they make is required to have an audit trail, including maintenance tasks such as upgrading to newer versions. These tasks are also considered documentation for testing; when updating part of the system, they need to check if all functionality still works as intended (D).

Five teams recorded user feedback and discussions with other team members in USs, mostly in the form of screenshots from chats or e-mail threads. While not included in the US itself, this information 'hides' the US itself, which is surrounded by screenshots and extra text. This aligns with the most common cause for violating the 'minimal' criterion: adding text in parentheses at the end of the USs. Four teams added extra information, such as "*this is different from what we have now*", "*for this we can use [x]*" or "*contact [colleague]*".

Other observations shared by three teams were the use of application names as US roles, referring to other documentation, including bug reports, describing administrative tasks (e.g., planning meetings, reserving time for tasks), and defining requirements by negation. Hotel stated they use USs for administrative tasks because their US set is their to-do list, akin to post-its discussed in their daily scrum meetings; planning a meeting with someone is a to-do item that needs to be checked off (H). Two teams were aware of their requirements by negation (AC). Examples are "*I want to not be able to do this*" and "*I want to not have to do [specific action] twice*".

VIII. Discussion

We first review the validity threats, using the four types presented by Yin for case study research [41]. Then, we discuss opportunities for research and industry.

A. Threats to validity

Construct validity. We gained a contextual understanding of how the teams worked through the pre-testing interviews. We also used these interviews to ensure that the teams used the studied variables in similar ways. We used the post-testing interviews to verify if their use of artifacts had not changed.

US quality is not limited to the four quality criteria we measured. To maximize objectivity, and due to confidentiality and time constraints regarding the data, we included three syntactic criteria from QUS ans one pragmatic criterion, but excluded the semantic criteria. The USs were also analyzed after completion using their final formulation, while it may have undergone changes. However, according to interviewees, most changes take place during refinement (see Sect. VII-E). Our main results pertain to the `In-Progress` state, so our findings remain unaffected. We also excluded unfinished USs, because they did not have time data for all states. The contents of acceptance criteria and test cases were not evaluated, we only considered their quantity. Due to the heterogeneity with which they were recorded in our data, we found that we could not reliably and consistently assess their quality, as we would have had to use different evaluation methods and criteria. US size was also not in scope, as we found that story points were not used consistently and, even if they were, the sizes of story points used by every team do not necessarily follow an interval scale, precluding comparisons. We operationalized our other variables (number of acceptance criteria, of test cases, and time) in an objective manner and used triangulation to confirm our findings.

Internal validity. The quantitative data was gathered over a longer timer period (nine months); thus, both the team members and the adopted guidelines may have changed. These, besides other uncontrolled factors such as dependencies across teams, may affect internal validity, but are essential characteristics of case studies, which aim at "*studying a contemporary phenomenon in its real-life context*" [17]. These uncontrolled factors, besides the co-evolution of our variables, made us opt for correlation over regression. To determine if correlations may indicate a causation, we relied on triangulation. Finally, most of the significant results in Sect. VII had low statistical power and some were ambivalent. In the latter case, we reported our results conservatively.

External validity. Although we cannot assess the extent to which results from this organization can be generalized to others, the participating teams did use popular methods (i.e., Scrum) and templates (i.e., Connextra's US template). The teams were also semi-randomly selected, only ensuring they belonged to different ARTs. It seems that the `In-Progress` state had the most reliable data, as this showed a smaller variance than the other states (see Table VI). A possible explanation is that this state is the basis for the organization to measure team performance. We used this contextual characteristic to our advantage by studying this state in more detail.

Reliability. The four US quality criteria were evaluated by a single researcher, discussing edge cases with a second researcher. Guidelines, however, were created beforehand and applied to all USs (see Sect. V-A), and we limited our analysis to mostly objective criteria. Previous research shows that the well-formed and atomic criteria can be assessed consistently, as shown by the high inter-rater reliability [35]. In addition, reliability is supported by the fact that both researchers have been familiar with the QUS framework since its inception,

one of them being a co-creator of QUS. We described the interview questions, preprocessing activities and all the results of statistical testing to improve transparency. USs and their time information could not be shared due to confidentiality agreements, but we have made data available for IV, MV_1, and MV_2, so that the tests for RQ_1 and RQ_2 can be verified.

B. Research directions

The post-testing interviews revealed out-of-scope factors that may affect the studied variables. It would be beneficial to consider these or mitigate them, but factors such as the experience of team members and the degree to which the teams in question are dependent on others may be hard to quantify.

As stated earlier, only the final artifacts were evaluated in this study, so it remains unclear how teams arrive at a 'high'-quality US. In addition, not all QUS criteria were evaluated in this study, while these may also impact the fulfillment of USs. The included criteria were assessed manually, but large language models (LLMs) may support the tagging of more data. We were unable to do so due to the sensitive information in the USs and we could not use a local LLM either.

We did not evaluate the quality or the content of acceptance criteria or test cases, although it is possible that their quality could also affect the fulfillment of a US. Moreover, while there are indicators that the use of acceptance criteria positively affects the on-time completion of a US, these results are not yet sufficiently conclusive. We do not know whether, for example, if there is an optimal number of acceptance criteria for a US.

Efficiency is not the only indicator of team performance. In this study, we excluded performance indicators such as quality of the delivered work due to our focus on time variables, but those are important directions for future studies.

The teams included in this study were eager to improve their work, but are limited by their available time; lightweight tools and solutions may be more suitable for them. Finally, interviewees expressed a need for tools or methods to document maintenance tasks and non-functional requirements.

C. Opportunities for industry

Practitioners were unfamiliar with the QUS framework, but were interested in learning about and experimenting with the criteria used in this study. Another noteworthy finding is that POs/SMs were sometimes unaware of what is included in their template and especially why. It seems that additions are often made to templates, but unused sections are rarely removed. This may be confusing to (newer) team members and it is reasonable to assume the findability of other information is hampered by large templates. We recommend POs and SMs to re-evaluate their templates regularly.

Furthermore, according to the results presented in Sect. VII, the use of acceptance criteria can be regarded as a best practice. Practitioners may also consider using other techniques that improve the understanding team members have of the issue at hand, since this is the most important factor in completing USs on time according to our interviewees. Examples are increasing the shared understanding within a team through example mapping [42] and the Requirements Specification for Developers approach, which aims at making requirements specifications more suitable for activities performed by developers (e.g., writing code, testing) [43].

Finally, teams might be able to save time on simple and tedious tasks, such as assessing US quality or completeness, by making use of AI.

IX. Conclusion

We have investigated whether the quality of USs has an effect on time efficiency. We have collected USs of eight Agile teams in a large organization and we considered information on their intrinsic quality (using part of the QUS framework), the number of associated acceptance criteria and test cases, and multiple variables regarding cycle time. In our case study, we performed triangulation by combining quantitative and qualitative methods. Quantitatively, we checked for the existence of correlations between US quality and time. Qualitatively, we discussed the findings with POs and SMs from the participating teams and asked them questions on the studied requirements artifacts. Below, we answer our research questions based on the results described earlier in this paper, referring to the specific findings presented in Section VII.

RQ_1. To what extent does the quality of a US affect the number of associated acceptance criteria? The statistical tests showed significance only for 2 of the 8 teams, with weak strength. None of the interviewees suggested that there may be a link, leading us to conclude that the quality of a US does not affect the existence of acceptance criteria (F1).

RQ_2. To what extent does the quality of a US affect the number of associated test cases? For this correlation, we could only obtain significance for the dataset as a whole, but the correlation strength is negligible. The interviewees made no mention of such a relationship either, so we conclude that the quality of a US does not affect the existence of test cases (F1).

F1 led us to considering the number of acceptance criteria and test cases as independent variables for the following analysis (changing our original hypothesis in Fig. 2).

RQ_3. To what extent does the quality of a US affect the time spent on that US? The interviewees believe that violating some quality criteria may increase efficiency. For instance, they specify multiple features in one US, violating the atomic criterion, to save time on administrative work (F2). In addition, none of the interviewees suggested that the quality of a US is important for quick development/completion of USs. Adhering to the four quality criteria investigated in this study is not a prerequisite for communicating well with the team, according to the interviewees. Our quantitative results also show there is no reason to assume that a higher quality US is fulfilled more quickly than a lower quality US (F3). We also specifically tested the relationship between atomic violations (the most frequently observed quality criterion violation; 19.7% of the analyzed USs) and the time a US spent in the `In-Progress`

state. Again, we find no evidence to support this relationship, which is in line with what the interviewees described.

RQ_4. To what extent does the use of acceptance criteria affect the time spent on a US? Five out of six interviewees agreed that developers understanding what is expected of them is of utmost importance for completing USs on time (F7), and acceptance criteria serve precisely that purpose (F5). This is also supported by literature, which shows that team members mostly rely on documentation and do not regularly speak with the customer [6] and that addressing stakeholder needs is an important indicator of team effectiveness [4]. In our statistical tests, we have also found a correlation between the use of acceptance criteria and the time a US spends in the `In-Progress` state. We observed this correlation in four teams and in the total dataset. To further test this correlation, we analyzed the relationship between the use of acceptance criteria and USs completed on time. Again, there seems to be some correlation, of weak strength, for three teams and for the total dataset. Considering all findings, we conclude that there may be a noteworthy relationship between the existence of acceptance criteria and on-time completion of USs (F4).

RQ_5. To what extent does the use of test cases affect the time spent on a US? Most interviewees believe that test cases are mainly important for testing and not building a US. Our statistical tests returned significant results, of weak strength, for three teams regarding the time a US spends in the `In-Progress` state and test cases, as well as the total dataset. We also studied the effect of test cases and USs completed on time, but found significant results for only two teams and the total dataset, of limited strength. Therefore, we conclude that test cases do not affect on-time completion of USs (F6).

MRQ: To what extent do requirements artifacts affect how efficiently USs are fulfilled? We have found no evidence that quality of USs and the use of test cases affect the efficiency with which USs are fulfilled. Our qualitative and quantitative findings, however, show that investing in the formulation of acceptance criteria plays an important and positive role and influences the on-time completion of USs.

The post-testing interviews highlighted five additional factors that influence how requirements artifacts such as USs, acceptance criteria, and test cases are used: (i) artifacts are mainly used to provide sufficient clarity to the team members, (ii) more experienced team members are suspected to need fewer artifacts and/or less exhaustive artifacts, (iii) dependence on other teams or third-party team members necessitates the use of artifacts, (iv) striving to save time can lead to the creation of fewer artifacts, and (v) organization regulations (e.g., audit trails) may determine which artifacts are created and how extensive they are. Future research could complement our study by investigating whether these factors, mentioned by the interviewees, play a significant role in team efficiency.

Acknowledgments

We would like to thank all the teams that participated in our study for their time and effort.

References

[1] M. Kassab, "The changing landscape of requirements engineering practices over the past decade," in *Proc. of the International Workshop on Empirical Requirements Engineering*. IEEE, 2015, pp. 1–8.

[2] Z. Masood, R. Hoda, and K. Blincoe, "Real world scrum a grounded theory of variations in practice," *IEEE Transactions on Software Engineering*, vol. 48, no. 5, pp. 1579–1591, 2020.

[3] N. Kurapati, V. S. C. Manyam, and K. Petersen, "Agile software development practice adoption survey," in *Proc. of the International Conference on Agile Software Development*. Springer, 2012, pp. 16–30.

[4] C. Verwijs and D. Russo, "A theory of Scrum team effectiveness," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 3, pp. 1–51, 2023.

[5] P. Hruschka, K. Lauenroth, M. Meuten, G. Rogers, S. Gärtner, and H.-J. Steffe, "RE@Agile Handbook," International Requirements Engineering Board, Handbook, May 2024.

[6] A. Hess, P. Diebold, and N. Seyff, "Understanding information needs of agile teams to improve requirements communication," *Journal of Industrial Information Integration*, vol. 14, pp. 3–15, 2019.

[7] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "The use and effectiveness of user stories in practice," in *Proc. of the International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 2016, pp. 205–222.

[8] M. Cohn, *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.

[9] 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.

[10] L. Cao and B. Ramesh, "Agile requirements engineering practices: An empirical study," *IEEE Software*, vol. 25, no. 1, pp. 60–67, 2008.

[11] R. K. Mallidi and M. Sharma, "Study on agile story point estimation techniques and challenges," *International Journal of Computer Applications*, vol. 174, no. 13, pp. 9–14, 2021.

[12] O. Malgonde and K. Chari, "An ensemble-based model for predicting agile software development effort," *Empirical Software Engineering*, vol. 24, pp. 1017–1055, 2019.

[13] P. Heck and A. Zaidman, "A systematic literature review on quality criteria for agile requirements specifications," *Software Quality Journal*, vol. 26, pp. 127–160, 2018.

[14] B. Wake, "INVEST in Good Stories, and SMART Tasks," http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/, accessed 12-03-2025, 2003.

[15] G. Lucassen, F. Dalpiaz, J. M. E. van der Werf, and S. Brinkkemper, "Improving agile requirements: the Quality User Story framework and tool," *Requirements Engineering*, vol. 21, pp. 383–403, 2016.

[16] A. R. Amna and G. Poels, "Systematic literature mapping of user story research," *IEEE Access*, vol. 10, pp. 51 723–51 746, 2022.

[17] C. Wohlin, "Case study research in software engineering—it is a case, and it is a study, but is it a case study?" *Information and Software Technology*, vol. 133, 2021, article nr. 106514.

[18] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87–108, 2016.

[19] M. Cohn, "How detailed should a user story be?" accessed 12-03-2025, https://www.mountaingoatsoftware.com/blog/what-level-of-detail-should-be-captured-in-a-user-story.

[20] R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, and F. G. de Oliveira Neto, "Requirements engineering challenges and practices in large-scale agile system development," *Journal of Systems and Software*, vol. 172, p. 110851, 2021.

[21] A. T. van Can and F. Dalpiaz, "Requirements information in backlog items: Content analysis," in *Proc. of the International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 2024, pp. 305–321.

[22] T. Sedano, P. Ralph, and C. Péraire, "The product backlog," in *Proc. of the International Conference on Software Engineering*. IEEE, 2019, pp. 200–211.

[23] C. d. O. Melo, D. S. Cruzes, F. Kon, and R. Conradi, "Interpretative case studies on agile team productivity and management," *Information and Software Technology*, vol. 55, no. 2, pp. 412–427, 2013.

[24] M. Trzeciak and P. Banasik, "Motivators influencing the efficiency and commitment of employees of agile teams," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 8, no. 4, p. 176, 2022.

[25] Y. Lindsjørn, D. I. Sjøberg, T. Dingsøyr, G. R. Bergersen, and T. Dybå, "Teamwork quality and project success in software development: A survey of agile development teams," *Journal of Systems and Software*, vol. 122, pp. 274–286, 2016.

[26] A. Poth, M. Kottke, and A. Riel, "Evaluation of agile team work quality," in *Proc. of Agile Processes in Software Engineering and Extreme Programming Workshops*. Springer, 2020, pp. 101–110.

[27] S. L. Ramírez-Mora, H. Oktaba, and J. Patlán Pérez, "Group maturity, team efficiency, and team effectiveness in software development: A case study in a CMMI-DEV Level 5 organization," *Journal of Software: Evolution and Process*, vol. 32, no. 4, p. e2232, 2020.

[28] L. Przybilla, M. Wiesche, and H. Krcmar, "The influence of agile practices on performance in software engineering teams: A subgroup perspective," in *Proc. of the ACM SIGMIS Conference on Computers and People Research*, 2018, pp. 33–40.

[29] D. Strode, T. Dingsøyr, and Y. Lindsjorn, "A teamwork effectiveness model for agile software development," *Empirical Software Engineering*, vol. 27, no. 2, p. 56, 2022.

[30] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, and J. Still, "The impact of agile practices on communication in software development," *Empirical Software Engineering*, vol. 13, pp. 303–337, 2008.

[31] C. Melo, D. S. Cruzes, F. Kon, and R. Conradi, "Agile team perceptions of productivity factors," in *Proc. of the 2011 Agile Conference*. IEEE, 2011, pp. 57–66.

[32] M. Ochodek and S. Kopczyńska, "Perceived importance of agile requirements engineering practices–a survey," *Journal of Systems and Software*, vol. 143, pp. 29–43, 2018.

[33] C. Tona, S. Jiménez, R. Juárez-Ramírez, R. G. Pacheco López, Á. Quezada, and C. Guerra-García, "Scrumlity: an agile framework based on quality of user stories," *Programming and Computer Software*, vol. 48, no. 8, pp. 702–715, 2022.

[34] N. Verhoeven, *Doing research: The Hows and Whys of Applied Research*, 5th ed. Boom uitgevers Amsterdam, 2019.

[35] J. Wouters, A. Menkveld, S. Brinkkemper, and F. Dalpiaz, "Crowd-based requirements elicitation via pull feedback: method and case studies," *Requirements Engineering*, vol. 27, no. 4, pp. 429–455, 2022.

[36] M. Wynne, A. Hellesoy, and S. Tooke, *The cucumber book: behaviour-driven development for testers and developers*. The Pragmatic Programmers LLC, 2017.

[37] Laerd Statistics, "Spearman's rank-order correlation using spss statistics," accessed 12-03-2025, https://statistics.laerd.com/spss-tutorials/spearmans-rank-order-correlation-using-spss-statistics.php.

[38] A. Field, Z. Field, and J. Miles, *Discovering statistics using R*. Sage, 2012.

[39] C. P. Dancey and J. Reidy, *Statistics without maths for psychology*. Pearson, 2007.

[40] S. Molenaar and F. Dalpiaz, "Online appendix of 'The impact of requirements artifacts on efficiency in agile development: A case study'," Sep. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.14990047

[41] R. K. Yin, *Case study research: Design and methods*, 4th ed. Sage, 2009, vol. 5.

[42] J. Berends and F. Dalpiaz, "Refining user stories via example mapping: an empirical investigation," in *Proc. of the International Requirements Engineering Conference*. IEEE, 2021, pp. 345–355.

[43] J. Medeiros, A. Vasconcelos, C. Silva, and M. Goulão, "Requirements specification for developers in agile projects: Evaluation by two industrial case studies," *Information and Software Technology*, vol. 117, 2020, article nr. 106194.