The Use and Effectiveness of User Stories in Practice

Garm Lucassen, Fabiano Dalpiaz(✉), Jan Martijn E.M. van der Werf, and Sjaak Brinkkemper

Utrecht University, Utrecht, The Netherlands
{g.lucassen,f.dalpiaz,j.m.e.m.vanderwerf,s.brinkkemper}@uu.nl

Abstract. [**Context and motivation**] User stories are an increasingly popular textual notation to capture requirements in agile software development. [**Question/Problem**] To date there is no scientific evidence on the effectiveness of user stories. The goal of this paper is to explore how practicioners perceive this artifact in the context of requirements engineering. [**Principal ideas/results**] We explore perceived effectiveness of user stories by reporting on a survey with 182 responses from practitioners and 21 follow-up semi-structured interviews. The data shows that practitioners agree that using user stories, a user story template and quality guidelines such as the INVEST mnemonic improve their productivity and the quality of their work deliverables. [**Contribution**] By combining the survey data with 21 semi-structured follow-up interviews, we present 12 findings on the usage and perception of user stories by practitioners that employ user stories in their everyday work environment.

1 Introduction

User stories [6] are a popular method for representing requirements using a simple template such as *"As a ⟨role⟩, I want ⟨goal⟩, [so that ⟨benefit⟩]"*. Their adoption is growing [14], and is massive especially in the context of agile software development [29]. Despite their popularity, the requirements engineering (RE) community has devoted limited attention to user stories both in terms of improving their quality [21] and of empirical studies on their use and effectiveness.

The purpose of this study is to go beyond anecdotal knowledge and gather scientifically rigorous data on the use and perception of user stories in industry. This includes data on the development methods they are used in, the templates for structuring user stories, and the existing quality guidelines. Additionally, we explore whether practitioners perceive an added value from the use of user stories: Do they increase productivity? Do they ameliorate work deliverable quality?

Earlier studies have shown that RE practices play a central role in agile development [11,29] albeit on a small scale and in a local context. Ramesh, Cao and Baskerville pinpointed agile RE practices and challenges by studying 16 organizations [26] but they have not studied the role of user stories in detail.

M. Daneva and O. Pastor (Eds.): REFSQ 2016, LNCS 9619, pp. 205–222, 2016.
DOI: 10.1007/978-3-319-30282-9_14

Other works studied the effectiveness of RE practice and artifacts through experiments [7–9,24] as well as the use and perception of practitioners [1,12].

This paper describes our conducted empirical research, which includes an online survey followed by semi-structured interviews with a subset of the survey respondents. Key findings of our analysis include the strong link between Scrum and user stories, the widespread adoption of the user story template proposed by Connextra, the perception that user stories help practitioners define the *right* requirements, the crucial role of explaining *why* a requirement is expressed, and a positive evaluation of quality frameworks by respondents that use one.

The remainder of this paper is structured as follows. Section 2 presents our research questions and describes the design of our empirical study. Section 3 analyzes the survey and interview results concerning the *use* of user stories in practice, while Sect. 4 reports on the *perceived effectiveness*. Section 5 discusses validity threats to our research, while Sect. 6 reports on related literature. We discuss our results and conclude in Sect. 7.

2 Study Design

The goal of this study is to understand how practitioners use and perceive user stories, which prompts us to formulate two research questions:

$\mathbf{RQ_1}$: How do Practitioners use User Stories? We investigate the context of user stories by looking at how practitioners approach working with user stories. What software development methods are appropriate for using user stories? Which templates and quality guidelines are popular among practitioners?

$\mathbf{RQ_2}$: How do Practitioners Perceive the Effectiveness of User Stories? In this study, we decompose effectiveness into productivity and quality of work deliverables; although many more aspects exist, these are two basic performance indicators for software development processes. We examine whether practitioners agree that user stories increase their work productivity and/or the quality of their work deliverables. Additionally, we investigate whether practitioners find that utilizing a template and/or a quality framework further improves these aspects.

To answer these research questions, we split our study design in two stages: (1) we conduct an online survey that we distribute worldwide among software professionals to collect quantitative information from practitioners on the use of user stories and their added value for RE, and (2) we perform follow-up interviews to gather clarification of the answers of a selected sample of survey respondents, improving our understanding of the survey findings.

The authors distributed the survey over a variety of channels including the professional network of the authors and online communities such as requirements engineering and software engineering mailing lists, Twitter, Hacker News and Reddit Agile. Over a span of two weeks, from July 7 2015 until July 21 2015, the survey obtained 197 responses. 49 survey respondents were invited to participate in a follow-up interview, 21 of which accepted and contributed with more in-depth, qualitative data on the subject.

We analyzed the survey responses using SPSS, Excel and R; we transcribed the follow-up interviews and categorized them using the qualitative data analysis tool Nvivo.

2.1 Research Protocol

The goal of the survey is to gather quantitative data on how practitioners use and perceive user stories. To achieve this goal, we formulated 21 questions that are available in our online appendix [20]. After a short introduction on our research, the survey asked five questions on the respondent's demographics and organizational context, followed by six questions on their usage of and experience with user stories, templates and quality guidelines. Next, respondents were asked to indicate whether they agree or disagree with the following six Likert-Type statements, which we reference by their number throughout the paper:

$\mathbf{S_1}$ Using user stories increases my productivity
$\mathbf{S_2}$ Using user stories increases the quality of my work deliverables
$\mathbf{S_3}$ Using a template for my user stories further increases my productivity
$\mathbf{S_4}$ Using a template for my user stories further increases the quality of my work deliverables
$\mathbf{S_5}$ Using a quality framework for my user stories further increases my productivity
$\mathbf{S_6}$ Using a quality framework for my user stories further increases the quality of my work deliverables

Finally, the respondents could optionally provide their contact details and comment on the research and the survey. The survey has been reviewed by two academics who are not part of the authors and was piloted with three practitioners: a developer, a designer and a project manager. Based on the pilot, we revised the survey by adding six questions, removing one question, changing the order of existing questions and making three questions optional.

The goal of the follow-up interviews is to capture the respondent's rationale behind the answers they provided in the survey. The interview protocol consists of 16 questions (see [20]). After the preliminaries, the interviewee was asked to explain the role of user stories in their organization and their general perspective on user stories. Next, the respondent was asked to explain the difference between a poor and good user story in his opinion and to clarify their answers to the Likert-type statements $\mathbf{S_1}$–$\mathbf{S_6}$.

2.2 Survey Respondents

Because we posted links to the survey on public venues, it is practically impossible to measure how many individuals we reached. The survey website page garnered 598 unique page views. Google Analytics defines this as *"Unique Pageviews is the number of sessions during which the specified page was viewed at least once"*. These page views led to 197 submitted responses; 6 of them were duplicates, while others contained impossible or invalid answers such as unclear experience or respondents claiming to be working with user stories since before the year of their introduction. In total, we retained 182 valid responses.

2.3 Follow-Up Interview Respondents

Out of the 119 respondents (65 %) who supplied their email address at the end of the survey, the authors identified 49 respondents that could potentially provide *opinionated* answers during a follow-up interview. We invited all respondents that either (i) provided very positive or very negative answers, (ii) gave varied answers to the Likert-type questions, or (iii) added a comment at the end of the survey. In total, 21 respondents participated, leading to a response rate of 43 %.

This group of respondents is quite diverse and its composition differs from that of the survey's respondents. Notable differences are that more practitioners participated that work in consultancy (9/21) and/or have the role of requirements engineer/business analyst (6/21). The average interviewee has 6 years of experience with user stories. Respondents originated from 7 different countries; 11 from the Netherlands, 5 from the United States of America, the remaining 5 were all from different countries: Argentina, Brazil, Canada, Portugal and the United Kingdom.

3 User Story Usage

This section reports on data collected related to $\mathbf{RQ_1}$ on the use of user stories by practitioners. We examine and report on the first part of the survey results and highlight specific findings from the follow-up interviews. Our twelve key findings are marked within the text as $\mathbf{F_1}$–$\mathbf{F_{12}}$.

3.1 Respondent Context

As recommended by Cohn [6], user stories are primarily used in combination with Agile methods. Scrum in particular is used by the majority of respondents. We asked respondents to indicate both which software development methods they used in general, and in which methods they employed user stories. The majority indicate they work with Scrum (94 %), but Kanban (40 %) and waterfall (29 %) are popular as well. XP (13 %), V-Model (7 %), Spiral (3 %) and 14 other methods (9 %) are considerably less common. Responses to this question accentuate the tight coupling of user stories with Agile methods: 99 % of respondents that work with Scrum employ user stories - all respondents but two ($\mathbf{F_1}$). As one follow-up interviewee noted: *"For me, user stories and Scrum are interconnected"*. Indeed, 17 out of 21 interviewees mention Scrum without it being a subject of discussion. Kanban and XP have a tight coupling as well: 79 % and 83 % of the respondents that use these software development methods do employ user stories. However, none of the interviewees mention either method during the interview. Users of waterfall and the V-model do not employ user stories often: 21 % and 31 % of them do so.

On average, respondents had 4 years of experience with user stories; 57 of them (31 %) had more than 5 years of experience. On average, the organizations of the respondents were working with user stories for slightly longer, 4.4 years; 64 (35 %) organizations were working with user stories for more than 5 years. Respondent roles include product manager (29 %), developer (21 %), requirements engineer (18 %), software architect/CTO (8 %), project manager (8 %) and other (16 %). Respondents work for fairly uniform organization types: software product (51 %), consultancy (20 %), custom software (19 %) and other (10 %). The organization sizes, however, are quite diverse: 1–9 (12 %), 10–49 (20 %), 50–249 (27 %), 250–499 (8 %) and 500+ (33 %).

Additionally, we asked respondents to self-assess their skill level. The average years of experience per skill level are as follows: Beginner - 1.91 (n = 34), Intermediate - 3.05 (n = 77), Advanced - 4.76 (n = 49), and Expert - 8.95 (n = 22). Surprisingly, the aggregate of our respondents did not fall victim to the Dunning-Kruger effect; a cognitive bias which causes individuals with low skill to overestimate their ability and performance in comparison to their highly skilled peers - and vice versa [15].

3.2 The Role of User Stories

After introductions, the first question of each follow-up interview was to describe the role of user stories in the interviewee's organization. In our 21 interviews, we collected as many different accounts of the role of user stories in their organization. The interviewees explanations range from very close to the approach described Cohn's book [6] to adaptations that are rather far from agile software development. The majority of interviewees, however, are somewhere in between because they have adapted user story theory to their own situational context. Nevertheless, all approaches have one crucial aspect in common: the user story is the most granular representation of a requirement that developers use to build new features.

3.3 Template

The use of a template when writing user stories can be considered standard industry practice - only 27 respondents (15 %) indicate they do not use a template. The most popular template is the 'original' one [6]. 59 % of respondents utilize the Connextra template ($\mathbf{F_2}$): *"As a ⟨role⟩, I want ⟨goal⟩, [so that ⟨benefit⟩]"*. An additional 10 % of respondents use the identical template, but without the *"[so that ⟨benefit⟩]"* clause. The remaining 32 respondents (18 %) are spread between 15 approaches, none of which have a significant share. One of these template omits the role, including only the what and the why.

In the follow-up interview, respondents were asked to explain whether they have a specific reason for using the template they use. Out of the 19 interviewees that use a template just one decided to study and select the most appropriate template for his situation. The remaining interviewees were taught or heard of a specific template at some point and never encountered the need to change to another template. This is likely a factor in explaining the prevalence of the Connextra template.

3.4 Quality Guidelines

The use of quality guidelines is commonplace among practitioners. The most well-known framework is INVEST [28], which posits that a good user story has the following characteristics: Independent, Negotiable, Valuable, Estimatable, Small and Testable. 33 % of respondents indicate they follow self-defined quality guidelines when writing user stories, while 23.5 % use the standardized INVEST approach. 39.5 % of respondents do not validate their user stories with any form of quality guidelines. The remainder use alternatives, or indicate that it depends on the situation (4 %).

When asked to explain what their self-defined quality guidelines entail, all 10 interviewees admit they do not have a well-defined, structured list of concerns they consult when writing user stories. Instead, they rely on the experience of the user story writer and multiple rounds of peer review to ensure the quality of their user stories. Interviewees that do not use quality guidelines, indicate this is not a conscious decision but rather that they are not aware of quality guidelines like INVEST ($\mathbf{F_3}$).

4 Perception of User Story Effectiveness

This section investigates $\mathbf{RQ_2}$: how practitioners perceive the effectiveness of user stories. We examine the second part of the survey to report on how practitioners perceive the impact of user stories, templates and quality guidelines in terms of their productivity and work deliverable quality.

As expected, the collected data is not normally distributed, making parametric statistics that rely on testing means inappropriate for our Likert-type questions [5]. Instead, we treat the answers as ordinal data. To report on central tendency and variability of ordinal data, Boone and Boone [4] recommend using the median or mode and frequencies. To confirm that the variability is from independent populations, Boone and Boone recommend using the statistical χ-square test for independence. Throughout the remainder of this section, applying this test enables us to determine whether a specific variable influences the outcome of the Likert-type questions.

4.1 User Stories in Isolation

Both the median and mode of the Likert-style questions indicate that practitioners *agree* that representing requirements as user stories and following a template increase their work productivity and deliverable quality. For quality guidelines, both median and mode are *neutral* for gained productivity and quality. For more insights regarding practitioners' opinion on user stories we examine the frequency distributions in Fig. 1. In the subsequent subsections, we analyze

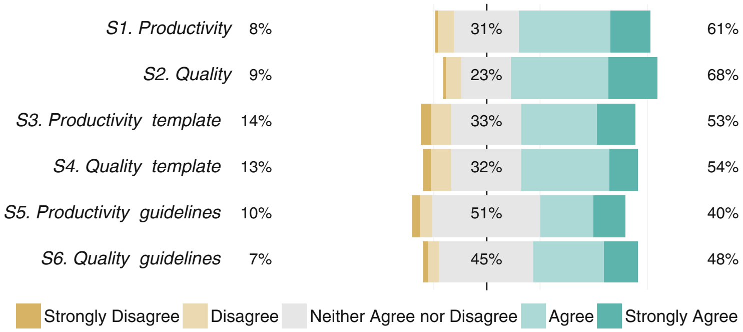

Fig. 1. Perception of user story effectiveness. The shown percentages (left-to-right) refer to Strongly Disagree + Disagree, Neither Agree nor Disagree and Agree + Strongly Agree, respectively. The same format is used in the following charts

specific slices of the data using frequency distributions and the χ-square test for independence.

Examining the frequency distribution of our respondents' answers one observation stands out: only a fraction of respondents perceive user stories, templates and quality guidelines to be detrimental to their work productivity and deliverable quality (the percentages on the left of Fig. 1 are all between 7 % and 14 %). Even when we consider neutral answers as negative, the majority of respondents agree or strongly agree that user stories and templates improve work productivity ($\mathbf{S_1}$: 61 %, $\mathbf{S_3}$: 53 %) and quality ($\mathbf{S_2}$: 68 %, $\mathbf{S_4}$: 54 %). Respondents are ambivalent about quality guidelines: 51 % and 45 % indicate they neither agree nor disagree that quality guidelines improve work productivity $\mathbf{S_5}$ or quality $\mathbf{S_6}$. During the follow-up interviews, respondents were asked to clarify their answers. From their comments on user stories in general, we present the following common sentiments to show how the interviewees perceive user stories.

The Right Software ($\mathbf{F_4}$): 10 interviewees mention that user stories are an enabler for developing the *right* software. In their experience, the technical quality of software does not improve by using user stories and neither do they directly impact the speed of software development. In fact, user stories require more work upfront because the stakeholders have to decompose a requirement into small, comprehensible chunks. This decomposition, however, forces all stakeholders to think and talk about the details of a requirement. This builds a common understanding within the team of what the end-user expects of the software. Thanks to the identification of the *right* requirements, developers are enabled to create the *right* software. According to the literature, this may prevent defects which cost 10–200 times as much to correct later in the software development lifecycle [3,22]. One interviewee reported that user stories force developers to meet the customer numerous times, resulting in code that is very close to customer expectations. This improves productivity, despite the significant amount of time that is devoted to interacting with the customer.

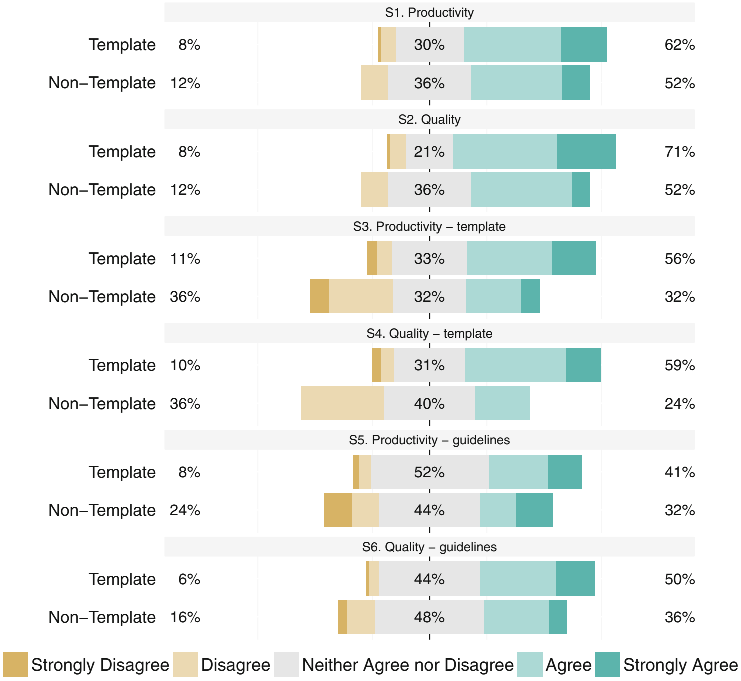

Fig. 2. Perception of respondents that use a template and those that do not.

"User Stories Optimize for Happiness" [2] **(F_5):** 5 interviewees do not view productivity or quality gains as essential contributions of user stories. Other aspects of agile development methods have a bigger impact on these concerns. The real advantage of user stories is that stakeholders enjoy working with user stories, fostering a pleasant work environment.

4.2 The Role of Using a Template

The first data slice we examine concerns respondents that follow a template for user stories (n = 155) versus those who do not (n = 27). The frequency distributions in Fig. 2 show that respondents using a template more often agree that user stories improve work deliverable quality (71 % vs. 52 %). However, because the two populations are not independent in a statistically significant manner ($\alpha = .187$), we cannot claim that respondents who use a template are more positive towards user stories.

For the statements on the impact of templates on work productivity ($\mathbf{S_3}$) and quality ($\mathbf{S_4}$), the populations are statistically independent with α's of .02 and .00. Indeed, the difference is striking on both the negative (11 % and 9 % vs. 33 % and 37 %) and positive (57 % and 60 % vs. 30 % and 22 %) sides of the distribution. These results indicate that respondents that use templates agree considerably more often that using a template contributes to productivity and work deliverable quality. The question is, however, if this difference is an objective judgment or is rather due to the fact that the respondents are persuaded by the choice of using a template. During the follow-up interviews, we asked respondents to clarify why or how they believe that templates contribute to work productivity and quality. They shared the following comments:

***A* Template, not *the* Template ($\mathbf{F_6}$):** 12 interviewees mention the beneficial impact of a standard structure for defining user stories. Recall, however, that in Sect. 2.3 all interviewees but one did not have an explicit motivation for using the template they use. 3 respondents remark that it does not matter which particular template is used. The use of *a template*, any template is what makes the difference. A single, agreed upon template ensures that everyone within a team works in the same way. When a team can rely upon a standardized structure, their alignment improves overall work productivity and quality. This quote by one respondent effectively illustrates why: *"It's not the template that improves quality, it's what we're doing - we're sharing requirements and a template makes that easy to do and more likely that we'll do it"*.

The why is Essential ($\mathbf{F_7}$): While the most popular template for user stories considers the *"[so that ⟨benefit⟩]"* or *why* section as optional, our respondents emphasize the importance of this part for reaping the full rewards of user stories. They attribute a variety of benefits to the inclusion of the purpose of a user story, which lead to work productivity and quality improvements. Adding the *why* part: (1) alleviates confusion among stakeholders, (2) reduces the amount of discussion necessary and (3) provides developers with autonomy in their work. This is, however, easier said than done. The *why* is difficult to find, as the following quote demonstrates: *"Typically, the why question is correctly answered if after the initial answer, you ask 'why?' again for three more times"*.

A developer with a negative opinion of user stories shared that in his experience business people will abuse a template to formulate the same old requirement in a different format. He complained that user stories become *"a blanket way to generally describe what the solution is the company has already defined for you"*, which conflicts with the principle that requirements should be problem-oriented [21,32].

4.3 The Impact of Using Quality Guidelines

The second data slice looks at the perceptions of respondents that follow self-defined quality guidelines (n = 60), INVEST (n = 43) or none (n = 72)[1]. Examining the frequency distributions in Fig. 3, we see that respondents that follow quality guidelines are more positive than those that do not ($\mathbf{F_8}$). The χ-square tests for independence of $\mathbf{S_1}, \mathbf{S_4}, \mathbf{S_5}$ and $\mathbf{S_6}$ are statistically significant; meaning that we can claim that respondents using quality guidelines more often agree that user stories and quality guidelines improve productivity, and templates and quality guidelines further improve work deliverable quality.

[1] Note that 7 responses are excluded. These respondents gave unique 'other' answers, whose samples are too small for statistical analysis.

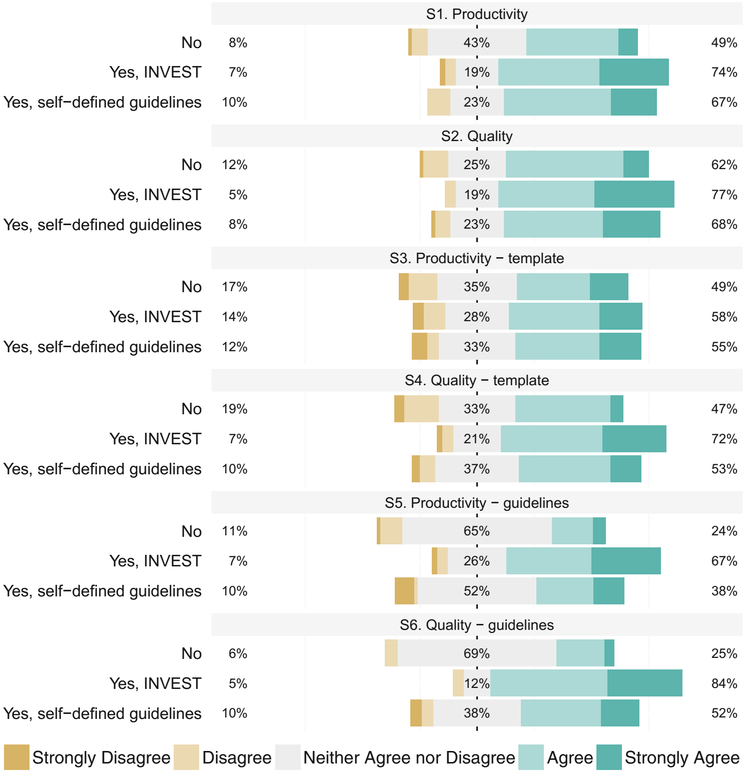

Fig. 3. Perception of respondents that use INVEST, self-defined quality guidelines or none.

The positive attitude of respondents that apply INVEST is remarkable. During the interviews, these respondents were capable of effectively arguing both for and against any productivity and quality gains. Their ideas can be summarized as follows:

INVEST is not a Checklist ($\mathbf{F_9}$): 3 interviewees mention that although the INVEST mnenomic can be used as a checklist, interviewees do not use it as such. Instead, the six characteristics of a good user story are internalized by the team and whenever a user story violates INVEST, a team member brings this up for discussion.

INVEST is Useful for Inexperienced Teams ($\mathbf{F_{10}}$): 2 interviewees indicate they primarily use INVEST as a training tool for inexperienced teams. INVEST's comprehensiveness is an effective starting-point for getting product owners started and the development team to understand how to judge user stories. After two or three months, however, stakeholders have sufficient experience with writing and interpreting user stories that the necessity of INVEST diminishes.

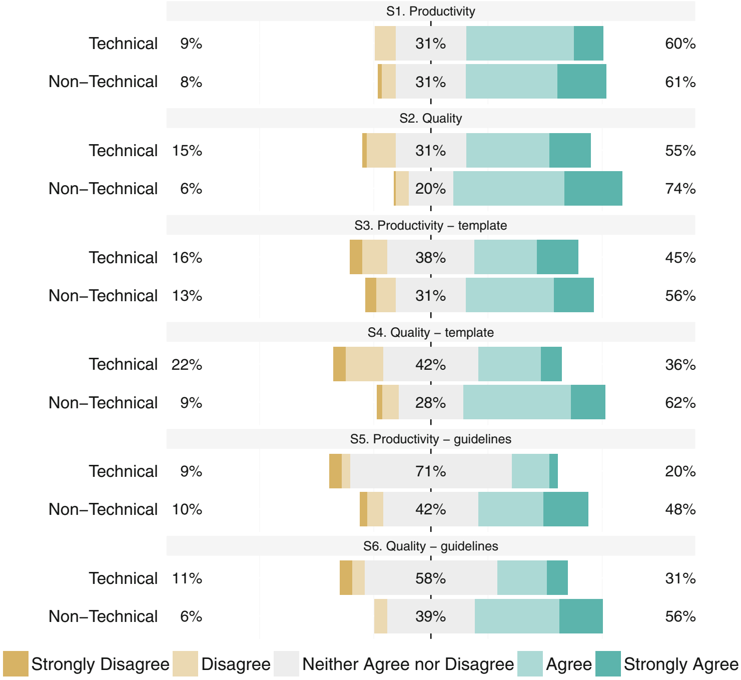

Fig. 4. Perception of respondents with technical and non technical roles.

4.4 Technical Vs. Non Technical Roles

To analyze the difference in perception between technical (n = 55) and non-technical stakeholders (n = 127) we categorize respondents by their role. Because the majority of respondents chose from the pre-defined list of roles, we could easily do this by designating roles containing the term 'software' as technical and those without that term as non-technical. The former primarily consists of developers, software architects and CTOs, while the latter includes everything else such as consultants, product managers and the occasional agile coach.

Approximately 60 % of both stakeholder types agree with $\mathbf{S_1}$ that user stories improve productivity, while for the other 5 statements non-technical stakeholders are considerably more positive (Fig. 4). The average positivity difference between technical and non-technical stakeholders is 22 %. For $\mathbf{S_4}$ ($\Delta = 26\,\%$), $\mathbf{S_5}$ ($\Delta = 28\,\%$) and $\mathbf{S_6}$ ($\Delta = 25\,\%$) the populations are independent with statistical significance (α's of .02, .001 and .003) ($\mathbf{F_{11}}$). During follow-up interviews technical respondents were ambivalent about the impact of user stories on their work productivity and quality. In their experience, software development is not necessarily significantly quicker nor do they encounter less bugs.

4.5 Influence of Expertise Judgement

For one of the contextual questions we asked respondents to self-assess their user story skill level. They could choose from 5 levels of expertise: novice, beginner, intermediate, advanced and expert. Because only 2 people chose novice, for this analysis we counted them as beginners. Studying the frequency distributions in Fig. 5, a pattern catches the eye: as respondents gain more expertise they select *neither agree nor disagree* less frequently, instead opting to agree that work deliverable quality and productivity improves thanks to user stories ($\mathbf{S_1}$ and $\mathbf{S_2}$) and quality guidelines ($\mathbf{S_5}$ and $\mathbf{S_6}$) ($\mathbf{F_{12}}$). This difference is particularly striking when comparing beginners to experts. From a statistical perspective, the answers to $\mathbf{S_1}$ and $\mathbf{S_2}$ on user stories are from independent populations for all four expertise levels. This statistic implies that the difference in their answers cannot be attributed to chance, but that each population has a different perception.

5 Validity Threats

External Validity: Many of the respondents to the survey came from the direct networks of the authors of this paper. Because our research group is focused on the software industry, 93 respondents (51 %) are employed by a product software company. Furthermore, 98 respondents (54 %) are from the Netherlands. Both have the potential to introduce a bias, which would impact the validity of the results. Examining their frequency distributions [20], we see that the percentage differences in the two comparisons are relatively small. Indeed, the χ-square tests for both threats results in significance values between .36 and .78, which is far above the significance threshold of .05. This means that both population pairs are not significantly different and these threats to validity do not hold.

In terms of its composition, the interviewee population is not representative of the survey respondents. In particular, the number of vocally negative interviewees is underrepresented. Although all negative survey respondents were invited for a follow-up interview, there is likely a self-selection process at play. To mitigate this issue, we positively discriminated remarks from negative respondents for inclusion resulting in the abuse paragraph in Sect. 4.2.

Internal Validity: One of the pre-requisites for participating in the survey was that the respondent expresses requirements as user stories. This decision introduces a selection bias for the respondent population. Potential respondents that decided not to employ user stories or stopped employing user stories are excluded from expressing their views. Thus, our results are generalizable only to user story practitioners.

The follow-up interviews were semi-structured. When an interviewee gave a long answer, the interviewer would summarize the answer and confirm with the interviewee if it was correct. In a small number of cases an experimenter bias occurred, including additional information in the summarization, followed by a potential acquiescence bias - better known as yea-saying. When detected during categorization of the transcriptions, these statements have been ignored.

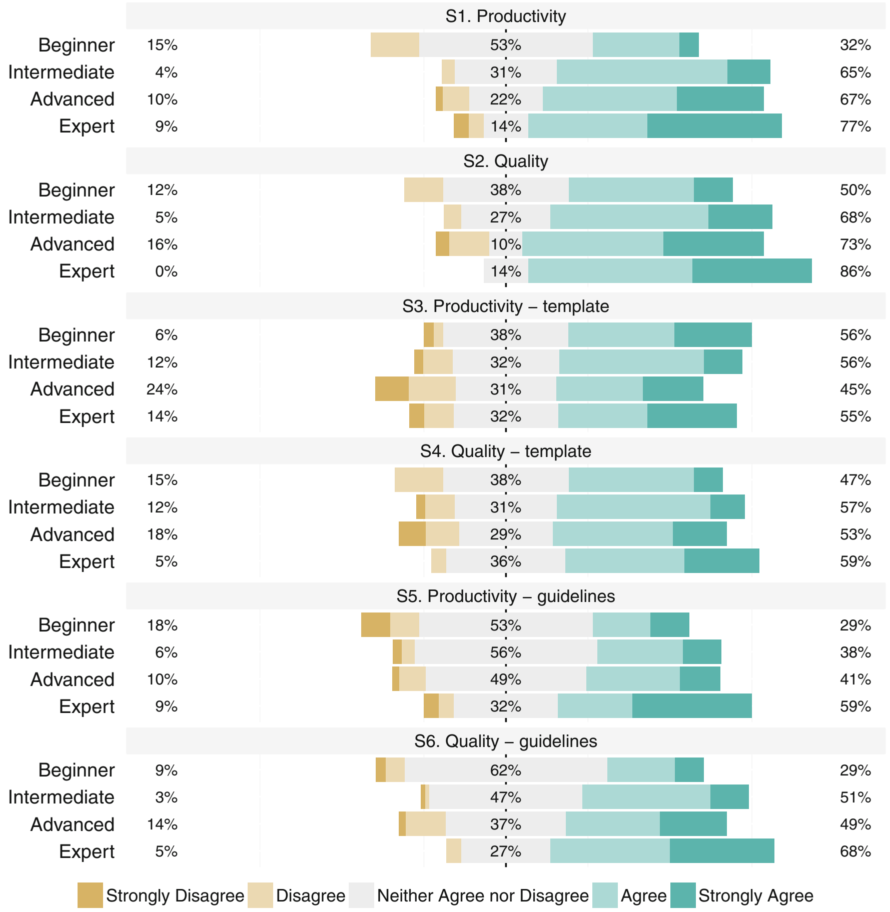

Fig. 5. Attitude differences per expertise level of respondents.

Construct Validity: The survey purposefully did not clearly define what we mean by *productivity* and *work deliverable quality*. Although metrics for quality and productivity in RE exist (e.g., [19]), these metrics were not appropriate for this survey because of our focus on practitioners' *perception*, and there is no general agreement yet on which specific factors do determine these qualities in RE. Additionally, a key phrase in $\mathbf{S_{3-6}}$ was *further* as in "using a template for my user stories *further* increases my productivity". However, it is impossible to confirm that all respondents fully understood the nuance that they were supposed to evaluate *'using a template'* disjoint from the user story concept itself. Although a significant threat to validity, we have reason to believe this does not invalidate the results. When the researcher put extra emphasis on this distinction during the follow-up interviews, none of the respondents indicated they misunderstood the question. Nevertheless, we cannot claim that the

Likert-type questions are 100 % mutually exclusive and exhaustive. Readers should view the survey results as an exploratory evaluation of practitioner's perception of user stories.

The survey contained questions on the subjective terms *expertise*, *quality guidelines*, *role*, *software development method* and *template*. To ensure a uniform interpretation and response, each question was accompanied by standard answers. Because respondents first had to read these, all free-form 'other' answers are expressed in a similar form to the examples. In the case of *quality guidelines*, an additional link to a webpage explaining the INVEST framework was included for additional context.

Furthermore, the focus of this study was the *card* aspect of user stories. We purposefully put less emphasis on the *conversation* and *confirmation* as explained by Ron Jeffries [13], which we will study in greater detail in the future.

6 Related Literature: User Stories, and Perception and Experiments in RE

Between 2003 and 2013, the adoption of user stories has grown tremendously [14]. In agile software development user stories are the predominant method to capture requirements [29]. Despite their popularity, research efforts concerning user stories are limited. Recent work has revisited user stories from a conceptual perspective. Wautelet et al. propose a unified model for user stories with associated semantics based on a review of 85 user story templates and accompanying example stories [30]. Gomez and colleagues propose a method for identifying dependencies between User Stories [10]. In an earlier paper, we presented a conceptual model that characterizes the structure of a valid user story and decomposes its parts linguistically. This conceptual model is the foundation upon which we built the Quality User Story Framework that proposes quality criteria that a user story should adhere to [21].

Liskin et al. investigate the expected implementation duration of user story as a characteristic of granularity. They find that in practitioners' experience combining the effort estimation of two small, clear-cut user stories produces more accurate results than when estimating a single, larger, more opaque user story [18]. Multiple authors have linked user stories with goals. Lin et al. [17] propose a mixed top-down and bottom-up method where an initial top-down analysis of the high-level goals is complemented by a bottom-up approach that derives more refined goals by analyzing user stories. A similar attempt has been implemented in the US2StarTool [23], which derives skeletons of *i** goal models starting from user stories. The key difference is that these models represent user stories as social dependencies from the role of the user stories to the system actor.

The number of papers that examine how practitioners use and perceive requirements engineering methods and artifacts is limited. Rouibah and Al-Rafee conducted a similar study to ours, investigating the "awareness", "use" and "perceived value generated" of 19 RE techniques based on survey responses by 87 practitioners from Kuwait [27]. Their findings include that the most used requirements elicitation techniques are interviews and surveys, but that the highest perceived value comes from decision trees, goal-oriented elicitation and prototyping. Other studies that study perception and use in the context of RE have a different focus. Hofmann and Lehner report on the self-perceived quality of RE service and RE products within RE teams without distinguishing between RE methods [12]. Abrahão et al. present a method to evaluate requirements modeling methods by gauging end-user perceptions, an adaptation of the Method Evaluation Model, and apply it to a Rational Unified Process extension that provides specific techniques for specifying functional requirements [1].

Nevertheless, the effectiveness of an RE method or technique is a frequent subject of academic literature. In fact, up to four different systematic reviews are available for some subdomains of RE. For example, Dieste and Juristo conducted a systematic review on the effectiveness of requirements elicitation techniques and found sufficient evidence to formulate five usage guidelines [9]. One example: unstructured interviews output more complete information than introspective techniques such as protocol analysis. Condori-Ferandez et al. did a systematic mapping study on empirical evaluation studies of software requirements specification techniques and found that most papers report on experiments that took place in academic environments [7]. The number of experiments conducted with actual practitioners is low. For example, Cruz-Lemus et al. conducted an experiment with practitioners to assess how composite states impact the understandability of UML statecharts [8]. They find the results are slightly more outspoken with a population of practitioners than a population of students. Penzenstadler, Eckhardt and Fernández even conducted two replication studies to validate their earlier evaluation of an artifact-based RE approach and tool [24]. These studies confirm that their simpler artifact model improves the quality of the created artifacts and ease of use.

7 Discussion and Conclusion

This paper has explored how practitioners that already employ user stories use and perceive them. Both the data from our survey with 182 valid responses and comments by follow-up interviewees indicate that software professionals are predominantly positive about *user stories* as well as the associated constructs *templates* and *quality guidelines*. Very few practitioners are downright negative about user stories. Our key findings on user stories are that:

F_1 Most of the user story adopters (94 %) use them in combination with Scrum.

F_2 The most prevalent user story template is the 'original' one proposed by Connextra.

F_3 Self-defined quality guidelines are unstructured and not using any quality guidelines is not a conscious decision.

F_4 The simple structure of user stories enables developing the *right* software, for it facilitates creating a common understanding concerning the requirement.

F$_5$ Stakeholders enjoy working with user stories, as they foster a pleasant workplace.

F$_6$ Using *a template* benefits RE, not *the template* that the team chooses.

F$_7$ Specifying the *why* part of a user story is essential for requirements quality.

F$_8$ Practitioners who use the INVEST quality guidelines are significantly more positive about the impact of user stories on productivity and the impact of templates on work deliverable quality.

F$_9$ INVEST is not a checklist, but a work guideline each team member should adopt.

F$_{10}$ INVEST is particularly useful for inexperienced teams. The necessity of INVEST diminishes for experienced teams.

F$_{11}$ Technical stakeholders are less positive about the effectiveness of templates and quality guidelines than non-technical stakeholders.

F$_{12}$ Practitioners with more expertise with user stories perceive them more positively.

We discuss **F$_4$**, **F$_7$**, and **F$_8$** in more detail. Throughout the interviews, respondents repeatedly mention that user stories help them create the right software. By requiring all stakeholders to think and talk about the details of a requirement, user stories build a common understanding of what the end-user expects of the software within a team. This identification of the right requirements enables development of the right software. This prevents expensive rework, improving productivity and work deliverable quality. Based on this finding, we hypothesize that using user stories reduces software development costs. An associated finding is the importance of the *why* part of a user story to deliver a common understanding and to support development of the right software. This confirms the fundamental theories in RE on the importance of the 'why' for software (process) analysis [16,25,31].

There also appears to be a correlation between relying on quality guidelines and the perception of user stories. Respondents that use INVEST are particularly positive in comparison to those that do not apply quality guidelines at all. A clear indication that having a structured list of characteristics of a good user story is beneficial. Recall, however, that our interviewees' self-defined quality guidelines are unstructured, informal approaches and that they are unaware of structured approaches like INVEST. Because of this, we call for an increase in the diffusion of knowledge concerning quality guidelines in order to further improve the positive perception of user stories.

This evaluation of practitioner's use and perception of user stories opens avenues for future research. To test whether adopting user stories reduces software development costs, we are planning to conduct a series of experiments. To improve the diffusion of structured quality guidelines like INVEST or the QUS Framework [21] we need to conduct a more thorough evaluation of their impact on software development. In particular, studies that take into account the opinion of practitioners that chose not to employ user stories or stopped employing user stories would fill a gap created by this work. Furthermore, despite user stories' increasing popularity, little to no advanced methods and tools originating from academia support them. As adoption of user stories increases, the importance of and opportunities for designing advanced methods and tools for user stories intensifies. We call for academia to focus more resources on user stories and its related concepts.

Acknowledgements. The authors would like to thank all survey respondents for participating in our research, the three respondents to the pilot survey as well as Leo Pruijt and Erik Jagroep for reviewing drafts of this paper.

References

1. Abrahão, S., Insfran, E., Carsí, J.A., Genero, M.: Evaluating requirements modeling methods based on user perceptions: a family of experiments. Inf. Sci. **181**(16), 3356–3378 (2011)
2. Bedell, K.: Opinions on Opinionated Software. Linux J. **2006**(147), 1, July 2006. http://dl.acm.org/citation.cfm?id=1145562.1145563
3. Boehm, B.W.: Understanding and controlling software costs. J. Parametrics **8**(1), 32–68 (1988)
4. Boone, H.N., Boone, D.A.: Analyzing likert data. J. Extension **50**(2), 1–5 (2012)
5. Clason, D.L., Dormody, T.J.: Analyzing data measured by individual likert-type items. J. Agric. Educ. **35**(4), 31–35 (1994)
6. Cohn, M.: User stories applied: for agile software development. Addison Wesley, Redwood City (2004)
7. Condori-Fernandez, N., Daneva, M., Sikkel, K., Wieringa, R., Dieste, O., Pastor, O.: A systematic mapping study on empirical evaluation of software requirements specifications techniques. In: Proceedings of the ESEM, pp. 502–505. IEEE Computer Society (2009)
8. Cruz-Lemus, J.A., Genero, M., Morasca, S., Piattini, M.: Using practitioners for assessing the understandability of UML statechart diagrams with composite states. In: Hainaut, J.-L., et al. (eds.) ER Workshops 2007. LNCS, vol. 4802, pp. 213–222. Springer, Heidelberg (2007)
9. Dieste, O., Juristo, N.: Systematic review and aggregation of empirical studies on elicitation techniques. IEEE Trans. Softw. Eng. **37**(2), 283–304 (2011)
10. Gomez, A., Rueda, G., Alarcón, P.P.: A systematic and lightweight method to identify dependencies between user stories. In: Sillitti, A., Martin, A., Wang, X., Whitworth, E. (eds.) XP 2010. LNBIP, vol. 48, pp. 190–195. Springer, Heidelberg (2010)
11. Hoda, R., Kruchten, P., Noble, J., Marshall, S.: Agility in context. In: Proceedings of OOPSLA, pp. 74–88. ACM (2010)
12. Hofmann, H., Lehner, F.: Requirements engineering as a success factor in software projects. IEEE Softw. **18**(4), 58–66 (2001)
13. Jeffries, R.: Essential XP: Card, Conversation, and Confirmation, August 2001
14. Kassab, M.: The changing landscape of requirements engineering practices over the past decade. In: Proceedings of EmpiRE, pp. 1–8. IEEE (2015)
15. Kruger, J., Dunning, D.: Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. J. Pers. Soc. Psychol. **77**(6), 1121–1134 (1999)
16. Lee, J., Lai, K.Y.: What's in design rationale? Hum. Comput. Interact. **6**(3), 251–280 (1991)

17. Lin, J., Yu, H., Shen, Z., Miao, C.: Using goal net to model user stories in agile software development. In: Proceedings of the SNPD, pp. 1–6. IEEE (2014)
18. Liskin, O., Pham, R., Kiesling, S., Schneider, K.: Why we need a granularity concept for user stories. In: Cantone, G., Marchesi, M. (eds.) XP 2014. LNBIP, vol. 179, pp. 110–125. Springer, Heidelberg (2014)
19. Lombriser, P., Dalpiaz, F., Lucassen, G., Brinkkemper, S.: Gamified requirements engineering: model and experimentation. In: Proceedings of the REFSQ (2016)
20. Lucassen, G.: Materials of survey and interviews on user story practice (2015). http://www.staff.science.uu.nl/~lucas001/user_story_materials.zip
21. Lucassen, G., Dalpiaz, F., van der Werf, J.M., Brinkkemper, S.: Forging high-quality user stories: towards a discipline for agile requirements. In: Proceedings of the RE, pp. 126–135. IEEE (2015)
22. McConnell, S.: An ounce of prevention. IEEE Softw. **18**(3), 5–7 (2001)
23. Mesquita, R., Jaqueira, A., Agra, C., Lucena, M., Alencar, F.: US2StarTool: generating i* models from user stories. In: Proceedings of the iStar (2015)
24. Penzenstadler, B., Eckhardt, J., Mendez Fernandez, D.: Two replication studies for evaluating artefact models in re: results and lessons learnt. In: Proceedings of the RESER, pp. 66–75 (2013)
25. Potts, C., Bruns, G.: Recording the reasons for design decisions. In: Proceedings of the ICSE, pp. 418–427. IEEE Computer Society (1988)
26. Ramesh, B., Cao, L., Baskerville, R.: Agile requirements engineering practices and challenges: an empirical study. Inf. Syst. J. **20**(5), 449–480 (2010)
27. Rouibah, K., Al-Rafee, S.: Requirement engineering elicitation methods: a kuwaiti empirical study about familiarity, usage and perceived value. Inf. Manage. Comput. Secur. **17**(3), 192–217 (2009)
28. Wake, B.: INVEST in Good Stories, and SMART Tasks (2003). http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/. Accessed, 18 February 2015
29. Wang, X., Zhao, L., Wang, Y., Sun, J.: The role of requirements engineering practices in agile development: an empirical study. In: Zowghi, D., Jin, Z. (eds.) APRES 2014. CCIS, vol. 432, pp. 195–209. Springer, Heidelberg (2014)
30. Wautelet, Y., Heng, S., Kolp, M., Mirbel, I.: Unifying and extending user story models. In: Jarke, M., Mylopoulos, J., Quix, C., Rolland, C., Manolopoulos, Y., Mouratidis, H., Horkoff, J. (eds.) CAiSE 2014. LNCS, vol. 8484, pp. 211–225. Springer, Heidelberg (2014)
31. Yu, E.S.K., Mylopoulos, J.: Understanding "Why" in software process modelling, analysis, and design. In: Proceedings of the ICSE, pp. 159–168. IEEE (1994)
32. Zave, P., Jackson, M.: Four dark corners of requirements engineering. ACM Trans. Softw. Eng. Methodol. **6**(1), 1–30 (1997)