

## An Empirical Study of Writing Effective User Stories

Vandana Gaikwad<sup>1\*</sup> and Prasanna Joeg<sup>2</sup>

<sup>1,2</sup>*Computer Engineering Department, Bharati Vidyapeeth University College of Engineering, Pune, 411043, India*

<sup>1</sup>*vandanagaikwad3@gmail.com, <sup>2</sup>joegpr@gmail.com*

### Abstract

*Software requirements should be accurately captured in the form of user stories so that development teams are able to meet customer expectations for the software. This study investigates the various challenges in capturing requirements of the agile scrum methodology and proposes some improvements. A user story writing workshop was conducted based on a standard set of business requirements. The resulting user stories were analyzed for correctness and problem areas were identified. Then a separate industry survey was conducted to confirm that the problem areas found in this workshop are indeed the common problem areas for industry practitioners. A set of improvements were then identified to address the problem areas. A second workshop was then conducted to check whether the improvements actually result in better user story writing. The results of the second workshop suggest that the improvements were quite effective in resolving most of the problem areas. They can thus be adopted by industry practitioners as best practices.*

**Keywords:** Agile, Scrum, User stories, Persona, Requirements Engineering

### 1. Introduction

About a decade ago, agile software development was introduced in industry, and since then it has gained widespread interest [1]. A recent study shows that a total of 63 countries are actively doing agile research [2]. Agile practices have evolved and been adopted worldwide. As per Version One sponsored survey of a large sample of global software development community in Oct 2014, more than 90% of respondents' organizations had adopted agile in their software organizations [3].

Additionally, Scrum is the most widely adopted flavor of agile, it makes as much as 86% of the various methods of agile [4]. This is an iterative, incremental and adaptable agile software development methodology [5], and has been extensively studied for adoption in globally distributed environments [6].

Requirements gathering is the most critical phase in software development, and its elicitation and understanding continues to be a very common problem even in an agile software development setting. In particular, requirements variability is a major challenge for all commercial software projects [7].

Furthermore, the scope of this study is to research challenges in the area of requirements engineering through an empirical method of conducting user story writing workshops. The user stories were reviewed as part of post-workshop analysis and issues were discovered with user stories that did not accurately represent end user requirements. The investigation of these issues led us to define a set of improvements, which were then validated in the second workshop. From the analysis of the second workshop's user stories, we found that the quality of user stories had improved. The proposed

---

\* Corresponding Author

improvements can thus be adopted by agile teams as best practices for creating user stories.

The remaining paper is structured as follows: Section 2 presents the methodology of how the research work was conducted. Sections 3 and 7 provide details of the experiments for workshop I and workshop II. Section 4 analyzes user stories developed in workshop I to identify shortcomings in how requirements were captured and subsequently recommend few improvements. Section 5 discusses findings from the industry survey on common problem areas and their correlation with findings from workshop I. Section 6 provides details on the proposed improvements that can be adopted as best practices. Section 8 describes analysis and results from both the workshops. Section 9 provides details on the impact of recommended improvements in increasing accuracy of user stories in workshop II. The conclusion is provided in Section 10.

## 2. Methodology

The user story writing workshops were conducted with a group of computer engineers in their second year of a Master's program, who had volunteered for the workshops. Several such studies have been conducted with students' participation [8] [9] [10]. In our first workshop, students were given standard training similar to what is conducted in the industry on the topic of agile principles, scrum framework, user stories and writing user stories. Such types of agile experiments have also been proposed and used in several other studies [11].

The participating students were divided into five groups and were given two problem statements each in the form of epics. The participants were asked to derive user stories for the epics and were required to write acceptance criteria and define a priority.

The authors played the role of product owners and were responsible for judging the correctness of the stories. We took help from two industry practitioners, who both have 4-6 years of experience using Agile, and who currently play the role of product owner and scrum master. They helped in reviewing our post-workshop analysis and their review comments were promptly incorporated.

The responses received from the participants were analyzed and the user stories were separated into accurate and inaccurate user stories. The inaccurate user stories were analyzed further and categorized into different problem areas. The problem areas identified have been reported in previous studies also [12] [13] [14].

In addition, a survey was conducted of 18 industry practitioners who are practicing agile for more than 5 years in various roles like product owners, scrum masters, project managers, user experience designers and software developers. The goal was to validate that the problem areas we found in workshop I are commonly occurring in the software industry where user stories are written. We also wanted to identify common hurdles that project teams encounter in developing software using the Scrum framework, with more focus on requirements engineering.

Through the analysis of common problem areas and study of industry best practices, we came up with improvement guidelines.

In the second workshop, we updated our training to incorporate improvement guidelines with supporting examples. The grouping of student participants created in workshop I was kept unchanged, but were given different epics and asked to write persona and user stories. The responses were again analyzed. The improvements we recommended were evaluated for their effectiveness in reducing inaccurate user stories.

The uniqueness of this methodology is that the improvement guidelines were cross-checked by industry practitioners whom we were consulting. This methodology can be used in other aspects of software engineering to identify gaps, define and validate improvements.

### 3. Experiment I: User Story Writing Workshop I

The user story writing workshop was conducted with a group of Computer Engineering Masters students from Bharati Vidyapeeth University College of Engineering in Pune, India. Five groups of four students were given a total of ten epics (two epics for each group).

The epics were selectively chosen from different software domains that represented requirements for developing software products and enterprise solutions. Few of the samples were as follows - shopping for electronic goods through an e-commerce website, an online voter registration application, a banking software for managing saving account online, a content management system, a pizza ordering and delivery mobile application. We asked them to write a set of user stories along with acceptance criteria, as well as define priority for each of the user stories. They were imparted training about agile principles, the Scrum framework, what user stories are and how to write user stories. In addition, a demonstration was conducted where few user story samples were created for an epic using the INVEST criteria [15] [16]. Handouts were given to the students for reference containing the summary of main points from the workshop. This was to help them recall the main points and also promote brainstorming within the group when creating user stories.

This was a one day workshop for 7 hours. First 3 hours of the workshop was dedicated for training and the remaining 4 hours was given for the students to form groups and write user stories for the different epics given to each group. The product owners explained the epics to the groups, also answered questions related to specific features as well as assumptions to be made. The students participated wholeheartedly throughout the workshop, each group member actively participated in writing user stories.

#### 3.1. Material and Experimental Tasks

For the workshop, as per the standard industry practice, the user stories were written on cards. The epic definitions were written on pink colored index cards and the user stories were written on yellow colored index cards. As shown in Figure 1, the participants were asked to write user story definition along with priority on one side of the card and acceptance criteria on the reverse side.

Front side of the card	Reverse side of the card
Title of User Story: User Story: As a <type of user> I want <perform a task> so that <unobvious goal> Priority:	Acceptance criteria:

Figure 1. User Story Card

### 4. Analysis: User Story Writing Workshop I

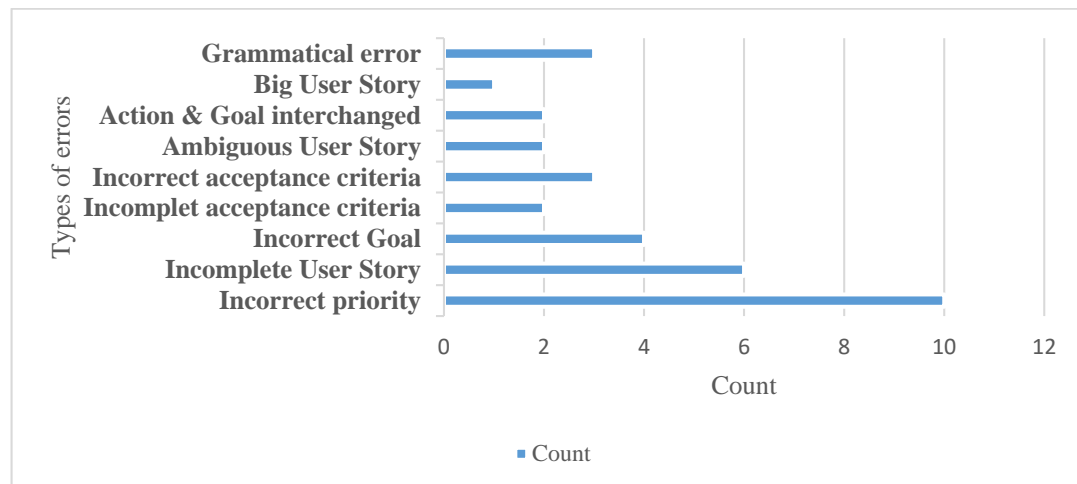
In the role of product owner, all the user stories created by the participants have been analyzed. Statistics from the user story writing workshop I are as follows:

**Table 1. Analysis of Workshop I**

Workshop parameters	Count
Total number of epics	10
Total number of user stories defined	59
Total number of user stories well defined	30
Total number of user stories with issues	29
Total number of errors found in user stories	33

As mentioned in Table 1, thirty user stories were found to be well defined and with correct priority definition. The story definition had independent user story goals, unambiguous action and correct role identified. The acceptance criteria for the user stories had been covered in sufficient details. Lastly, the grammar and structure of the user stories was found to be correct. These observations were cross-checked by the industry experts assisting the product owners.

For the remaining user stories, our analysis found thirty three errors. The error distribution is shown in Figure 2.



**Figure 2. Error Distribution for Incorrect user Stories in Workshop I**

As observed in the above figure, the highest number of errors were incorrect priorities defined across user stories. The second highest errors were incomplete user stories, where acceptance criteria or action/goal was missing. Remaining errors were due to ambiguous user stories, incorrect goals and user stories with very large scope. There were also few stories with grammatical errors.

## 5. Correlation of Problem Areas between Survey Responses and Workshop I

We did an independent industry survey of agile practitioners to understand if the problem areas analyzed from workshop I were also occurring in the industry.

The survey questionnaire was structured in two sections. Section 1 profiled the Org size, the role played in scrum team, team's scrum maturity level. Section 2 focused on the practice of requirements gathering and writing user stories; also challenges / best practices around them. After analyzing the survey responses, it was observed that there was a correlation between errors found by practitioners when writing user stories, and the problems found in user stories in workshop 1.

The findings from the industry survey for which we found similarities in problem areas are listed in Table 2 below. The first column lists the question asked in the survey, the second column provides the analysis from responses, while the related errors from workshop I are listed in the last column.

**Table 2. Correlation of Problem Areas Found from Survey Responses and Workshop I**

Questions asked in the Survey	Responses received from the survey	Analysis of workshop I with similar problem areas
What are the common errors found in user stories?	Most common errors were ambiguous user stories, incorrect priorities, related user stories.	Errors found in user stories were incomplete user stories, incorrect goal, ambiguous user stories, missing goal, action and goal interchanged, incorrect priorities.
Do you follow INVEST criteria for writing user stories?	70% respondents said they did not use INVEST criteria but would have benefited from using it. Also, a template for writing user stories would help.	INVEST criteria was introduced but not followed rigoursly which led to errors like ambiguous user stories, big user story.
What are the common problems faced during requirements gathering?	75% respondents stated that 'incomplete requirements' was one of the common problems.	Incomplete requirements which manifested as incomplete user stories.
Are non-functional requirements (NFR) captured as independent user stories?	70% respondents indicated that NFRs were not being captured in user stories.	Non-functional requirements had been left out in user stories.
For a tool to help with writing user stories, which features will be desirable?	Tool having user role modelling and persona support will be desirable.	Many user stories had missed out important user behaviors, so a tool having user role modelling would be needed

As observed, the issues which were discovered from the analysis of the user story writing workshop were similar to the problems faced by industry practitioners in the process of deriving user stories during requirements gathering.

## 6. Improvements Proposed for Writing User Stories

Analysis report of user stories for the workshop I indicated that around 55% of user stories were accurately written with correct definition, acceptance criteria and priority. Analysis was done on the remaining user stories and the issues were categorized across a few specific areas. We also analyzed the common issues from our industry survey and found a correlation in the problem areas.

Problem areas that were found in workshop I were similar in nature to those found in the industry. To overcome these issues and to achieve greater validity of results, we propose seven improvements in the procedure for writing user stories. These are described in subsequent sections.

### 6.1.Improvement 1: User Role Modeling and Persona Support

Personas are used to model user's behavior for using an application. They are derived from initial market research by interviewing end users and observing how users interact with the system. The idea is to expand and validate profiles by identifying goals and motivations. The need of user role modeling and persona support has also been recognized [17]. To simplify this process, we have proposed a new template based on Roman's persona template [18]. The Roman's persona template only provides guidelines about what each section should contain. The persona template which we created directly captures most important information about the user: information like persona's name, persona's details including characteristics and computer skills. In addition, it captures end goals in the form of benefits to be achieved or the problems to be solved, and the reason why the persona would want to use or purchase the product. Using the template, the user would only need to fill up required sections for the persona to make it ready. Advantages of using this new persona template are:

- It reduces possibility of introducing vague information
- Ensures that all information with regards to user's behavior gets captured
- The template is generic enough to capture persona information for all types of users (e.g., business users, administrators managing the application)

The persona template is as depicted in Figure 3 below

The figure shows a persona template form with the following sections:

- Epic**: A text input field.
- Persona**: A section header.
- Name of Persona**: A text input field.
- Profile**: A section header.
- Profile Content**: A large text area containing the following template text:  
[Name of Person] Works as [ role/post]/for [ Name of Organization/Type of business]  
He/ She is [ age ] years old,  
He/ She has completed his [Degree/Qualification/Education] from [University/place] in [Specialization/Field].  
He / She has got more than [ ] years of experience in [Area/Work details]  
He/ She is [type of temperament] Person,  
His/ Her characteristics are [ ]  
His/ Her computer skills are at [User /Power User /Administrator] level.
- End Goal**: A section header.
- End Goal Content**: A text area containing the following template text:  
He/ She wants to buy/use [Product ] for [ specific goal ]  
Product is very useful for him/her because [ reasons ]

Figure 3. Persona Template

## 6.2. Improvement 2: Template for Writing User Story

We have created a web form template for writing user stories so that the participants ensure completeness and uniformity. To write user stories on physical cards or on post-it notes is a commonly followed practice, but it has an additional overhead of physically handling the cards. Hence if a web form is used, it will be very convenient to manage. The additional benefit is that the web form can be accessed using different devices like smartphones, laptops, and tablets allowing different stakeholders at distributed locations (even the end customer) to view the user stories. One more change that has been incorporated is in writing acceptance criteria. The structure the acceptance criteria into three sections: namely positive scenarios, negative scenarios and non-functional requirements. This is to make sure that the user stories are able to satisfy all the important aspects when they are delivered to the end user in product releases. The user story template is depicted in Figure 4.

The form is a vertical rectangle with a light gray border. It contains the following elements from top to bottom:

- User Story ID**: A label in red text followed by a horizontal line for text entry.
- Title of User Story**: A label in red text followed by a rectangular input box.
- User Story**: A label in red text followed by a large rectangular input box.
- Priority**: A label in red text followed by a horizontal line for text entry.
- Acceptance Criteria**: A label in blue text followed by three stacked rectangular input boxes:
  - The first box is labeled **Positive Scenario** in blue text.
  - The second box is labeled **Negative Scenario** in blue text.
  - The third box is labeled **Non-functional Requirement** in blue text.

**Figure 4. Template for Creating a User Story**

## 6.3. Improvement 3: Accurately Defining User Stories

The following issues were observed with user stories defined by participants in workshop I

- ‘Action’ and goal interchanged [12]
- ‘So that’ criterion or end goal is missing [13]
- Ambiguous user story definition [19]
- Big user stories [12]

To avoid above problems and define user stories accurately, following improvements were suggested:

- Need to understand the difference between action/ task and goal/benefit/value. A user story should follow a proper sequence, which means it should always start with action first and end with a goal. The goal should be connected by 'so that' criterion.
- Need to write simple and small user stories, avoid unnecessary details and ambiguous terms. It will be of help to refer the INVEST criteria after user stories have been written.

#### **6.4. Improvement 4: Assigning Correct Priority to User Story**

The user story definition must be enough to decide priority for the user story. Priority indicates the importance of story to the business, so the business value is the key factor in determining priority. The product owner is primarily responsible for defining priority. Following guidelines can be used to assign priority to a user story.

For assigning priority, we recommend following factors that a product owner should consider:

- **Value to the Business**

This is the most important factor which product owners must consider when assigning priority. User stories that are critical to the success of the business, as well as those that represent primary functionality for the end user to use the system, will qualify to be at a higher priority.

- **Associated Risk**

It is very important to prioritize user stories collaboratively with the team because the technical risk associated with user stories involving complex development or testing can be identified upfront.

- **Size of User Story**

When user stories have similar business values and stand independent, then user stories that deliver a small unit of functionality compared to their peers who deliver larger and complex functionality will rank higher in priority. Smaller user stories have a better chance of being completed so they reduce the risk of failing in the sprint. Also, they improve the flow in the sprint and are accurately estimated.

In workshop I, the priority of user stories have been quantified using the following classifications - Must have, Should have, Good to have. This is MoScow prioritization method that has been mentioned by DSDM Consortium [20]. However, it was found that the criteria 'Must have' and 'Should have' were confusing for the participants. Hence, we revised this to use the following classification - Critical, High, Medium and Low and represented them on a numerical scale from 1 (Critical) to 4 (Low). Priority definition is detailed in Table 3, supported with examples for a banking customer's use of debit card at an ATM machine. Following examples were used in our upgraded training conducted in workshop II.



**Table 3. Prioritization Scheme**

Priority Scale	Priority Level	Definition
1	Critical	This user story has the highest priority and represents functionality critical to business. It represents the features that must be included before the product can be launched and the failure of which will result in loss of business. User Story: As a customer, I should be able to withdraw money from my bank account so that I do not need to visit the bank.
2	High	This user story represents an important business feature which is most required in a product launch, the failure of which will result in impaired functionality for the end user and will be a show stopper. User Story: As a customer, I should be able to see updated balance on the ATM screen after a transaction so that I have information on current balance in the account.
3	Medium	This user story represents a feature which is required, but where the application may have alternate ways to achieve the end outcome. The user story can possibly be moved into next release. User Story: As a customer, I should be able to get the history of past transactions from the ATM machine so that I can verify them for accuracy.
4	Low	This priority indicates that user story is desirable, but not very critical. Functional or quality enhancing features are included here. These stories can be reprioritized from a software release without having a huge impact on the release. <b>User Story:</b> As a customer, I should be able to change PIN of the debit card so that I don't need to visit bank or call customer care

There are different techniques available for prioritization, but no matter the prioritization technique, we need to ensure that user stories that give the most value to the end customer are accomplished first. In addition, the vendor's domain knowledge is a key asset for setting up a successful client-developer collaboration. We had emphasized in our training that if we don't use prioritization technique, it will risk delivering low priority user stories that are less desirable to the end user compared to the essential ones. Also, it is essential that the product owner reviews and if necessary revise the priority of user stories regularly.

### 6.5. Improvement 5: Writing Effective Acceptance Criteria

Acceptance criteria, which are an important part of every user story, are statements having a clear pass/fail result, that specify both functional and non-functional requirements [21]. Acceptance criteria should be complete and should not be very generic or too specific as they are the requirements that need to be met for a story to be considered complete. Following are the recommended improvements along with examples that were used in the training of workshop II. In the training, we recommended to use the following format for writing acceptance criteria -

**“Given some precondition when I do some action, then I expect some result”** [22].

When we write acceptance criteria in this format, it not only provides a consistent structure, but we are also helping testers to determine as to when to begin and end towards testing for that specific work item.

**Example 1: User Story:** As a mobile subscriber, I want to pay my mobile bill online so that I do not need to visit the store.

**Acceptance Criteria:** Given that my mobile bill is due and when I attempt to pay my bill on the online portal using my valid credit card, then the payment should be completed without errors or warnings.

Each time it may not be possible to write acceptance criteria in the given/when/then format.

There is no hard and fast rule that every acceptance criteria should fit in the above template of acceptance criteria. You can write acceptance criteria by creating a verification checklist [22].

- **Positive and Negative Scenarios in Acceptance Criteria**

In the training for workshop II, we recommended that acceptance criteria should have both positive and negative outcomes covered as in the example below.

**Example 2: User Story:** As a website user, I want to register as a paid user of the website so that I can access paid services on the website.

**Acceptance Criteria:**

- A user has to fill all mandatory field data in acceptance criteria, else it should throw an appropriate error message.
- Registration payment can be done using a credit card or debit card.

#### **6.6.Improvement 6: Coverage of NFR in User Stories or in Acceptance Criteria**

Non-functional requirements would cover qualities of the system that cut across user-facing features and are of the type like security, reliability, performance, scalability and availability. In training for workshop II, we recommended specifying non-functional requirements in user stories and acceptance criteria [23]. According to, Cao and Ramesh, customers often “focus on core functionality and ignore NFRs” [24].

A common challenge with writing user stories is as to how to handle a product's non-functional requirements (NFRs) [25]. It has been proposed that the team should discuss NFRs with the Product Owner, who in turn should provide guidance on what the customer would expect for that user story.

- **Non-functional Requirements in User Stories or in Acceptance Criteria**

Following are some examples that demonstrate non-functional requirements in user stories for different aspects.

**Example 1: User Story:** As a bank account user, I should be able to use my debit card at ATM of non-parent banks so that I can withdraw money, anywhere in the country.

**Acceptance Criteria:**

- ATM should check the validity of the debit card.
- Only fixed amount can be withdrawn.
- The non-parent bank should be an authorized nationalized bank.

**Example 2: User Story:** As a travel agent, I want my website should provide 24x7 service so that customer can access the latest information and get good service.

**Acceptance Criteria:**

- It should always show ‘NEW’ tag for the latest information.
- The customer should get acknowledged by his request within 5 seconds by email/SMS.

- Outdated information should not be displayed on the website.

### **6.7. Improvement 7: Typos and Grammatical errors in user Stories and Acceptance Criteria**

Descriptions of user stories, acceptance criteria *etc* are given in simple English and so are prone to misinterpretation by reading and defining user stories. Moreover, effective communication plays a key role in Scrum, so it is very important to avoid typos and grammatical errors.

We have recommended doing a proofreading exercise in workshop II within the Scrum team, once all the user stories were written. Also, every tool that captures user stories should include a spelling and grammar checker.

## **7. Experiment II: User Story Writing Workshop II**

The workshop was conducted with the same set of student participants. Before starting with the user story writing exercise, we covered the Scrum training again and it was enhanced by explaining the seven improvements proposed in Section 4, along with relevant examples. Templates for user story writing and persona were provided and it was made sure that only those templates were used during the user story writing workshop.

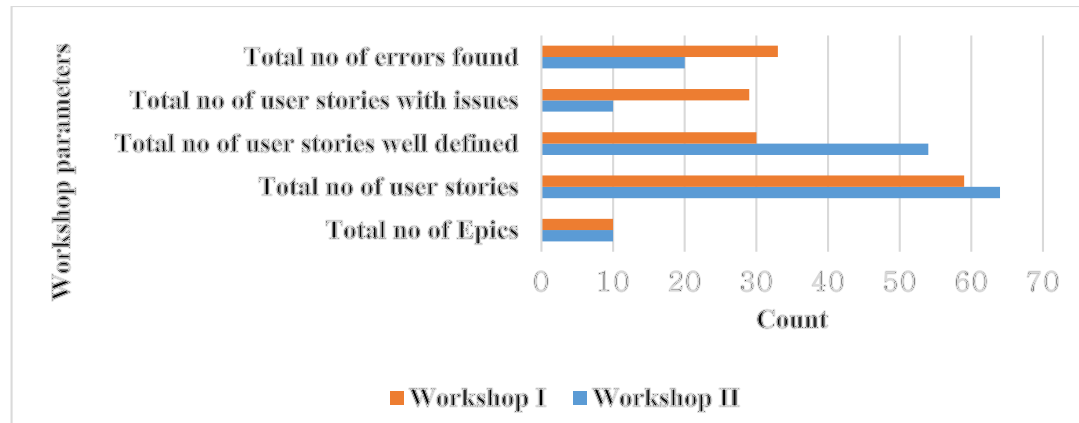
The same epics used in the workshop I were used for this workshop and the participants were asked to write a set of user stories along with acceptance criteria as well as define priority for each of the user stories. The epics were exchanged in such a way that the group that had already worked on one epic was given a different epic in order to ensure better validity of the results.

### **7.1. Material used for the Workshop II**

For the workshop, the product owner asked the participants to write persona using the new persona template explaining the benefits of the new template. The product owner also asked the participants to write user stories using the new user story template. In this style, the acceptance criteria had been divided into three sections, namely the positive scenario, negative scenario and non-functional requirements (NFRs). The Product Owner also provided clarifications on the epics to participants during the user story writing when workshop was in progress.

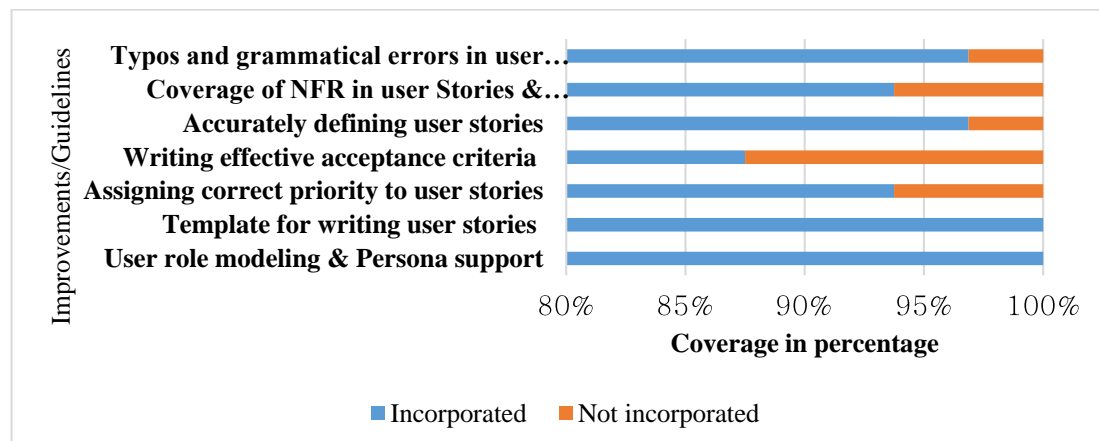
## **8. Results and Discussions**

We have analyzed all the user stories developed by the participants in workshop II. Figure 5 shows a high-level analysis of how the participants fared with writing user stories in workshop II as compared to workshop I.



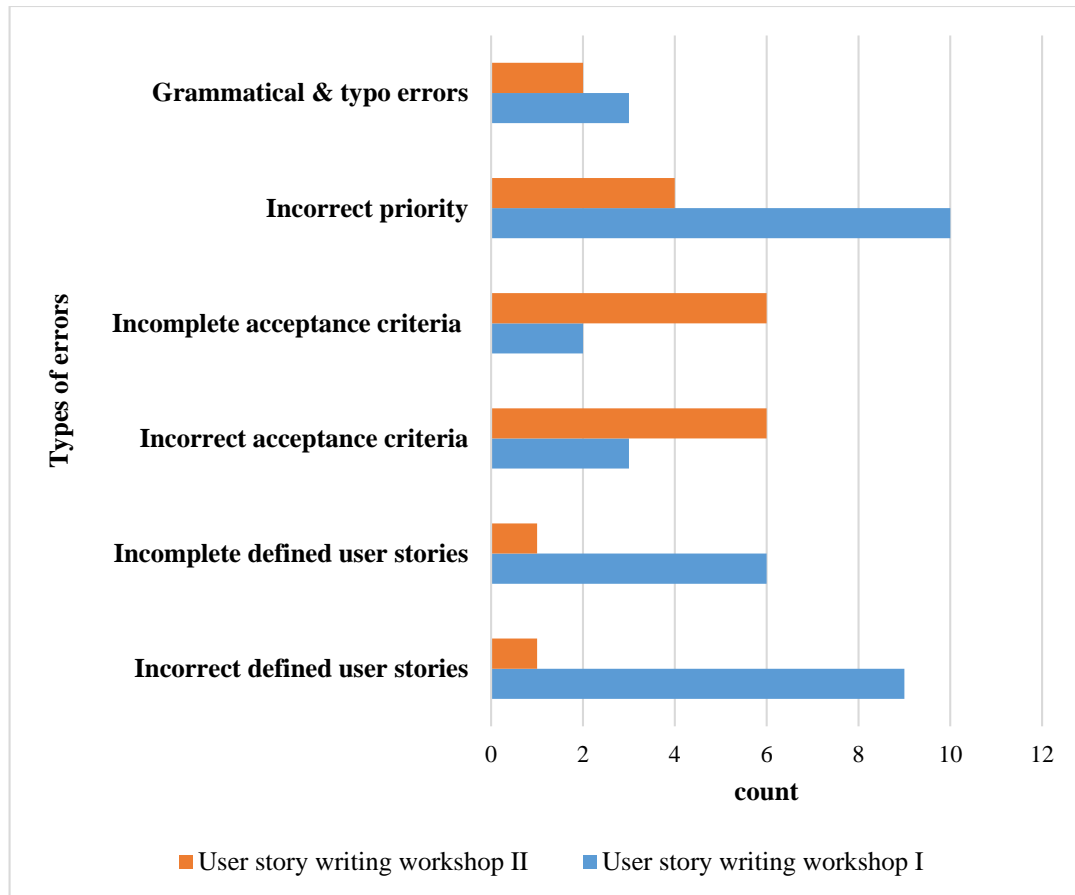
**Figure 5. Analysis of User Stories from Workshop I and Workshop II**

Notice that the user stories that had been correctly defined in workshop II increased to 54 from 30. Guidelines/improvements presented to the participants before the workshop were found very beneficial and resulted in less number of errors in user stories. In workshop II, 84% of the user stories were correctly written as compared to 55% in the previous workshop (as confirmed by the product owners and industry experts). Even persona and user role modeling support were found very useful by the participants. Using persona, they understood persona details which helped them write user stories for end users.



**Figure 6. Analysis of Improvements/Guidelines Incorporated in Workshop II**

Figure 6 shows the percentage of guidelines/improvements that were incorporated by participants when they wrote the user stories in workshop II. These were calculated by checking all of the user stories as to whether they incorporated the improvements/guidelines. Note that the participants may not have incorporated the improvements 100% because all the participants may not have understood the improvements fully.



**Figure 7. Analysis of Errors Occurred in Workshop I and Workshop II**

For the inaccurate user stories that had been identified in workshop I and II, Figure 7 shows the distribution of errors across various error types. Note that there is an increase in errors for type incomplete and incorrect acceptance criteria in workshop II because we expected acceptance criteria to be clearly articulated in sections - positive scenario, negative scenario and non-functional requirements. In workshop I, we did not have this format.

### **8.1. Improvements in Workshop II from Learning in Workshop I are as follows –**

**8.1.1. Across User Stories:** In all respects, we observed that the user stories written in the workshop II were more accurate and standardized after making modifications to the user story template. Also all user stories had been defined correctly with accurate and unambiguous action and goals. Better results were achieved by defining user stories with the use of persona technique.

**8.1.2. Across Priority of User Stories:** From the feedback received from participants after workshop I, we observed that the criteria ‘Must have’ and ‘Should have’ were confusing for the participants. Hence, the criteria were revised to use the following classification - Critical, High, Medium and Low and represented on a numerical scale from 1 (Critical) to 4 (Low). In workshop II, all participants understood prioritization classification well and accordingly assigned priority to user stories. Sample persona written in the workshop is depicted in Figure 8.

<b>Persona</b>
<b>Name of Person:</b> Samantha
<b>Profile</b>
Samantha works as business manager for a private company localized to a single city. She is 38 years old. She has completed her MBA from university in managing External Affairs and Business Relations She has more than 10 years' experience in her business. She is an exact person. Her characteristics are being time bound, punctual and informative. Her computer skills are at power user level.
<b>End Goal</b>
She wants to use the cab service app for catching flight and booking cabs to be available at airports. Product is useful for her because she doesn't always have personal vehicles available at every travel station among many cities.

**Figure 8. Sample Persona Written in Workshop II**

**8.1.3. Across Acceptance Criteria:** In workshop II, while writing acceptance criteria for user stories participants have considered all scenarios like positive, negative and non-functional requirements. Also, the acceptance criteria written were found to cover all aspects of acceptance criteria specific to the user story. In workshop II, participants were asked to use the given/when/then template wherever necessary, or alternatively use a verification checklist.

#### **8.1.4. User Stories which were Found to be Correctly Defined in Workshop II**

**Example 1: User Story ID:** Bank\_04, **Title of User Story:** Account Summary

**User Story:** As an online banking customer, I want to transfer money to another account, so that I can make successful transactions. **Priority:** 1

##### **Acceptance Criteria**

**Positive Scenario:** Given that the destination account number and bank name is valid when a user requests to transfer money, then money should get transferred to the destination account.

**Negative Scenario:** If the transfer money is more than the amount available in the account, then the transaction is discarded and an error message should display about insufficient balance.

##### **Non-functional Requirements:**

- The transaction should be secure and encrypted.
- The time span for transaction compilation is less than 15 seconds.

**Example 2: User Story ID:** adv\_04, **Title of User Story:** User registration

**User Story:** As a business owner, I want to register myself on the online advertising service portal so that I can post my products. **Priority:** 1

### Acceptance Criteria

**Positive Scenario:** Given that the online portal opens the registration window when the user enters details, then the user should get registered successfully.

**Negative Scenario:** Given that the online portal opens the registration window when a user misses filling details, then the user should get an error message on the screen.

**Non-functional requirement:** The registration should not take more than a minute to create the user account.

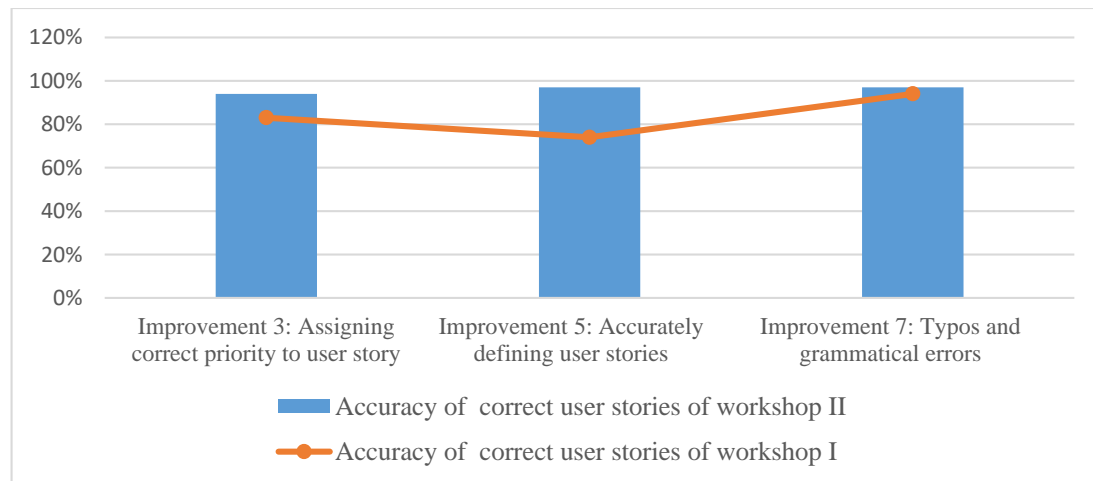
## 9. Validity of Results

In table below, we calculated the increase in accuracy of user stories across three improvements. For the remaining four improvements, accuracy data in workshop I was not available because the guidelines were not followed when the user stories were created. Few examples were - Coverage of positive scenario, negative scenario and NFR in acceptance Criteria, user role modeling and persona support and template for writing user stories. In Table 4 below, the total count of user stories with errors (for which improvements were then defined) for Workshop I and Workshop II have been listed, as identified in the analysis phases. Based on the knowledge available the total number of user stories 59 in workshop I and 64 in workshop II, the accuracy of the correct user stories was calculated. Note that the total count of user stories in both workshops was different because the same group did not work on the same epics in both workshops.

**Table 4. Analysis of Different Types of Errors in User Stories**

Type of errors and analysis	Workshop	User stories without specified error type	User stories having specified error type	Accuracy of correct user stories (derived)
Analysis of errors in context of assigning correct priority to user story	Workshop I	49	10	83%
	Workshop II	60	4	94%
Analysis of errors in context of accurately defining user stories	Workshop I	44	15	74%
	Workshop II	62	2	97%
Analysis of errors in context of typos and grammatical errors	Workshop I	56	3	94%
	Workshop II	62	2	97%

The mapping of errors to improvements and improvements to errors has a 1:1 relationship. Figure 7 demonstrates the impact of the improvements in increasing the accuracy of user stories from workshop I to workshop II. As observed, improvement 5 that provides guidelines to defining user stories shows the largest increase in improving the accuracy of user stories.



**Figure 7. Impact of Recommended Improvements in Increasing Accuracy of Correct User Stories**

## 10. Conclusion

This paper focuses on investigating and analyzing different challenges around how requirements can be most effectively converted into good user stories. As an outcome of workshop I, we came up with the idea of improving the requirements analysis process. We conducted a survey of agile practitioners in the software industry to understand common problems faced while developing user stories. We observed that the nature of issues was similar, which then led us to validate the improvements through a second workshop. We also developed templates for persona definition, a web form for user story definition and a structure for writing acceptance criteria. As part of workshop II, we updated the training done in workshop I to incorporate the improvements/guidelines and conducted an enhanced version of the training with the templates. From the post-workshop analysis, we observed that the count of well-defined user stories had improved and the count of errors in user stories had reduced.

As part of future work, we intend to provide the guidelines to industry practitioners and validate that similar improvements in user story creation are also seen in the industry. Our long term work will involve studying the other aspects of Scrum methodology to see if there are additional opportunities for improvements. We also intend to create a user story management tool which will incorporate the improvements recommended in this study.

## Acknowledgments

The authors would like to acknowledge the efforts of all the students of M.Tech Computer Engineering, Bharati Vidyapeeth University College of Engineering, Pune, India for their participation in the workshops.

## References

- [1] K. Beck, "Manifesto for agile software development", (2001) Feb., <http://agilemanifesto.org/>.
- [2] T. Dingsøyr, S. Nerur, V. Balijepally and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development", *Journal of Systems and Software*, vol. 85, no. 6, (2012), pp. 1213-1221.
- [3] 9TH ANNUAL State of Agile™ Survey, VERSIONONE.COM, VersionOne, Inc, (2015).
- [4] D. Giudice, "The 2015 State of Agile Development Forrester Research", (2015) August.
- [5] K. Schwaber and M. Beedle, "Agile Software Development with Scrum, Prentice Hall", (2002).
- [6] M. Cristal, D. Wildt and R. Prikladnicki, "Usage of SCRUM practices within a global company", 3rd IEEE International Conference Global Software Engineering, (2008), pp. 222-226.



- [7] A. S illitti and G. Succi, "Requirements Engineering for Agile Methods", Engineering and Managing Software Requirement, (2005), pp. 309-326.
- [8] C. Von Wangenheim, R. Savi and A. Borgatto, "Scrumia: An educational game for teaching scrum in computing Course, Journal of Systems and Software, vol. 86, no. 10, (2013), pp. 2675–2687.
- [9] V. Mahnic and T. Hovelja, "On using planning poker for estimating user stories", Journal of Systems and Software, vol. 85, no. 9, (2012), pp. 2086-2095.
- [10] V. Devedzic, S. Milenkovic and V. Deved, "Teaching Agile Software Development: A Case Study", IEEE Transactions on Education, vol. 54, no. 2, (2011), pp. 273-278.
- [11] V. Mahnic and T. Hovelja, "Teaching User Stories within the Scope of a Software Engineering Capstone Course: Analysis of Students' Opinions", International Journal of Engineering Education, vol. 30, (2014), pp. 901–915.
- [12] M. Cohn, User Stories Applied: For Agile Software Development, Addison-Wesley, Boston, MA, (2004).
- [13] K. Kaczor, "5 Common mistakes we make writing User Stories", (2011), <https://www.scrumalliance.org/community/articles/2011/august/5-common-mistakes-we-make-writing-user-stories>.
- [14] R. Pichler, "5 Common User Story Mistakes", (2013), <http://www.romanpichler.com/blog/5-common-user-story-mistakes/>.
- [15] B. Wake, "INVEST in Good Stories, and SMART Tasks", (2003), <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>.
- [16] L. Buglione and A. Abran, "Improving the User Story Agile Technique Using the INVEST Criteria, 8th International Conference on Software Process and Product Measurement", IEEE Computer Society, (2013), pp. 49-53.
- [17] S. Dimitrijevic, J. Jovanovic and V. Devedzic, "A Comparative Study of Software Tools for User Story Management", Information and Software Technology vol. 57, (2015), pp. 352-368.
- [18] R. Pichler, "Roman's Persona Template", (2014), <http://www.romanpichler.com/tools/persona-template>.
- [19] D. M. Berry and E. Kamsties, "Ambiguity in Requirements Specification", in Perspectives on Software Requirements, Springer, vol. 753, (2004), pp. 7–44.
- [20] DSDM Consortium, DSDM public version 4.2, [www.dsdm.org](http://www.dsdm.org).
- [21] S. Povilaitis, Acceptance Criteria, 2014, <http://www.allaboutagile.com/acceptance-criteria/>.
- [22] Guides to Agile Practices, <http://guide.agilealliance.org/guide/gwt.html>
- [23] M. Cohn, Non-Functional Requirement as User Stories, (2008), <https://www.mountangoatsoftware.com/blog/non-functional-requirements-as-user-stories>.
- [24] L. Cao and B. Ramesh, "Agile requirements engineering practices: An empirical study", IEEE Software, vol. 25, no. 1, (2008), pp. 60-67.
- [25] A. Alashqar, A. Elfetouh and H. El-Bakry, "Requirement Engineering for Non-Functional Requirements, International Journal of Information and Communication Technology Research, vol 5, no. 2, (2015), pp.21-27.

## Authors



**Vandana Gaikwad**, she is an Associate Professor at Bharati Vidyapeeth University College of Engineering, Pune, India and a Ph.D. student in the same University. She has over 12 years of academic experience. Her area of research interest is Software engineering.



**Prasanna Joeg**, he received his B.Tech and M.Tech degrees from Indian Institute of Technology Bombay and his Ph.D. is in Computer Science from Indiana University, USA. He has over 25 years of academic and industry experience. His research interests are in Software engineering, Machine learning and big data algorithms.

