Human Stories - A New Written Technique in Agile Software Requirements

Nguyen Thien Khanh
Vincent Many School of Science and Technology
Assumption University of Thailand
Bangkok, Thailand
tknguyen.designer@gmail.com

Jirapun Daengdej
Vincent Many School of Science and Technology
Assumption University of Thailand
Bangkok, Thailand
jirapun@scitech.au.edu

Habibi Husain Arifin
Vincent Many School of Science and Technology
Assumption University of Thailand
Bangkok, Thailand
habibihusainarifin@gmail.com

ABSTRACT
Software requirement is one of the most important issues for starting any software projects. Especially in Agile software development, it is hard to be specific and complete, while customers/users keep asking about time and cost to finish their project. Fortunately, some researchers found that by improving the quality of requirements, the effort of estimation can be reduced. In this paper, we due with the quality in Agile software requirements focusing on their different written techniques. User-story and persona-story are two most famous written methods to deal with Agile requirement quality. While user-story focuses on the holistic view of the users that playing with role, persona-story focuses on the detailed view of the individuals that playing with people. However; throughout our research and other arguments, both of them have their strengths and weaknesses. Some researchers found that there is a coherent relationship between them, but the agreements are still the discussions. Therefore, we introduce human-story as a promising written technique to overcome the disadvantages of both existing stories, to enhance the key benefits of them and to fill the gap between differences. INVEST Grid is used to evaluate the quality of Agile requirements and to give the comparison among three techniques, the results show that persona-story and human-story have the higher scale values than user-story in most of the type of software requirements. Because the Grid cannot give us a clear comparative result to the significant different among three stories, the introduced Agile requirement checklist makes sure our proposals meet our goal. The results displays the percentage of the completeness in human-story is 84% while user-story and persona-story are 44% and 56%.

CCS Concepts
• Information systems → Enterprise information systems

Keywords
Agile software requirement; user-story; persona-story; role; persona; human-story; INVEST grid; agile requirement quality checklist.

ICSCA 2017, February 26-28, 2017, Bangkok, Thailand
 ISBN 978-1-4503-4857-7/17/02…$15.00
DOI: http://dx.doi.org/10.1145/3056662.3056680

1. INTRODUCTION
In Agile software development, the more defined requirement will lead the accuracy of the size and schedule of the project [1-3]. Also, if the requirement is well defined at the beginning, we can clarify what is important to the customers for prioritization [4]. However, requirement document is hard to be complete and specific at the early state and the customers always come up with the question "*how long*" and "*how much*" for any new projects [5]. Fortunately, the estimation effort can improve when focusing on the quality of the requirement such as been written in a structured way, but in the opposite way [6]. And, there is a direct relationship between requirement quality and the effort of sizing of requirement [7]. Moreover, if the quality of these requirements improves, it will affect the effort such as time and cost of estimation [8]. By focusing on the writing way of software requirement to improve its quality, user-story is mostly used in requirement documentation method [5]. On the other hand, another well-known method calls persona-story. While user-story plays with the roles, persona-story plays with people. Despite many different perspectives on requirements' quality exist [9] such as role and persona, there are also the arguments between them.

In this paper, we introduce human-story as the promising Agile requirement written technique to deal with the occurring issues with user-story and persona-story.

The paper will be organized as the overview of two existing methods in their history, definition, improvement and their key differences such as role and people (Section II.); the arguments in their disadvantages and comparisons in their benefits and their relationships (Section III. & IV); human-story based on proposed criteria and its evaluation methods in comparison to two other stories (Section V. & VI.); finally, our limitation, conclusion and future works (Section VII, VIII and IX).

2. LITERATURE REVIEWS
2.1 User-Story Overview
2.1.1 History
In 1996, user-story first used in extreme programming project, C3 at Chrysler. In 1999, it was appeared in the book eXtreme Programming Explained by Kent Beck. The idea of describing requirements as stories was much in *human-computer* interaction [10].

2.1.2 Definition & improvement
User-story as defined as a part of Agile approach that focuses on talking about requirements rather than writing. User-story

includes the desired functionality that was written in two or more sentences and the series of conversations [11]. Mike Cohn defined the functionality of user-stories that is valuable to user and purchaser of a system or software. User-story includes three aspects: a written description, conversations and tests [12]. User-Story was written on *cards*, the *conversation* between customer to programmers and *confirmation* showed to the customers by programmers after finished story at the end of each iteration [13].

For the form of the user-story, at the beginning, it did not have a specific form [10]; therefore, a team in the United Kingdom - Connextra, in 2001 proposed the first template of user-story: *As a <role>, I want<goal/desire> [so that <benefit>]*. The role is played in the first form of user-story. During a lot of improvements, Mike Cohn popularized the form of user-story: *As a <type of user>, I want <goal>, so that <some reason>* [12]. For example, "*As an Administrator, I want to receive an email when a contact form summited, so that I can respond to it*" [9]. Based on a key aspect in RE [14], user-story shows the essential elements of a requirement: *who* it is for, *what* it expects from the system and *why* it is important.

2.1.3 The role in user-story

For the role specification, Kristen Nygaard defined "*A role is defined through a specified task or a group of closely related tasks, which are performed by persons during the development and/or operation system*" [10]. In requirements engineering, role is the specific behavior of an entity participating in a particular concept [15]. In software development, Jacobson made use of roles in user cases. Role was captured in user case models in the *actor* [16] – Actors in user case describe the interaction between users and external systems [17]. The main function of role is to describe activities and responsibilities of an individual [10].

2.2 Persona-Story Overview

2.2.1 History

The term persona has been used for a long time. According to William Hudson, it is quite old and its origination from Latin called *person* [10]. Far from 1975, Paul Freiberger and Mike Swaine credit Cooper with writing the "*first serious business software from microcomputers*". Until 1983, Cooper personas became a more effective approach to make the software more user-friendly [18]. In 1999, persona was firstly applied in Alan Cooper book The Inmates Are Running the Asylum to characterize users of an interactive system during design [19]. Recently, using persona has been suggested by Grudin and Pruitt [20-21] in conjunction with scenarios [17].

2.2.2 Definition & improvement

In the context of product development, "*a persona is an archetype of a fictional user representing a specific group of typical users*", it includes names, likenesses, occupations, friends and other personal data [20]. In particular, a persona is an archetypical character [22] while its photographs are of models that have a likeness to an archetypical one [17]. This archetype represents the ideal user of your product. The purpose of persona is to understand the personalities and characteristics of the target audiences to solve their own problems [22]. The main goal of persona is to create empathy and to motivate the team [10, 23 and 24]. William Hudson argued the original purpose of personas. According to William, to be Agile, we need *minimal, collaborative personas* [10].

For the form of persona-story, William suggests the form as *<persona [: role]> <performs a task> [so that<unobvious goal>]*, while *persona: role* is based on the Unified Modeling Language notation *name: class* [10]. Another form is introduced by Fred Henry Williams called A.A.R.T (Actor, Action, Results and Test) while an actor is a real people [25].

2.2.3 The persona in persona-story

For the persona specification, according to Scott W. Ambler, different from actors that represent the roles, personas play with people [26]. Dananthi Arnott defined two types of personas in playing with people such as the *buyer persona* who invests the products and the *user persona* who uses the product [22]. "*Persona stories are written: about personas, not roles*" [10]. While taking all the users as a group is abstract [19], a persona represents a "*proxy*" of users and by calling them by name. Therefore, persona provides meaningful information such as talk and reason about the group through their characteristics of one fictional individual [17].

3. PROBLEM STATEMENTS

Throughout our research, we found that both user-story and persona-story include their own strengths and weaknesses. In particular, user-story is good for understanding user relationships (Table 1) while persona-story can increase the empathy basing on the real users (Table 2). On the other hand, persona-story leads to misunderstanding user relationship while user-story reduces empathy. Therefore, the advantages of one story can serve the disadvantages of other. Two tables below indicate the disadvantages and advantages of two stories.

Table 1. Role's Pros and Persona's Cons

Role-Understand the user relationships *Looking at holistic view*	**Persona- Misunderstand the user relationships** *By looking at individuals*
From developers: understanding the essentials of user relationship [27] such as job functions, permissions, responsibilities, skills and qualifications [21]; simple and faster to develop [27]	*From developers*: cannot write stories from single perspectives because of different backgrounds and different goals [12]; struggling for developers in the effort of identifying empathy [27]; take time (months, even years) to master [18] and limited resources [27]

Table 2. Role's Cons and Persona's Pros

Role - Decrease Empathy Based on *assumption*	**Persona-Increase Empathy** Based on *real people*
From developers: knowing nothing about *context of use* [10]; wrong people [10], misunderstood of users [22], not understanding the needs of users, lack of enthusiasm, working based on assumption, etc. [10]; confusing of developers because of different desires and experiences of users [22]; problems with making decisions [17]; wrong time - estimation becomes abstract because of assumptions [10]	*From developers*: reducing the bias with developers by the word – user [28]; build the system bases on specific users without assumption [29]; more understanding about the needs and behaviors of users [25]; working with the real users with the real problems [19]; more understanding what tasks that users doing [19, 28 and 30]

From users: by treating all of users as the same, problems with power users without knowledges of the system [17]; problems with representative users do not really understand others' needs [20]; users may do not care about that requirement because it does not represent the needs of them	*From users*: persona increases stakeholder empathy, and its data gains better understanding of prospective attackers and their motivation [28]; user happiness can be imagined [29]; users may have a chance to express their real facing problems, their desires and more care about the requirements

4. ROLE & PERSONA COMPARISON

4.1 The Importance of Role

By looking at the holistic view of the users, user role brings up understanding the essentials of user relationships. In particular, "*user role modeling, however, offers a compact way to capture all the essential variants in how various users can and will relate to a new system.* [27]"

Moreover, based on user responsibilities and qualifications, role defines various job functions in an organization. Using role, users can easier resign their job functions; just change the role. The ease of use can be easy to see when new permissions coming as new applications and systems, roles can be revoked from roles as needed [21].

In comparison between role and group, group is a collection of the users, but role combines both a collection of users on one side and a collection of permissions on the other [21]. Therefore, role helps us to look at the users in a holistic view as a whole. This view includes their permissions, their responsibilities, their qualifications and their job functions.

4.2 The Importance of Persona

On the other hand, persona looks at users as individuals in details. Particularly, persona leads more understanding about target customers' behaviors and their needs [17]. Also, persona help designers focus and make users as the real people [19] in the real world. However, not only understanding users' perspectives, needs and goals is enough, but persona also shows up the tasks that they intend to use the system for and context within their work [19, 28 and 30].

The last significant point here is persona helps the reducing biases with developers from the word *user* [29]. These biases come from the assumption of developers about what should users do because the word *user* is unspecific or general.

4.3 Role & Persona Comparison

In this section, we give fully comparisons such as the comparison of the differences between role and persona (Table 3). In addition, the coherent relationships between role and persona are showed in Table 4. Next, Table 5 sums up the ideas or methods of role and persona combination; however, these methods/ideas are also the discussions and including many drawbacks. Finally, the reasons why the combination is hard to be done are displayed in Table 6.

Table 3. Role and persona differences

Role Holistic Views of Users	Persona Detail Views of Individuals
Understand the user relationships [21 and 27]	Misunderstand the users relationships [12, 18, 27]
Decrease empathy [10, 17, 20, 22]	Increase empathy [19, 25, 29, 28 and 30]
Focus on *group* or *entity* [10 and 15]	Focus on *people* or *individuals* [25 and 26]
Focus on *job functions*, *responsibilities*, *skills*, *permissions* of a *group* [10, 21 and 31]	Focus on the *needs*, *behaviors*, *goals* of *specific users* [17 and 19]
Focus on *characteristics of users* [27,29 and 30]	Focus on *characters of individuals* [27, 29 and 30]

Table 4. Role and persona relationship

Role *versus* Persona
A persona is an *imaginary representation* of **a user role**. [12]
Personas focus *on users* and **roles** focus *on their relationship*, both of them are *complement each other.* [27]
Role is about *characteristic* focus *designer attention* while **persona** is about *character* focus *identification* [29 and 30]. Therefore, **persona** can become the *personification* of **roles**. [27]
Role and persona is clearer in *accurately targeting the audience and product design.* [31]
User roles feed the persona *creation process* while **persona** serve as *primary design communication vehicle* within the product design and development teams. [31]
Persona *provides information* about task, goal and scenario and **multi-roles** or just a few responsibilities of a role *are performed by* **one persona**. [31]

Table 5. Previous role & persona combination ideas

Idea/Method	Disadvantages
Mike Cohn [12]	Just about user-story, no place for persona-story Persona cannot be specific in its advantages because of limited market and demographic research Persona is too long that can fix in one card Cannot write a story for single perspective
Larry Constantine and Lockwood [27]	User role is more than enough for developers because to understand the persona, it is not an easy part. User role is much more simple and faster to develop while persona is hard to be done because of limited time and resources
William Hudson [10]	Many disadvantages of using role without realizing the advantages of roles
S. Faily and J. Lyle [29]	Not focus on software requirement, but engineering activities Some of the attributes are not suitable to software requirement, especially in Agile.
V. Hill and V. Bartek [31]	Useful to understand the coherent relationship between role and persona, but not focusing on improving software requirement.

Table 6. Previous role & persona combination ideas

Difficulties	Explanations
Complication	Role is simple, easy to understanding and faster to develop [12 and 27]; persona is too long, too detailed, complicated, limited time, resources [18 and 27].

Empathy difficulties in developers	Developers make many efforts develop persona or understand users and it is not easy [10 and 27].
Empathy understanding - User's needs and behaviors	So many perspectives such as storytelling, scenario [12 and 31]. However, how to tell the story and how storytelling gains empathy are still the discussions.

5. ROLE & PERSONA COMBINATION: HUMAN STORIES

5.1 Combination Criteria

Based on the role and persona comparison above, to overcome role and persona combination difficulties, these criteria were introduced:

5.1.1 The combination should be simple

Stories should be included in one card [13] while persona stories seem too long. We would like to make the combination as simple as possible. Moreover, according to The Stacey Matrix [32], if the combination is simple, the requirement will be close to certainty and agreement whereas it will be far from certainty and agreement.

5.1.2 The combination should enhance developers' memory and attract their attention.

Most of the developers are not working on just one project at the same time [33]. Especially in Agile Software Development, developers work with iterations, and each iteration is typically last from one to four weeks [3]. Despite the fact that the role in user stories is often repeated [10] and the persona in persona stories is often too long [10 and 12], the question is how can we draw attention and enhance the developers' memory in the short term like iterations with a lot of issues.

5.1.2.1 We may forget the advantages of Mind-map

"*A Mind-map is logically a simple tree structure of keywords, but has a radiant shape with colors, pictures and drawings*" [34]. Roger Sperry, a Nobel Prize winner in philosophy 1989, found that the cerebral cortex has two hemispheres, and the cortex tends to divide the major intellectual functions between them. Therefore, using Mind-map is a powerful tool to increase memory rather than words [34]. Moreover, Mind-map was also applied to written material to improve memory for medical information [35].

5.1.2.2 The pattern that draws attention

The pattern that draws attention: one of the studies about eye tracking when reading these contents, based on 232 users looked at thousands of Web pages [36]. They found that orderly users read firstly the first horizontal movement, secondly read the second horizontal movement and finally read the vertical movement. The reason why was explained by Jakob Nielsen about F Pattern, the slow-motion replays of users' eye movements as they read and scan across a page seems like the F-Pattern [36]. However, sometimes the pattern looks like an E than an F (Figure 1).

Another research is about the effectiveness of an ordered form and a non-form. In particular, our recognition from an ordered form is easier than the making of disorder, non-form [37].

Figure 1. Eye Movements Scan like an F and E

5.1.2.3 The effect of colors

A research found that the impression of sight memory for the product is within just 0.67s while the first impression comes from the colors counts for 67% of purchasing process. The result of this research reveals that colors also guild the customers' recognition of brands [38].

Furthermore, another framework using a continuous recognition task with colored and monochrome gray-scale images of nature scenes at short exposure durations proves that color also increases recognition memory by conferring an advantage during encoding and by strengthening the encoding-specificity effect [39].

However, each color tells different meanings, choosing the wrong color may deliver the wrong messages [38] or change the entire purpose. Based on a research tracks more than 600 participants' performance from 2007 to 2008, the results show that red boosted performance on detail-oriented tasks such as memory retrieval and proofreading as much as 31% compared to blue, while blue boosted creative tasks such as brainstorming [40 and 41].

5.1.3 The combination should know the customers

"*Requirement is too often seen as a stenographer's task* [42]"; however, the big question is how to know what customers want and need. In fact, people do not know what they want until they see it [42]. In the sales conversation, clients do not care about our expertise, but they care about their condition and how we improve it [43]. And, we should consider that we mostly design the software for the users by non-engineers [18], so any software is designed based on assumptions are often faraway from user expectation because of empathy reduction [10]. Therefore, knowing our customers is one of those principles that has value at almost level of our system [44].

Moreover, to create meaningful innovations, we need to know our users and care about their lives. To comprehend what they need, what they want and what is meaningful to them, *empathy* is one of the key answers. The question is how to gain *empathy* for who they are and what is important to them. The answer is observation and engagement [45]. In particular, "*stories can be a powerful tool for persuasion, useful in the context of understanding customers, building brands and leading teams* [46]". For example, Steve Jobs, co-founder of Apple teaches a lesson and illustrates the importance of staying true to yourself [47].

How to tell the story, Morin suggests that using photos and visuals to keep it simple. "*We include text because some people do read it, but the photos and headline of the projects should explain it well enough for the user to be able to make the project*" [48]. In conclusion, to know what customers want and need, we should let customers raise their voice (The Voice of Customer – VOC [49]) by telling their stories.

5.2 Human Stories

5.2.1 The name

Combination among role, persona and storytelling should be showed up in the name. Reading the book The Storytelling Animal: How Stories Make Us Human [50] by Jonathan Gottschall, we come up with the name *Human-Story*.

5.2.2 Simplicity

Combination should be fixed in one single card [10, 12 and 13]; should concern about resources [27] and time [18].

5.2.3 Mind-map, Patterns & Colors

First of all, combination shows up the mind-map, patterns (F-Pattern, shape, etc.) and all of the elements into order. Secondly, red is chosen to express important information such as *what* and *why* while blue is chosen to express creativity such as *storytelling*. Finally, the content would be as *first vertical movement-who* (image in sharp; demographics such as name, age, gender, organization and location; permission- *role* and expression); *first horizontal movement-what* (in red); *second horizontal movement-why* (in red); *third horizontal movement-storytelling* (in blue).

5.2.4 Storytelling

Which story should be told? Throughout our adolescence, we have had hundreds, thousands, or even millions of stories to tell. Should we tell the story on daily routine at work? Or, Should we tell the story about our personal life? Or the story of our job? These questions should be asked whenever letting users tell their stories. Most of the existing storytelling techniques are often too long, too details with limited time and resources as discussed. In fact, we would like to say that every story is important and understand someone before developing something for them is also important. However, importance means it serves our purpose. For example, one of the users wants to implement chatting service in his/her own website, the story should be told is without built-in chatting service, what and how he/she did before. His/her story, for example, would be "*I'm using the social networks and my personal email to contact with the customers. However, some of the customers using Facebook or Twitter, some of them using emails, and some of them even do not have emails or social accounts. In some situations, I'm confused because I have to manage so many my accounts in multi-service and sometimes I even do not know who they are.*" In the previous forms such as "*As a user, I want to implement a chatting service on my own website, so that I can contact with my customer easier*", without the real story, we could not know what it looks like and how to do it.

5.2 5 Result

Using the example above, the human-story is about Sophia Loren would be like Figure 2.

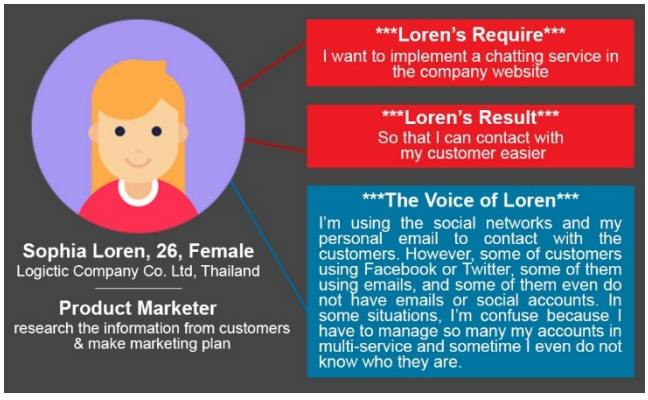

Figure 2. Sophia Loren – Human-Story about implementing chatting service

6. EVALUATION

6.1 INVEST Criteria – INVEST Grid

6.1.1 Existing methods

Although there are many different perspectives to measure the quality of software requirement exist such as Quality User Story framework (QUS) consists 14 quality criteria that a written story should strive [9]. However, QUS is just suitable to apply for user-story itself only. Other quality criteria [51] recommend eight quality characteristics; however, these of criteria are not developed for Agile environment nor our stories.

6.1.2 INVEST grid

The well-known criteria for a good quality Product Backlog Item (PBI) in Agile software development were introduced by Bill Wake in Extreme Programming Explored and Refactoring Workbook called INVEST – Independent, Negotiable, Valuable, Estimable and Small. Mike Cohn mentioned about six attributes of INVEST in [12] for a good story. Based on INVEST criteria, INVEST Grid [52] as introduced by using the scale to evaluate the quality of a story in Agile Requirement Management.

6.1.3 Sample data

In software requirement, there are many different types of requirements [53, 54 and 55]. In this paper, we classify the data of three stories (human, persona and user-story) into four types of software requirement [54] such as functional requirements - FRs, non-functional requirements - NFRs, behavioral requirements - BRs and development quality requirements - DQRs.

6.1.4 Results

As the result, in most of the types of requirements, persona-story and human-story, the scale values are much higher than user-story. For instance, in FRs, the values of human-story and persona-story are 12 and 11, but 4 for user-story. Like away, NFRs and DQRs, both of the stories are greater than user stories. However, in BR, the values among three stories are slightly different. In details, persona story value is 11 while user-story and human-story are remained the same value, 8.

6.1.5 INVEST grid discussion

It is too early to conclude that persona-story and human-story are better than user-story because based on our criteria and arguments as well as disadvantages and advantages of using role and persona, the Grid could not give us a clear explanation. However, the INVEST Grid sheds a holistic view on different types of requirements showing us a comparison among three stories. Therefore, in next section, we devise a quality checklist.

6.2 Agile Requirement Quality Checklist

6.2.1 Existing methods

In Agile quality framework checklist for the feature request, the checklist bases on *Completeness*, *Uniformity*, *Consistency* and *Correctness* [56]. However, although some perspectives about Agile requirement exist, it could not serve our purpose in each specific story.

6.2.2 Quality Checklist

The checklist (Table 7) is not the standard for any kinds of software requirement in Agile, we just use it in the case to make sure that our proposals meet our goals.

Table 7. Agile Requirement Quality Checklist

Agile Requirement Quality Checklist Criteria
The Combination Should Be Simple
All of the information of requirements can be fixed in one single card – content concerns. [10, 12 and 13]
All of the information of requirements can track easily – resource concerns. [27]
All of the information of requirements could not take much time for researching – time concerns. [18]
The Combination Should Enhance Developers' Memory and Their Attention
Increasing developers 'memory by using mind-map [34 and 35]
Drawing developers' attention by using Patterns (F and E Patterns or any) [36 and 37]
Drawing developers' attention by applying all of the elements of requirement in order [37]
Drawing developers' attention and increasing developers' memory by using colors [38, 39, 40 and 41]
The Combination Should Know The Customers/Users
The real stories of the users/customers are telling (such as Scenario) [43, 45, 46, 47 and 48]
Empathy Understanding
Reducing the bias without treating all of the users as the same manner – developer concerns [28]
Dealing with the real user with the real problems without assumptions – developer concerns [19 and 29]
Understanding the needs specific users – developer concerns [25]
Understanding behaviors of specific users – developer concerns [25]
Understanding the tasks that users doing – developer concerns [19,28 and 30]
Focusing on the needs of specific users – user concerns [17 and 19]
Focusing on the goals of specific users – user concerns [17 and 19]
Focusing on the behaviors of specific users – user concerns [17 and 19]
Focusing of characters of individuals in general such as demographics, images – user concerns [27,29 and 30]
Focusing of characters of in details such as personal information, skills, activities, habits, etc. – user concerns [27, 29 and 30]
User Relationships Understanding
Looking at the holistic view at the users [12]
Focusing on group and entity [10 and 15]
Understanding the essentials of user relationship such as job functions [10, 21, 27 and 31]
Understanding the essentials of user relationship such as permissions [10, 21, 27 and 31]
Understanding the essentials of user relationship such as responsibilities [10, 21, 27 and 31]
Understanding the essentials of user relationship such as skills [10, 21, 27 and 31]
Understanding the essentials of user relationship such as qualifications [21]

6.2.3 Sample data

We classify one same story with the same information and requirement into three stories (human, user and persona-story). The checklist will be calculated by the percentages of each criterion and the overall percentages of all stories.

6.2.4 Result

The results show that human-story can fill the gap between user-story and persona-story as well as strengthen the advantages and remove the disadvantages of two other stories. In overall, the percentage of the completeness in human-story is 84% while user-story and persona-story are 44% and 56%. In details, for empathy understanding, one of the significant advantages of persona, persona-story counts for 100%, human-story – 90%, but user-story – just only 30%. On the other hand, user relationships understanding, one of the main benefits of role, user-story counts for 71%, human-story – 57%, persona-story – just only 28%. Another criteria are also significant difference such as *simplicity* (human-story and user-story – 100%, but persona – 0%); *knowing customers/users* (human-story and persona-story – 100%, but user-story – 0%) and *attention and memory enhancement* (human-story – 100%, persona-story – 25%, but user-story – 0%).

7. LIMITATION

7.1 Lack of Practices

There is nothing such as perfect and our combination is not an exception. In addition, we may able to accept that there is a gap between theories and practices. The thing is we do not have any current projects that applying three or two different techniques of writing software requirements – most of them using user-story. In most of the Agile projects, persona-story is quite new while human-story in another new thing that we would like to know how the actually practical results are.

7.2 Lack of Evaluation Methods

The INVEST Grid was used to evaluate three stories is not enough to come up with a realistic result and conclusion. In general, INVEST criteria is using to measure the quality of items in product backlog in Agile; however, our stories are not simple as the items in backlog. On the other hand, Agile requirement quality checklist is good to measure our goals. However, this checklist could be hard or even impossible to cover all of the quality perspectives of Agile requirements. Therefore, we could leave this issue in future research.

7.3 Lack of Data Variety

The sample data shows us the variety of different types of requirements and different kinds of writing software requirement. However, we do not examine on a large data of many requirements to see other perspectives and more reliable results.

8. CONCLUSION

In this paper, we showed up how the importance of the quality in Agile software requirement and different perspectives of its quality improvement. Two most popular ways in describing software requirement were indicated in this papers are user stories and persona stories. In particular, user-story was described in its history, definition, characteristics and its improvement as well as persona-story. The point should be noticed here is the role and persona are being played in these two stories. This paper made a comparative study between them (role vs persona) in their differences, strengths, weaknesses and relationships. Finally, the human-story was introduced could fill the gap between two existing methods and strength the advantages of using role and persona into one story based on the augments and our criteria.

Despite the inevitable limitation, our evaluation methods can give us a holistic view of differences among three stories in different perspectives such as quality in INVEST criteria and the criteria in Agile Requirement Quality Checklist. The results showed that human-story is a promising technique in writing Agile software requirements that may solve the problems of the existing ones without removing their benefits.

9. FUTURE WORKS

9.1 Agile Software Requirement Quality

In the future, we may think of developing a quality framework to measure software requirement quality, give a more precisely perspectives, reduce the biases in our human-story in comparison with two stories – user and persona stories and also fix the limitations of the existing methods.

9.2 The Software for Visualizing Human Stories

In Scrum or Kanban, requirements are written down on a piece of paper and stuck on board. For our human-story, we would like to build up the application that can generate the file or material to serve other purposes.

9.3 Agile Software Requirement Prioritization

This human-story leads us to a future research about human stories prioritization to know which human stories should be considered as the most important stories and should build it first and which ones are less important than others. In particular, INVEST criteria and Mike Cohn mentioned that user-story should be independent ideally [12].

10. REFERENCES

[1] Stober, T., & Hansmann, U. *Agile Software Development: Best Practices for Large Software Development Projects*. Springer Publishing, New York, 2009.

[2] Awad, M. A. A comparison between agile and traditional software development methodologies. In *The University of Western Australia, Australia* (2005), Unpublished doctoral dissertation.

[3] Cockburn, A. *Agile Software Development: The Cooperative Game (2nd Edition)*. Pearson Education, 2007.

[4] Popli, R., & Chauahn, N. Managing uncertainty of story-points in Agile software. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (2015), IEEE, 1357-1361.

[5] Wang, X., Zhao, L., Wang, Y., & Sun, J. In *The Role of Requirements Engineering Practices in Agile Development: An Empirical Study*. CCIS. Springer, 2014.

[6] Abrahamsson, P., Fronza, I., & Moser, R. Predicting Development Effort from User Stories. In *2011 International Symposium on Empirical Software Engineering and Measurement* (2011), IEEE, 400-403.

[7] Dumas-Monette, J.-F., & Trudel, S. Requirements Engineering Quality Revealed Through Functional Size Measurement: An Empirical Study in an Agile Context. In *Software Measurement and the International Conference on Software Process and Product Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)* (2014), IEEE, 222-232.

[8] Buglione, L. The Next Frontier: Measuring and Evaluating Non-Functional Productivity. *Metric View*, 6, 2 (August 2012), 11-14.

[9] Lucassen, G., Dalpiaz, F., & Werf, J. M. Forging high-quality User Stories: Towards a discipline for Agile Requirements. In *Requirements Engineering Conference (RE), 2015 IEEE 23rd International* (2015), IEEE, 126-135.

[10] Hudson, W. User Stories Don't Help Users: Introducing Persona Stories. *ACM Special Interest Group on Computer-Human Interaction, 20*, 6, (2013), 1-4.

[11] Ghani, I., Jawawi, D. N., Dorairaj, S., & Sidky, A. *Emerging Innovations in Agile Software Development*. IGI Global, 2016.

[12] Cohn, M. *User Stories Applied: For Agile Software Development*. Addison-Wesley Professional, 2004.

[13] Jeffries, R. *Essential XP: Card, Conversation, Confirmation*. (2001). Retrieved from http://ronjeffries.com/xprog/articles/expcardconversationconfirmation/

[14] Yu, E., & Mylopoulos, J. Understanding "why" in software process modelling, analysis, and design. In *Proceedings of 16th International Conference on Software Engineering* (2002), IEEE, 159-168.

[15] Jacobson, I., Booch, G., & Rumbaugh, J. *The Road to the Unified Software Development Process*. Cambridge University Press, Cambridge, United Kingdom, 2000.

[16] Jacobson, I., Christerson, M., Jonsson, P., & Övergaard, G. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison Wesley Longman, Redwood City, CA, USA, 2004.

[17] Miller, G., & Williams, L. Personas: Moving Beyond Role-Based Requirements Engineering. *Microsoft and North California State University.* (2006). Retrieved from http://agile.csc.ncsu.edu/SEMaterials/Personas.pdf.

[18] Cooper, A. *The origin of personas*. (May 12, 2008). Retrieved from Cooper: https://www.cooper.com/journal/2008/05/the_origin_of_personas

[19] Cooper, A. *The Inmates Are Running the Asylum*. Macmillan Publishing, Indianapolis, IN, USA, 1999.

[20] Grudin, J., & Pruitt, J. Personas, Participatory Design and Product Development: An Infrastructure for Engagement. In *The Participatory Design Conference* (Malmö, Sweden 2002), 144-161.

[21] Sandhu, R., Coyne, E., Feinstein, H., & Youman, C. Role-based access control models. *IEEE*, 29, 2 (1996), 38-47.

[22] Arnott, D. *Using Personas for Writing User Stories*. (2016). Retrieved from FrontRowAgile: https://www.frontrowagile.com/blog/posts/29-using-personas-for-writing-user-stories

[23] Nordgren, L. F., & McDonnell, M.-H. M. The Scope-Severity Paradox: Why Doing More Harm Is Judged to Be Less Harmful. *Social Psychological and Personality Science*, 2, 1 (2011), 97-102.

[24] Sears, D. O. The person-positivity bias. *Journal of Personality and Social Psychology*, 44, 2 (1983), 233-250.

[25] Williams, F. H. *The A.A.R.T. of Writing User Stories*. (2013) Retrieved from Agile Prague: http://agileprague.com/pool/vzor/upload/AARTproposalDraft4.pdf

[26] Ambler, S. W. *Personas: An Agile Introduction*. Retrieved from Agile Modeling: http://www.agilemodeling.com/artifacts/personas.htm

[27] Adlin, T., & Pruitt, J. Chapter 8. In *The Persona Lifecycle: Keeping People in Mind throughout Product Design*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2015.

[28] Faily, S., & Fléchais, I. Barry is not the weakest link: eliciting secure system requirements with personas. In *Proceedings of the 24th BCS Interaction Specialist Group Conference* (Oxford University, BCS Learning & Development Ltd. Swindon, UK 2010), ACM, 124-132.

[29] Faily, S., & Lyle, J. Guidelines for integrating personas into software engineering tools. In *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems* (University of Oxford, London, United Kingdom 2013), ACM, 69-74.

[30] Cooper, A., Reimann, R., & Cronin, D. *About Face 3: The Essentials of Interaction Design*. Wiley; 3rd edition, 2007.

[31] Hill, V., & Bartek, V. Telling the user's story. In *Proceedings of the 2007 symposium on Computer human interaction for the management of information technology* (Cambridge, Massachusetts 2007), ACM.

[32] Stacey, R. *Strategic Management and Organisational Dynamics: The Challenge of Complexity*. Prentice Hall, Harlow, 2002.

[33] Hohman, M. M. Estimating in Actual Time. In *Proceedings of the Agile Development Conference* (Northwestern Univ., Evanston, IL, USA 2005), IEEE.

[34] HIRANABE, K. *Exploring User Requirements through Mind Mapping.* (2008). Retrieved from Change Vision: http://www.change-vision.com/en/ExploringUserRequirementsThroughMindMapping_Letter.pdf

[35] Farrand, P. A., Hennessy, E., & Hussain, F. The efficacy of the 'mind map' study technique. *Medical Education*, 36, 5 (June 2002), 246-431.

[36] Nielsen, J. *F-Shaped Pattern For Reading Web Content*. (April 2006). Retrieved from Nielsen Norman Group: https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/

[37] Frutiger, A. *Signs and Symbols: Their Design and Meaning*. Van Nostrand Reinhold, New York, 1989.

[38] Chang, W.-L., & Lin, H.-L. The impact of color traits on corporate branding. *African Journal of Business Management*, 4, 15 (November 4, 2010), 3344 - 3355.

[39] Spence, I., Wong, P., Rusan, M., & Rastega, N. How Color Enhances Visual Memory for Natural Scenes. *Association for Psychological Science* (2006), 1 - 6.

[40] Mehta, R., & Zhu, R. Blue or Red? Exploring the Effect of Color on Cognitive Task Performances. (University of British Columbia 2009), Sciencexpress.

[41] University of British Columbia. *Effect of Colors: Blue Boosts Creativity, While Red Enhances Attention to Detail.* (2009, February 6). Retrieved from ScienceDaily: https://www.sciencedaily.com/releases/2009/02/090205142143.htm

[42] Maiden, N., Robertson, S., & Robertson, J. Creative Requirements: Invention and Its Role in Requirements Engineering. In *ICSE '06 Proceedings of the 28th international conference on Software engineering* (Shanghai, China 2006), ACM, 1073 - 1074.

[43] Dawson, T. *Improving the customer's condition*. (February 16, 2016). Retrieved from LinkedIN: https://www.linkedin.com/pulse/improving-customers-condition-business-transformation-guidance

[44] Kahle, D. *Knowing Your Customers*. (2005). Retrieved from Dave Kahle & The DaCo Corporation: http://www.davekahle.com/article/knowing.html

[45] Plattner, H. . *An Introduction to Design Thinking - PROCESS GUIDE.* Retrieved from Institute of Design at Stanford: https://dschool.stanford.edu/sandbox/groups/designresources/wiki/36873/attachments/74b3d/ModeGuideBOOTCAMP2010L.pdf?sessionID=9435c2b6ec2fd3386cee3ca7946c8a5290fb90bb

[46] Aaker, J. *The Power of Stories in Business.* (2014). Retrieved from Stanford Graduate School of Business: http://stanford.edu/class/gsbgen542/cgi-bin/files/PoS%20Syllabus%2012_final_2x2layout.pdf

[47] Friedman, S. D. Clarify What's Important to You. In *Chapter 2*. Harvard Business Press, Boston, Massachusetts, 2008.

[48] Aaker, J., & Schifrin, D. *How Stories Drive Growth: Skype.* (April 28, 2014). Retrieved from Stanford Graduate School of Business: https://www.gsb.stanford.edu/faculty-research/case-studies/how-stories-drive-growth-skype

[49] Griffin, A., & Hauser, J. R. The Voice Of The Customer. *Marketing Science*, 12, 1 (1993), 1-27.

[50] Gottschall, Jonathan. *The Storytelling Animal: How Stories Make Us Human*. Mariner Books, 2013.

[51] IEEE Computer Society. IEEE Recommended Practice for Software Requirements Specifications. In *The Institute of Electrical and Electronics Engineers* (New York, NY 10017-2394, USA 1998), IEEE.

[52] Buglione, L., & Abran, A. Improving the User Story Agile Technique Using the INVEST Criteria. In *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement* (2014), IEEE, 49-53.

[53] Faulk, S. R. Software Requirements: A Tutorial. In *Naval Research Laboratory* (Washington, D.C 1995), 1-22.

[54] Alspaugh, T. A. *Kinds of Software Requirements*. (May 1, 2015). Retrieved from Thomas Alspaugh : http://www.thomasalspaugh.org/pub/fnd/kindsOfRequirements.html#behavioral-developmentalQuality

[55] Shrivathsan, M. *Types of Software Requirements*. (April 4, 2012). Retrieved from Accompa, Inc: http://rmblog.accompa.com/2012/04/types-of-software-requirements/

[56] Heck, P., & Zaidman, A. A Quality Framework for Agile Requirements: A Practitioner's Perspective. *ArXiv: 1406.4692v1, [sc.SE]*. (June 18, 2014). Retrieved from https://arxiv.org/pdf/1406.4692.pdf