

# QVD: Improving Quality – User Story as a Quality Value Driver (QVD) in Software Development

MOSHE DAVIDIAN

Department of Management  
Ben-Gurion University of the Negev  
P.O.Box 653, Beer-Sheva, Israel 84105  
ISRAEL

YOTAM LURIE

Department of Management  
Ben-Gurion University of the Negev  
P.O.Box 653, Beer-Sheva, Israel 84105  
ISRAEL

SHLOMO MARK

Department of Software Engineering  
SCE - Shamon College of Engineering  
84 Jabotinsky St. Ashdod, 77245 Israel  
ISRAEL

*Abstract:* - The term "software quality" is widely used, and although it has many definitions, no one definition is universally accepted. Often, the definition refers to specific phases of the software development process and not to software as a whole. In our article, in order to improve the quality of the software, we decided to improve the quality of the phases. To do that, we will focus attention on the drivers that advance quality value and use a new concept called QVD - quality value drivers. We focused on the requirements phase of software development, which probably ranks as the crucial first step. Hence, as a QVD we present the idea of "User Story" (US), a short and simple description of a functionality valuable to a user of a system. In the study, a comparison is made between requirements written by the US method and requirements that are not. After analyzing the results, we concluded that requirements written by the US method have been better understood and evaluated as less difficult to develop. The overall quality rating of their writing is higher than previously. In addition, learning the US method gives the person the ability to better assess the quality of requirements. Finally, improving the writing of the requirements using the US method as QVD improves the quality of the development process, eventually improving the software quality.

*Key-Words:* - QVD, Software quality, User Story

Received: March 7, 2021. Revised: September 10, 2021. Accepted: October 2, 2021. Published: October 23, 2021.

## 1. Introduction

"Software quality" is a term that has been widely used for decades and has more than one definition. It lacks a single accepted definition that is universally applied. Often, the definition depends on the role of the definer in the development process and according

to the type of system developed [1][2][3]. Most of the definitions of software quality can be classified into two main types: Development process quality and Final product quality. In addition, there is a wide variety of different models for assessing the quality of the software based on different criteria and characteristics [4] [5]. . Moreover, methods that

support the achievement of product quality goals or requirements are not usually described in the process models, and little is known about the interaction of these methods with other process elements [6] [7]. The ability to define and measure software quality before deploying the system will positively affect the software, reduce maintenance costs and increase the company's reputation that developed the software by identifying and fixing problems before deploying the system. Since the definition of quality depends, among other things, at the phase of the development process to which we relate, when we want to improve the quality of the software, we need to improve the quality of the development process or the different phases [8]. To do that, we will focus attention on the drivers that advance quality value and introduce a new concept called QVD - quality value drivers. QVD is a managerial approach (i.e., practices, roles, ceremonies, artifacts) that delivers ethical values to the process, within a specific phase, consequently making measurable improvements that will affect the overall quality of the process, eventually providing better quality software. As has been demonstrated in the previous researches [9][10], there is a significant correlation between the inclusion of ethical values in the development process and the achievement of improved performance in three quality parameters: schedule, product functionality, and cost.

This article focuses on the requirements phase of the software development process, which probably ranks as the crucial first step. We decided to emphasize the integration of ethical values into the requirements phase and chose to improve values such as communication and clarity of requirements to better reflect the needs and expectations of the customer. The process-focused values of communication and clarity provide for a rich understanding and mutual agreement, understanding other stakeholders' perspectives and enabling better communication by putting everyone on the same page. The requirements phase aims to define what a system should do and the constraints under which it must operate. In this phase, we gather, write, analyze and document the requirements of the product. The software requirements are the description of features and functionalities of the system and represent the expectations of users from the product. Poor decisions in this early phase of development are exceedingly costly in the later stages of software production [11][12].

Hence, as a QVD we present the idea of "User Story". User stories are short and simple descriptions of a software's features from the perspective of the person who needs it. Traditionally, they are handwritten on index cards as they are easy to store, display,

rearrange and distribute to development teams [13]. We check if using the user story method (US) improves the requirements writing technique, which eventually improves the requirement phase and ultimately improves the overall quality of the software.

## 2. The QVD approach.

Since the quality of the software consists of many phases, each phase refers to a part of the development process; in order to improve the quality of the software, we need to improve the quality of the different development phases (gathering requirements, design, code writing, etc.). In order to improve the quality of a particular phase, we will focus attention on the drivers that advance quality value and introduce a new concept called QVD - quality value drivers.

QVD is a managerial approach that delivers quality by introducing particular ethical value-driving processes into the process, consequently aiming to make measurable improvements and evaluations that will affect the overall quality of the process eventually providing better quality software. The QVD acronym includes Quality, Value, and Driver. The steps in the QVD approach are: First, describe the value that you want to achieve or create (Value), e.g., reaching mutual agreement, prioritizing product features. Second, describe the managerial approach (practices, roles, ceremonies, artifacts) that helps achieve the value (Driver), Such as "user stories", which we will discuss below. And finally, describe the metrics and the method used to measure the improvement achieved by adopting the driver (Quality).

As a general definition, a QVD is an activity that adds value to the development process. More specifically, QVD refers to those activities or capabilities that increase profitability, reduce risks, and promote growth according to strategic goals. QVD are the most significant factors impacting the company's success, and they are specific to different industries and companies.

The rapid development of technology that has led to improvement in the capability of artificial intelligence systems and their power, the ethical challenges from these systems are also evolving and increasing. In the software development process, one of the main ethical concerns of the developing company is to choose development methods and practices that will lead to delivering value to the customer. The dependence on the human factor in the development of a software product in particular and a technological product in general constitutes an extensive basis for research and the definition of up-

to-date development methodologies. This is another area where the QVD approach can be successfully integrated as it focuses on the human factor and the integration of ethical values.

The field of ethics in AI systems is not sufficiently researched and research in the field is not approaching the pace at which technology is advancing. The main research premise in this research paper is that the investigation of ethical issues and the development of customized ethical tools during the development cycle (SDLC - Software Development Life Cycle) is extremely important both to ensure the quality of the systems and to ensure that the AI application deals properly with ethical issues. We are currently working on integrating the QVD approach into AI systems in the healthcare field.

The QVD approach has two main characteristics that distinguish it from other methods in the field of software quality. First, this approach focuses on achieving ethical values and integrating them as an integral part of the development process. Second, this approach advocates not only the integration of drivers in the development process but also the presentation of evaluation and measurement methods for assessing the improvement we have obtained from using the driver. In the test case researched in this article, we not only propose to integrate the User Story method as part of the development process but also to emphasize the ethical values that the method provides for the development process, such as: achieving mutual agreement among all stakeholders. In addition, we present how we evaluated and measured the improvement achieved from using the user story method.

### **3. QVD: User Story as a Driver towards better understanding.**

*The Value* - As part of the development process, there was a decision to focus on ethical values that improve the requirements phase. Specifically, we decided that the principal value we want to achieve will be written requirements that better reflect the needs and desires of the customer. In other words, the requirements phase is supposed to allow for a rich understanding by mutual agreement, understanding other stakeholders' perspectives, and enabling better communication by putting everyone on the same page. This is performed through writing the requirements. We will achieve this through the practical technique of using the 'User Story method (US)', which leads to an improvement in several values:

- *Mutual agreement*: US helps to look at the product from each stakeholder's perspective.
- *Representing each of the stakeholders*: US brings clarity regarding the needs of the critical stakeholders.
- *Prioritizing product features*: when the application functionality is described from a user's perspective, it drives the developer to concentrate on the main tasks instead of small insignificant features of the application which are irrelevant at the planning stage.
- *Putting everyone on the same page*: US serves as a bridge between developers and business-minded players and simplifies the process of decision-making when parties with different backgrounds are involved.

*User Story as a Driver* - In order to achieve better-written requirements, we will focus attention on a driver that advances this quality value, the User Story (US) method [14]. The source of US is in the Extreme Programming (XP) methodology [15]. The US became the most common method to handling Agile projects' requirements, and their use has been adopted in many books about agile development [16] [17]. In the spirit of XP, the production of US are kept as simple as possible. US are short, simple descriptions of a feature from the perspective of the person who needs the new capability, usually a user or customer of the system. Traditionally the US are handwritten on index cards as they are easy to store, display, rearrange and distribute to development teams [11]. Each US is written in one or two sentences that represent a small piece of functionality of the final software from the end-user point of view. The US can be written by the customer (in addition to the system designer). The developers refine it by estimating the amount of development time needed to implement its functionality. There are several methods for estimating. One of the accepted methods is poker planning, in which each team assigns each story points that represent the effort required to perform it. The points are usually calculated and given according to the number of working days. A good US is a US that can be developed in few days. If the development team estimates that the implementation of the US will take more than a few days, the developers have the responsibility to split the US into smaller parts [18]. There are several methods to assess and improve user story quality [19] [20]. User stories are commonly estimated using group processes and individual processes. For user stories to be considered good, they should meet the INVEST's criteria, i.e. be (1) Independent, (2) Negotiable, (3) Valuable to users and customers, (4) Estimable, (5) Small, and (6) Testable. When estimation of all user stories is completed, the

user stories are organized for release and iteration planning. The goal of release planning is to let the customer choose the stories to be implemented in the next release. In the iteration planning, each user story is broken up into development tasks. Responsibility for each task is assigned to a team member. The order of implementation is decided by the team and based on multiple factors.

*User Story Structure:* Although there are some differences, all authors agree on three essential components of a user story [21]: The user story will be described as a short piece of text, there will be conversations between the various stakeholders to get a variety of perspectives on the same user story and criteria will be set that will allow deciding whether to accept each user story. Currently, there is no single agreed-upon template used for writing user stories. Indeed, a commonly used pattern relates three elements: Who it is for, a WHAT is expected from the system, and possibly a WHY it is important [22]. This pattern has several concepts for representing these dimensions (can be found in literature or blogs). The most widespread format and de facto standard, popularized by Mike Cohn [14] is: "As a <type of user>, I want <goal>, [so that <some reason>]."

As we wrote earlier the US mostly write on cards, and card is more effective when it containing a very small amount of information. Hence, the recommendation is that the card will contain the following information [11]: US number and title, Person responsible for the story, Date, Estimate of implementation time (typically in days), Risk level of the technology used, US description, and three optional information: Unit test description, User interface needs, Other related story numbers.

*The Quality - Measure the improvement:* To measure the improvement, that we are achieving, in requirements writing as a result of using the US as a QVD, we conducted research that we will present and discuss in the next section.

## 4. The Research

In our research, we wanted to examine whether there is a connection between the knowledge and use of the US method and the quality of writing the requirements (reflected in the values we presented). The dependent variable, which we want to explain, is the quality of writing the requirements. Our predictive variable is the knowledge and use of the US method.

*The research hypothesis:* knowledge and use of the US method leads to the writing of more qualitative requirements.

### 4.1 Research Design

Exploratory research using intervention and comparing the group in which intervention was carried out with the control group, aimed at examining whether the US method improves the way the requirements are written, and more specifically, did we improve the values we chose to achieve. Measuring the impact of US on the quality of the requirements will be done by comparing requirements that are written according to the US method, with requirements that are not written according to the US method.

Our experiment is based on behavioral intervention and is done in three cycles, on three different groups. The first group consists of 20 students, the second group consists of 30 students, and the third group consists of 50 students. Each of the students was required to write five requirements and test five requirements of other students. In total, the experiment was conducted on about 500 requirements.

Our experiment was built from 4 stages:

1. At the first stage of the experiment, all the students received the same task. The task contains a case that involves software specification. The students were required to write five requirements about that software.
2. At the second stage, the students were divided (randomly) into two groups. The first group underwent a process of learning and practicing the user story method (20 minutes lesson then 10 minutes practice). The second group did not learn the subject of the user story method and was a control group.
3. At the third stage, the first group (that learned use of the US) received the task again. This time they had to write the requirements according to the US method that they learned and practiced. (In order to test the hypothesis that their writing technique will be improved.)
4. At the fourth stage - the test stage - each of the two original groups was divided again, thus creating two subgroups within the original two groups. The group that learned the user story method, the intervention group, was divided into subgroups A and B, and the control group that did not learn the user story method was divided into subgroups C and D. Each student from each of the four subgroups received a list of five requirements written by a student from the other subgroup, as shown in the table below (Table 1). The student received five identical questionnaires (one for each of the requirements in the list) containing

several questions and statements regarding the quality of the requirements (see appendix i ).

EXAMINER GROUP		EXAMINEE GROUP
A	→	B
B	→	D
C	→	A
D	→	C

**Table 1 - The test stage**  
 Groups A & B - learned US,  
 groups C & D - didn't learn US

*The population:* The population on which we conducted our research were Software Engineering students from Sami Shamoon College of Engineering. The first round of the research was conducted on students of software engineering from the Beer-Sheva campus, who are at the end of their first year of undergraduate studies. They have already completed at least one software writing course. The second and the third rounds of the research were conducted on students of software engineering from the Ashdod campus, who are in their third year and second year of undergraduate studies.

*The questionnaire:* we developed the questionnaire following our review of the user story method, from which we analyzed the key parts of the user story method that we wanted to investigate. Our research was designed to test whether we achieved improvement in the values we predefined. For example, we want to achieve the value of writing requirements to better reflect the needs and desires of the customer and the value of representing each of the stakeholder's requirements. To test the effect of using the user story method on these values, in the final phase the students answered a questioner in which we entered questions like "How clear is it, for who need to develop the requirement?", "How clear is it, what the action that the user wants to perform in the requirement?", "How clear is it, why the user needs this requirement?" and "How much do you think this understanding contributed to the success of the development process?". Comparing the subjects' responses helped us understand the level of influence that the US method had on the values we presented. The questionnaire can be found in appendix i.

*Data collection procedure:* The questionnaire was filled by approximately 100 students from our sample population. The first part contains general statements about the requirement that the subject should evaluate. Each item is ranked on a five-point scale: from 1 ("absolutely disagree") to 5 ("very agree").

The second part contains specific questions about parts of the same requirement (according to the user story method, the requirement was divided into three parts). Each item was graded on a five-point scale: From 1 ("not at all contribute") to 5 ("contributed very much"). The final

section contains specific questions about the requirement. Each item is rated on a five-point scale: from 1 ("very low") to 5 ("very high"). In order to determine if our hypothesis is correct, we analyzed the ranks the students gave the statements in the questionnaires. For each statement, we calculated the average rank received from each group.

## 5. The Research Results

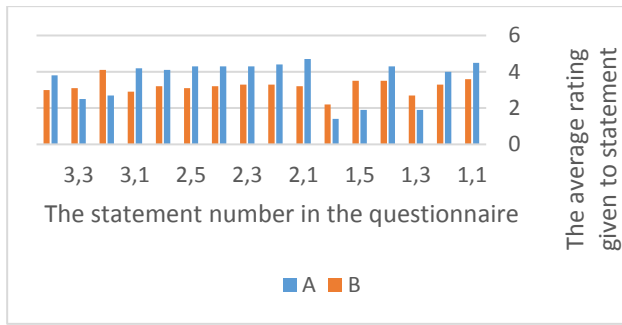
This section presents the main three findings related to the research question. Eventually, to test whether the change in values that we received is a minor change or a significant one, we conducted a T-Test for 2 Independent Means (the P-value we chose is 0.01). The average ranks of all the groups to the research questionnaires can be found in appendix ii.

*First result:* In the first part of the research we examined how the group that learned the user story method evaluates requirements written by a group that did learn the user story method how they evaluate requirements written by a group that did not learn the user story method. (Group A examined requirements written by students who learned the US method. Group B examined requirements written by students who did not learn the US method). The results of the part are summarized in Figure 1. In the T-Test (for each comparison), we got that the P-value is less than 0.01. The results show that the change in values is significant.

Figure 1 shows that group A evaluates the requirements written by those who learned the US method as more understandable (1.1), more detailed (1.2), and can be able to understand the requirements faster (1.4) than those that examined by group B. In addition, it can be seen that requirements written by the US method are more unambiguous (1.5) and do not contain unnecessary details (1.6) relatively to those who were not written by the US method.

The second section of the questionnaire checked the parts of the requirement, and it shows (2.1 – 2.6) that the requirements which are written by those who learned the US method got better quality evaluation in all the categories that checked, relatively to those written by who did not learn the US method.

In the third section of the questionnaire, we can see that the requirements written by those that learned the US method are better structured in terms of syntax (3.1) than requirements that were written by those who did not learn the US method, so it will also be harder to improve their writing (there is less to improve) (3.2). Two important points to note are that the requirements written by those who learned the US method have been evaluated as less difficult to develop (3.3), and the overall quality rating of their writing is higher (3.4) than the requirements which were written by those who did not learn the US method.

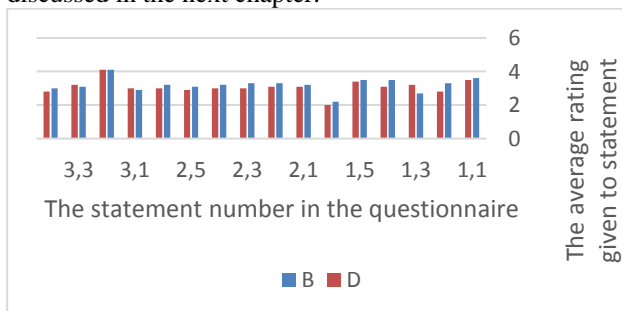


**Figure 1 - Comparison between groups** which learned the US method and evaluates requirements written according to the US method –A, and group in which the requirements was not written according to the US method - B.

*Second result:* In the second part of the research we checked whether the group that learned the US method evaluates requirements objectively and not only searches for the pattern of US. Therefore, we compared the rank that the members of group B (learned US) gave to the questionnaire statement and the rank that the members of group D (did not learn US) gave to the statement of the questionnaire. Both groups examined requirements that were not written according to the US method. The results of the part are summarized in Figure 2.

In the T-Test (for each comparison) we got that the P-value is more than 0.01, except for questions 1.2 (P-value 0.006) and 1.3 (P-value 0.005). The test results indicate that the students assessed the requirements are almost the same. Apart from two cases, for Question 1.2 and Question 1.3, we will discuss this in the Discussion section.

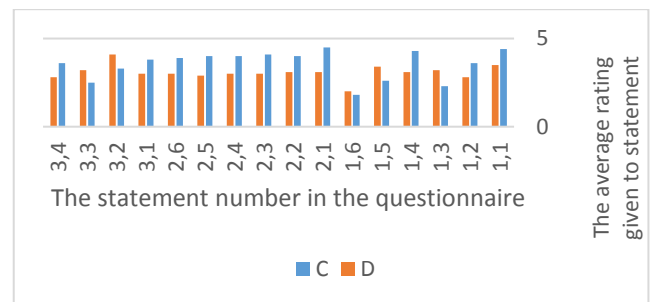
Figure 2 shows that almost all the ranks the students gave to the questions and the statements were similar. Despite the similarity, we will highlight the minor differences that exist. Group B was able to understand the details of the requirement better (1.2) and understand more quickly what they were asked to do in the requirements (1.4) than group D. In addition, group B noticed that the requirements they examined, contained more unnecessary details than group D noticed for requirements written by those who did not learn the US method (1.6). The second section of the questionnaire (2.1 – 2.6), shows that it was easier for group B to divide the requirement to its components than for group D. In section three (3.1 – 3.4) the responses are almost identical. The explanation for these results is discussed in the next chapter.



**Figure 2 - Comparison between the group which learned the US method – B, and group which did not learn the US method - D.** (Both groups examined requirements that were not written according to the US method.)

*Third result:* In the third part of the research we took students who did not learn the US method, and check how they evaluate requirements written according to the US method (group C), compared to how they evaluate requirements not written by that method (group D). The results of the part are summarized in Figure 3. In the T-Test (for each comparison) we got that the P-value is less than 0.01 (except 1.6, the p-value is 0.13). The results showed that the change in values is significant.

Figure 3 shows that in the first section, the requirements written according to the US method are more understandable (1.1, 1.4) and written more sharply (1.5) than requirements not written according to the US method. In addition, those requirements are detailed enough and contain fewer unnecessary details (1.2, 1.6). In the second section of the questionnaire, which checked the parts of the requirement, we can see (2.1 – 2.6) that the requirements written according to the US method received slightly better ratings in all categories examined than requirements not written according to the US method. In the third section of the questionnaire, we can see that the requirements that written according to the US method are better structured (3.1), there is less how to improve the writing (3.2), evaluated as less difficult to develop (3.3), and the overall quality rating of their writing is higher (3.4) than requirements not written according to the US method.



**Figure 1 - Comparison between groups that did not learn the US method and evaluated requirements written according to the US method –C, and group in which the requirements were not written according to the US method - D.**

## 6. Discussion

In this section, we discuss three main points (In line with the three results we presented in the previous section) that we can obtain from the research results. The first point is the difference in the requirements' quality evaluation given by groups who learned the US method to requirements written according to the US method and requirements not written according to the US method. The second point is the difference between the requirements' quality

evaluation given by the group that learned the US method to the evaluation given by the group that did not learn the US method. Both of them evaluate requirements that did not write according to the US method. The third point is the difference in the evaluation given by the two groups that did not learn the US method, to requirements written according to the US method, and to requirements not written according to the US method.

First point: Group A and group B both learned the US method, group A evaluated requirements written by those who learned the US method and group B evaluated requirements written by those who did not learn the US method. As we saw in comparing the results in the previous chapter, the subjects evaluated requirements written according to the US method as more qualitative requirements in the various categories examined.

The high score that group A gave to the requirements is due to a change in the requirement structure of those who learned the US method. In their written requirements, the sentence is divided into three main parts, leading to better understanding than requirements containing one long sentence. The second section of the questionnaire, that checked the parts of the requirement. The requirements are written by those who learned the US method got a better quality evaluation in all the categories that checked than the requirements written by those who did not learn the US method. In this case, we assume that the cause to the better rank was dividing the requirement into three parts representing the parts of 'Who?', 'What?' and 'Why?'. The comparison results strengthened our hypothesis that writing requirements according to the US method improves the quality of requirements. However, these results may be due to the fact that the groups look for a structure of requirements as they were learned, and therefore gave a higher rank to the requirements written according to the US method. To overcome this case, we conducted the next two comparisons.

Second point: Groups B (that learned the US method) and D (that did not learn) were required to evaluate requirements written by students who did not learn the US method. As can be seen in the comparison, the groups' estimates were close in most of the quality indices. From this data, we learn that although group B learned the US method, it did not look only for the US structure. Group B did not give lower rank to requirements that were not written according to the US method, same as group D. This fact strengthens the conclusion of the first point and teaches us that those who learn the US method will not disqualify requirements written according to another method, but will evaluate them objectively.

A few differences there existed in ranks 1.2, 1.3 and a little in 1.4, show that students in group B stated that the requirements were "detailed enough" (relative to what they noted those who did not study the US method) and, in our opinion, that because they learned how to write a requirement briefly and concisely. And this is also why they gave a lower grade for clarity of requirement writing. We also believe that learning the US method has helped them understand the requirements more easily.

In addition, in the second section of the questionnaire, it can be seen that it was easier for students in group B to extract relevant data from the requirement, for example, what role the requirement was written for, and exactly what was required to perform it. We think that this is an important conclusion. In addition to the advantages of writing requirements according to the US method, those who learned the US method improved the ability to read and understand requirements more than those who did not learn the US method, whether the requirements were written according to the US method or not.

Third point: Groups C and D did not learn the US method. Group C, evaluated requirements that were written by those that learned the US method and group D evaluated requirements that were written by those that did not learn the US method. This comparison is intended to see how students who did not learn the US method assess requirements written according to the US method compared to requirements not written according to the US method.

The comparison results show that students who did not learn the US method evaluated the requirements that were written according to the US method as significantly higher quality requirements than requirements that were not written according to the US method. In addition, requirements written according to the US method received a better rank throughout all the parameters examined, both in the general estimation parameters and in those related to specific parts of the requirement. We assume that these unambiguous results can be explained by the fact that writing requirements according to the US method yields a short and clear requirement that includes only the important components for understanding the requirement. This comparison of the results, not only confirms the results we received at the first point but also significantly strengthen our hypothesis,

## 7. Conclusions

In order to improve the quality of the software, we must focus on improving the software development process, which consists of several phases. When we succeed in improving the quality of the phases that will affect the overall quality of the development process and eventually provide better quality to the finished software.

In this article, we focus on the requirements phase, the first phase of the development process. In order to obtain high-quality products, it is necessary to write and manage software requirements carefully. This not only simplifies system design and implementation but also reduces the number of defects in the implementation phase. In order to improve the quality, we proposed integrating quality drivers into the development process by focus attention on the particular and various drivers that serve to advance quality value. We introduced a new concept called QVD - quality value drivers. A managerial approach that delivers quality values to the organization within the development process, consequently aiming to make measurable improvement and evaluation. As a particular test case of a QVD we presented the method of "User Stories". US is a short, simple description, of a feature, from the point of



view of the person who needs the new capability (usually a user or customer of the system). According to the US method, the requirement needs to be written in the format: Who, What, and Why.

In order to test whether the use of the US method leads to an improvement in the desired value, we conducted research of about 500 requirements. Some of the requirements were written according to the US method (the other not). The requirements were written and evaluated by undergraduate students in software engineering. Each requirement was examined according to a questionnaire we developed. After collecting and analyzing the results of the research we conclude that requirements written by those who learned the US method have been more understandable, evaluated as less difficult to develop, and the overall quality rating of their writing is higher than requirements written by those who did not learn the US method. In addition, this method improved the ability to read and understand requirements whether they are written according to the US method or not.

#### References:

- [1] Denning, P.W. (1992). What is software quality? *Communications of the ACM*, 35(1), 13–15.
- [2] Deutsch, M., & Willis, R. (1988). *Software Quality Engineering: A Total Technical & Management Approach* (1st Edition). Prentice-Hall, Englewood Cliffs, NJ.
- [3] Schneidewind, N. F. (2002). Body of knowledge for Software quality measurement. *IEEE Computer*, 35(2), 77–83.
- [4] Mordal, K., Anquetil, N., Laval, J., Serebrenik, A., Vasilescu, B., & Ducasse, S. (2013). Software quality metrics aggregation in industry. *Journal of Software: Evolution and Process*, 25(10), 1117–1135.
- [5] Khan, H. A. (2013). Establishing a defect management process model for software quality improvement. *International Journal of Future Computer and Communication*, 2(6), 585–589.
- [6] Gorla, N., & Lin, S. C. (2010). Determinants of software quality: A survey of information systems project managers. *Information and Software Technology*, 52(6), 602–610.
- [7] Chassidim, H., Almog, D., Mark, S. (2021). "Continuous Software Engineering and Unit Testing: From Theory to Practice," *WSEAS Transactions on Computer Research*, vol. 9, pp. 113-124.
- [8] Lurie, Y., & Mark, S. (2016). Professional Ethics of Software Engineers: An Ethical Framework. *Science and engineering ethics*, 22(2), 417–434.
- [9] Abdulhalim, H., Lurie, Y., & Mark, S. (2018). Ethics as a Quality Driver in Agile Software Projects. *Journal of Service Science and Management*, 11(01), 13.
- [10] Sohacheski, D.B. Lurie, Y. Mark, S. (2021) "Software Identifier Naming Conventions & Dictionary," *WSEAS Transactions on Computer Research*, vol. 9, pp. 125-135.
- [11] Rees M.J. (2002) A Feasible User Story Tool for Agile Software Development? *Proceedings of the Ninth Asia-Pacific Software Engineering Conference (APSEC'02)*.
- [12] Bakun-Mazor, D., Davidian, M., and Mark, S. (2019). "User story technique in scientific software product project: Measuring rock block displacement", *AIP Conference Proceedings* 2186, 170007.
- [13] Chassidim, H., Almog, D., & Mark, S. (2018). "Quality development (QDev) unit in a software engineering school". *World Transactions on Engineering and Technology Education* 16 (3), 249-253.
- [14] Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Boston, MA: Addison Wesley
- [15] Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change*. (2nd ed). Addison-Wesley: Boston, MA.
- [16] Cockburn, A. & Highsmith, J. (2001). *Agile Software Development: The Business of Innovation*. *Computer*, 34(9), 120–127.
- [17] Cockburn, A. (2002). *Agile Software Development*. Addison-Wesley: Boston, MA.
- [18] Mahnic, V., & Hovelja, T. (2012). On using planning poker for estimating user stories. *Journal of Systems and Software*, 85(9), 2086–2095.
- [19] Buglione, L., & Abran, A. (2013). 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*, 49–53, IEEE.
- [20] Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. & Brinkkemper, S. (2016). Improving agile requirements: the quality user story framework and tool. *Requirements Engineering*, 21(3), 383–403.
- [21] Dimitrijevic, S., Jovanovic, J. & Devedzic, V. (2015). A comparative study of software tools for user story management. *Information and software technology*, 57, 352–368.
- [22] Wautelet, Y., Heng, S., Kolp, M., Mirbel, I., & Poelmans, S. (2016). Building a rationale diagram for evaluating user story sets. In *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*, 1–12, IEEE.

#### Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

Moshe Davidian performed this research as part of his Ph.D dissertation. When Yotam Lurie and Shlomo Mark are Moshe's supervisors in his Ph.D. dissertation.

#### Appendix i:

##### The questionnaire

Below is a list of five requirements for developing a school management system, read the requirements and answer the following questionnaire: (Answer for each requirement separately)



### User Story survey questionnaire

Name-code: \_\_\_\_\_ Year of Study: \_\_\_\_\_ Group: \_\_\_\_\_  
Do you agree to participate in the study? Yes / No  
Have you learned the "User Story" topic? Yes / No  
Requirement Number: \_\_\_\_\_

#### First section:

For each of the questions the answer bar represents:  
1 - absolutely disagree, 5 - very agree

Topic	Q#	Question
general	1.1	The Requirement was understandable.
	1.2	The requirement is detailed enough.
	1.3	The requirement is <b>not</b> written clearly.
	1.4	I understood the requirement on first reading without the need for processing.
	1.5	The requirement can be interpreted in a number of ways.
	1.6	The requirement contains unnecessary details.

#### Second section:

1 - Not at all clear, 5 - Very clear  
1 - Not at all contribute, 5 - Contributed very much

Topic	Q#	Question
Parts of the requirement	2.1	How clear is it, for who needs to develop the requirement?
	2.2	How much do you think this understanding contributed to the success of the development process?
	2.3	How clear is it, what is the action that the user wants to perform in the requirement?
	2.4	How much do you think this understanding contributed to the success of the development process?
	2.5	How clear is it, why the user needs this requirement?
	2.6	How much do you think this understanding contributed to the success of the development process?

#### Third section:

1 - very low, 5 - very high

Topic	Q#	Question
Summary	3.1	How well do you think the syntactic structure of the requirement? (regardless of its meaning.)
	3.2	Is it possible to write the demand better than the form in which it was written?
	3.3	What is the difficulty in developing the code for this requirement?
	3.4	How would you rate the quality of the requirement?

#### Appendix ii:

##### The Questionnaire results

QUESTION	GROUP A	GROUP B	GROUP C	GROUP D
1.1	4.5	3.6	4.4	3.5
1.2	4	3.3	3.6	2.8
1.3	1.9	2.7	2.3	3.2
1.4	4.3	3.5	4.3	3.1
1.5	1.9	3.5	2.6	3.4
1.6	1.4	2.2	1.8	2
2.1	4.7	3.2	4.5	3.1
2.2	4.4	3.3	4	3.1
2.3	4.3	3.3	4.1	3
2.4	4.3	3.2	4	3
2.5	4.3	3.1	4	2.9
2.6	4.1	3.2	3.9	3
3.1	4.2	2.9	3.8	3
3.2	2.7	4.1	3.3	4.1
3.3	2.5	3.1	2.5	3.2
3.4	3.8	3	3.6	2.8

### Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

[https://creativecommons.org/licenses/by/4.0/deed.en\\_US](https://creativecommons.org/licenses/by/4.0/deed.en_US)