Non-Functional Testing In Agile Development

Ms.Shivani Dave[1], Mr.Hitesh Patel[2], Mr.Ashvin Prajapati[3]
[1,2,3] *Hasmukh Goswami College of Engineering, Naroda-Dehgam Road, Vahlelal-Dascroi, Ahmedabad, Gujarat 382330*

Abstract- **Non-functional requirements define the overall qualities or attributes of a system. Although important, they are often neglected for many reasons, such as pressure of time and budget. In agile software development, there is a focus on the feature implementation and delivery of value to the customer and, as such, non-functional aspects of a system should also be of attention. Non-functional requirements testing is challenging due its cross-functional aspects and lack of clarity of their needs by business in the most part of projects. The goal of this paper is to empirically investigate how do agile team members handle non-functional testing in their projects, aiming to identify preliminary factors influencing the testing of non-functional requirements, specifically performance and security in agile development. Also, work with the high-quality perception is crucial in accordance the participants.**

I. INTRODUCTION

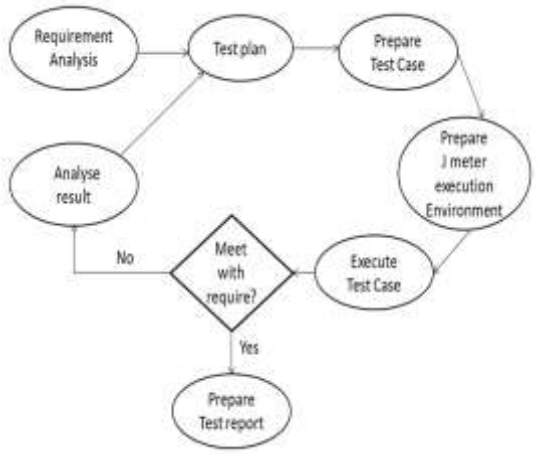
Today, in every field of operation one thing is essential that is computer to run desired applications. Software application development is playing a vital role in current industries including automobile, telecommunication, retail, governance, banking, etc. Software development includes some steps to follow such as gathering requirements, creating application, testing, deployment at the user end and maintaining the application. Every mentioned step is linked, it could create a lot of rework if any critical bug found at maintenance step, so the deployed software product should be properly tested [8]. And with the huge competition between software companies, time of delivery of product plays a critical role. Also the software - testing process is a costly process, and complete testing is practically impossible. Many software organizations are spending up to 40% of their resources on testing [9].Along with that, the testing of non-functional requirements has not been taken seriously [12] and it is very often classified as low-risk due to its characteristics [11][10]. Non-functional testing requires long time of execution and an open minded approach. The necessity of an overall approach and the necessity of a long execution time can be also listed as an additional concern since agile development brings a focus on the feature implementation and faster delivery of value to the customer (generally functional requirements), bringing even more difficulty to identify non-functional aspects [12]. Non-functional testing needs can emerge from different sources, from business, as a customer need [12], as a technical piece of a given requirement, as part of an architectural change [13],or as part of the production behaviour, which in agile would be for example part of the DevOps integration [14].

II. THEORETICAL BACKGROUND

Non-functional requirements (NFRs), also known as technical requirements, quality of service, cross-cutting, or quality attributes focus on aspects that typically involve or cross-cut several functional requirements [15]. Although considered important and crucial to project success [16], it is common to see non-functional requirements losing attention in comparison to functional requirements [17].Crispin and Gregory (2009) [12] argue that with that business partners might assume that the development team will take care of non-functional requirements such as performance, reliability, and security, compromising the final product. In addition, due to the agile philosophy that stimulates delivering user value early and often the prioritization of quality attributes can be hard in early deliverables increments resulting in hard-to-modify, unreliable, slow, or in-secure systems [19] [20] [21]. Testing happens for a lot of reasons. Different types of testing can be exercised to achieve different goals depending of the feature and software process which is being followed. This is not different in Agile software development, where development can even start from the test perspective [22]. Security testing aims to validate the system in

terms of security vulnerabilities such as data protection, confidentiality, integrity, authentication, availability and authorization issues [11]. Security testing is highly dependent on the security requirements imposed by the system line of business, e.g. a financial application. Several types of security test can be applied depending of the system, such as: Vulnerability scan, Vulnerability Assessment, Security Assessment, and Penetration Test [12] [11][24].Another issue is that agile development teams are generally composed by a small number of developers, who also many times act as testers [25]. However, some non-functional testing such as performance requires specialized tools such as profilers and might need specialized knowledge [12]. Given this need for specialized knowledge, a team member with specialized skills might be required for project success or to avoid issues in production [11]. Programmers might not be aware that non-functional testing such as performance and security might be a high priority and key to quality which touch a cultural mind-set change. In addition, due to its nature of involving so many features of a given system, non-functional testing cannot be executed as part of an unit test, and as commonly take time and cannot be executed in a normal-continuous-integration-system cycle [20].

III. PROPOSED WORK



(Figure 1.1 Proposed work)

1. Step 1: Get the information according to testing.
2. Step 2: Make a test plan for the testing according to requirement.
3. Step 3: prepare test cases according to test plan.
4. Step 4: Make J meter platform for the execution.
5. Step 5: Implement the test case.
6. Step 6: Check whether requirement is fulfilled or not.
7. Step 7: If No than analyse the result and execute again.
8. Step 8: If Yes than Prepare the test report.
9. According to the conclusion from literature survey I have prepped this proposed work.

IV. PROBLEM STATEMENT

Non-useful requirements outline the general characteristics or attributes of a gadget. Although important, they may be regularly left out for lots motives, which includes strain of time and finances. In agile software program improvement, there is a focal point on the function implementation and transport of value to the purchaser and, as such, non-useful elements of a system have to additionally be of interest. Non-functional necessities trying out is challenging due its pass-useful elements and absence of clarity of their desires with the aid of commercial enterprise in the maximum a part of tasks. Acceding to the Literature survey to empirically how do agile deal with non useful checkout, specifically performance and safety in agile development. Also, work with the amazing notion is critical in accordance the contributors. Quality on pinnacle of thoughts at some stage in the whole manner and put up-delivery. Participants also stated code assessment exercising as an tool that enables within the identification and execution of non-useful checking out. All roles operating with the extraordinary in mind and attention of non-useful wishes.

V. CONCLUSION

Experience, culture and awareness were emphasized as the most important factors influencing the identification of agile testing needs, execution and resolution of non-functional issues. Non-functional necessities trying out are challenging due its pass-useful elements and absence of clarity of their desires with the aid of commercial enterprise in the maximum a part of tasks.

REFRENCES

[1] Cristina Rosa Camacho, Sabrina Marczak, Daniela S. Cruzes" Agile Team Members Perceptions on Non-Functional Testing" 2016 11th International Conference on Availability, Reliability and Security.

[2] Roaa Elghondakly, Sherin Moussa, Nagwa Badr" Waterfall and Agile Requirements-based Model for Automated Test Cases Generation" 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS'15).

[3] Wentao Wang, Arushi Gupta, Nan Niu" Mining Security Requirements from Common Vulnerabilities and Exposures for Agile Projects" 2018 1st International Workshop on Quality Requirements in Agile Projects.

[4] Gaurav Kumar, Pradeep Kumar Bhatia "Impact of Agile Methodology on Software Development Process ".International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 4, August 2012

[5] Ambler, S. W. "Beyond functional requirements on agile projects",Dr. Dobb\'s Journal, vol. 33–10, 2008, pp. 64–66.

[6] Baca, D.; Boldt, M.; Carlsson, B.; Jacobsson, A. "A novel securityenhancedagile software development process applied in anindustrial setting". In: Availability, Reliability and Security(ARES), 2015 10th International Conference on, 2015, pp. 11–19.

[7] Paetsch, F.; Eberlein, A.; Maurer, F. "Requirements engineeringand agile software development". In: null, 2003, pp. 308.

[8] Pavlovski, C. J.; Zou, J. "Non-functional requirements in businessprocess modeling". In: Proceedings of the fifth Asia-Pacificconference on Conceptual Modelling-Volume 79, 2008, pp. 103–112.

[9] D. J. Mosley and B. A. Posey, Just Enough Software Test Automation,Prentice Hall, 2002

[10] F. Elberzhager, A. Rosbach, J. Münch and R. Eschbach, "Reducing test effort: A systematic mapping study on existing approaches" Information and Software Technology, vol. 54, issue 10 54, pp. 1092– 1106, October 2012.

[11] Paetsch, F.; Eberlein, A.; Maurer, F. "Requirements engineering and agile software evelopment". In: null, 2003, pp. 308.

[12] Crispin, L.; Gregory, J. "More Agile Testing: Learning Journeys for the Whole Team".Pearson Education, 2014.

[13] Singh, P.; Tripathi, A. K. "Treating nfr as first grade for its testability", 2012

[14] Bass, L.; Nord, R.; Wood, W.; Zubrow, D. "Risk themes discovered through architecture evaluations". In: Software Architecture, 2007. WICSA'07. The Working IEEE/IFIP Conference on, 2007, pp. 1–1.

[15] Httermann, M. "DevOps for developers". Apress, 2012.

[16] Ambler, S. W. "Beyond functional requirements on agile projects", Dr. Dobb\'s Journal, vol. 33–10, 2008, pp. 64–66.

[17] Martens, N. "The impact of non-functional requirements on project success", 2011.

[18] Lawrence, B.; Wiegers, K.; Ebert, C. "The top risk of requirements engineering", Software, IEEE, vol. 18–6, 2001, pp.62–63.

[19] Crispin, L.; Gregory, J. "Agile testing: A practical guide for testers and agile teams". Pearson Education, 2009.

[20] Baca, D.; Boldt, M.; Carlsson, B.; Jacobsson, A. "A novel securityenhanced agile software development process applied in an industrial setting". In: Availability, Reliability and Security (ARES), 2015 10th International Conference on, 2015, pp. 11–19.

[21] Bellomo, S.; Ernst, N.; Nord, R. L.; Ozkaya, I. "Evolutionary improvements of cross- cutting concerns: Performance in practice". In: Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference on, 2014, pp. 545–548.

[22] Wäyrynen, J.; Bodén, M.; Boström, G. "Security engineering and extreme programming: An impossible marriage?"in: Extreme programming and agile methods-XP/Agile Universe 2004, Springer, 2004, pp. 117–128.

[23] Yahya, N.; Bakar, A. A.; Sham, N. "Test driven development contribution in universities in producing quality software: A systematic review". In: Information and Communication Technology for The Muslim World (ICT4M), 2014 The 5th International Conference on,2014, pp. 1–6.

[24] Glinz, M. "On non-functional requirements". In: Requirements Engineering Conference, 2007. RE'07. 15th IEEE International,2007, pp. 21–26.
[25] Hibshi, H.; Slavin, R.; Niu, J.; Breaux, T. D. "Rethinking security requirements in re research", Technical Report, Tech. Rep. Report CSTR-2014-001, Univ. Texas at San Antonio, 2014.
[26] Sommerville, I. "Software Engineering, Ninth Edition". Boston,MA: Pearson, 2010, 792p.