

Think Software? Think Testing? Think Design!

Sushmitha S Prabhu^{1*}, Zoya Kapoor²

Corresponding Author: sushmithaprabhu13@gmail.com

DOI: <https://doi.org/10.26438/ijcse/v7i5.16111616> | Available online at: www.ijcseonline.org

Accepted: 21/May/2019, Published: 31/May/2019

Abstract— Software testing in Software Development Life Cycle (SDLC) can be thought of as a sub process as Software Testing Life Cycle which is the one of the testing methodologies that is followed in the agile developmental model. Understanding quality as a process in the cycle of software development, it is also necessary to understand the importance of quality since quality wins customer. So, to better the process of quality it becomes necessary to think out of the box on how to deal with the quality of the product. Though big players in the software industry have been investing ideas on better test methodologies to raise the quality standard of the product there's always been a gap between the ideal bug free software product and the realistic one. There are plenty of reasons as for why such a gap is witnessed, some of them are lack of right process model, lack of experienced resources, lack of knowledge on tools and so on. On the other hand, Design Thinking (DT) viz., the key for creative resolution of problems has evolved over time that can be sought as a path for solving problems of varsity, and problems that are less understood to be dealt with. So, why can't DT be a part in the process of quality of a product? In this paper, I propose an approach of the software testing methodology that involves the synergic effect of design thinking strategy on software testing, trying to bridge the gap between the idealistic and realistic view of a software product, discuss many aspects of how design can be a part of quality and conclude on heading to a new path of creative testing.

Keywords— *Agile process, Design thinking (DT)*

I. INTRODUCTION

In this constantly growing agile software world, there is a lot of lack of a clear picture on what are the problems, what are the ones that need to be solved and how. In such a case, any business idea would click into the market only if it can solve the problem as expected and as quickly as possible since an early bird always wins the world. For this to be a success, it is necessary that every little piece of code or design that is delivered is of high-standard. But, the usual test process that is followed in any agile based development has not been able to do anything commendable to meet this principle and thus still facing a setback on achieving the desired quality in the product. Hence, this write up attempts to build a foundation to achieve what is known as an idealistic quality in any software product. This paper can act as a source of reference to establish and implement better methodologies described here for use in practice. The objective of this paper is to imbibe guided process into Software Testing by involving design thinking to create a synergic effect that leads to good quality software.

The rest of the paper is organized as follows, Section I contains the introduction of involving best software testing practices, Section II contains the keywords description, Section III contains the design overview, Section IV contains the assumptions made, Section V explains the current software testing practices used and its problems. Section VI describes the proposed approach at solving the problem, Section VII contains the implementation of the solution that is discussed in the proposed approach section, Section VIII concludes the research work with future directions and Section IX describes the various abbreviations used throughout this paper.

II. KEYWORDS ELABORATED

A. Agile Process

An agile process is something that involves software development and test methodologies, where the product for the proposed requirement is built by self-organizing cross-functional teams arriving at a high-standard solution. [1]

B. Design Thinking



Fig 1. The 5 Stages of Design Thinking [3]

Design thinking refers to creative strategies' designers use during the process of designing. It has also been developed as an approach to resolve issues outside of professional design practice, such as in business and social contexts. [2] The creative solution that design thinking offers provides a better way to resolve problems at hand in a different way. A design thinking process involves empathizing, defining, ideating, prototyping and testing the design solution.

- a. Empathize: To empathize is to understand the human centric views as in what the current experiences are, opinions, pain points, challenges, desires and so on. This can be done by engaging, observing and discussing a lot of questions and expectations that crop up in the customer's mind.
- b. Define: In the define stage, you refine the information collected in the empathize phase into meaningful problem statements or concise insights into problems that exist in the area you're designing for.
- c. Ideate: This stage involves a lot of discussions on presenting ideas and a lot of brainstorming sessions to get into multi-level ideology for the problem statement. This stage encourages and gives opportunity to an ocean of ideas.
- d. Prototype: This phase caters to the first level of solution where in prototypes of the considered idea in the previous phase can be brought into an understandable implemented solution in the form of sketches, a basic working model, or an activity and so on.
- e. Test: In this stage, the prototypes are tested as if this is the solution that is developed, giving way to a lot of refinement in the design and trigger more empathy with the users.

III. DESIGN OVERVIEW

Design thinking is an iterative process where in practically better solutions are visualized and implemented that might not have crossed minds in the first step of problem-solving approach. It basically focuses on empathizing the behavioral pattern of the user. The entire solution is developed targeting the user in mind. Design thinking is an approach where in every part of the problem at hand is questioned, every view is considered, every single assumption is questioned, and the design implications are realized. Problems that have no defined solutions or unknown path to solutions can be better analyzed and implemented in the human understandable way. This is done by pouring in many ideas on the plate with a lot of brainstorming sessions, performing hands-on, developing prototypes and testing them. It is never an end solution or a fully complete approach to problems, it is an experiment which can evolve by trying out a lot of concept work and building in a lot of ideas.

So how can we relate DT in the vast world of software testing? In the software development cycle, testing is involved at the end of any workable set of modules, or these days at the beginning of the product cycle and DT is involved in the initial stages where one solution is realized, this tells that DT is only used as a single time approach for designing solutions which is only at the initial stages. When design thinking as the capability of bringing in better solutions it shall become necessary to involve it in every stage of the SDLC process. In an order to achieve this, involving design thinking in the testing process can make the product better assuring better quality solution with less glitches or rework.

In a way to answer why we need design thinking in software testing, we can realize few of the necessities of it by reading further. Firstly, a quality product can give us huge customer wins, which means to say that the lower the quality, the smaller the number of customers, or even loss of customers since buggy products do not tend to last long in the marketplace. It is important to understand that in this fast-growing world where technology and social media has a huge control on minds of people, every issue that gets online has a ripple effect. Hence, the goal should be to prevent problems before hand and do not let such a ripple effect hit the product. Second, when there is low quality, that means efforts need to be spent again on bettering the solution which might delay the planned launch of the software product which in turn could mean a late bird into the market losing on customers. It is always understood that cost of fixing is more than cost of development. Third, quality is not only seen as an external aspect, but it is also an important internal aspect where poor quality can make think the stakeholders to look for someone else to do the job. Stakeholders experience is as critical as the raw number of defects when the software is being developed, tested and implemented. Fourth, poor quality can mean efforts are still needed to optimize or better the solution experience, which can be understood as there would be less time to think of next set of features that are already on the minds of the customers.

IV. ASSUMPTIONS

In applying design thinking into software testing, below are few of the assumptions in more detail based on the key observations made earlier.

- There is no system with 0 defect achievement. If such a system exists, it would only be an ideal system which is less practical to realize.
- Huge development means lot of quality issues. This means it is valid that half the effort of development cycle is needed into the testing process.
- It is difficult to say when any testing is complete.
- Any design that is involved into the process of quality might always not cater to the problem 100%
- The design thinking that is involved into software testing is an evolving process and is an iterative process which is based on the human views.
- When speaking of automation, it is not possible to achieve 100% automation on all the features or code-lines, since it is not possible to automate all scenarios.
- Also, it should be better understood that the customers' requirements might not be a finite set always, so it depends on the product team to know what is to be prioritized and put into the market quickly
- It is also difficult to sometimes think or use a product like the customer does since every individuals view is different.

V. CURRENT PROBLEMS

To understand the solution that is based out of this paper, let us first understand the current problems or setbacks that are present in the software testing process that is followed across any product development:

- Absence of right process**
Many companies fail to set a right process in testing, which means many bugs might not be reported on time. Hence, it is necessary to clearly define process to avoid misunderstandings. It also becomes difficult to set a uniform process across locations of the same company.
- Gap in communication**
It is evident that improper communication might lead to wrong testing or even wrong defects might be reported which in turn consumes time for the team to understand what changes went through in the requirements. Hence, it is necessary that the QE team be involved in every discussion that focuses on huge changes in the requirement either by the Dev team or by the product management team. The baseline is that the QE must be informed of any immediate or sudden changes so that the quality of testing is not hampered.
- It is unrealistic to cover all test combinations**
As the point says, it is impractical to test all possible combinations or scenarios for any application. Instead of testing all the scenarios it is better that those business goals and features which need clear definition targeting the end user be prioritized and tested.
- Lack of SPOC**
Sometimes, absence of SPOC (Single Point of Contact) to discuss about certain kind of defects which might be of considerable significance might lead to problems in delivering high quality result by testers and the same defect once witnessed might crop up from the end users point which is then to be tagged as customer issue leading to hot fixes and losing customer retention.



Fig 2. Test document multi-level review structure

The above is few of the many problems that are seen in testing in the current trend of product development, few other problems might be lack of proper testing tools, lack of knowledge with respect to the severity and priority of defects, different ways of bug tracking leading to inconsistency, less control on test environment etc.,

VI. PROPOSED APPROACH

As you read in this paper few of the problems faced in the testing let me take you through one of the thought-through ways of reducing or helping the less clear process toward achieving a better-quality process by involving what we know is the **Design Thinking** methodology.

As the name speaks, design thinking is to think design to arrive at a solution basically not failing to empathize with the user. Let us consider an example of how any usual testing would go on in any application/product development: The requirement set comes up from the customer, stakeholders → the requirement is reviewed, prioritized, picked up as deliverables of that release → the development team then starts with the development of the requirements by following a lot of design, architecture discussions and reviews → meanwhile, the test team might be held up with understanding the requirements before it is actually available for testing and would prepare the test data and environment for testing → the requirements are delivered into the Q environment and the testers test them reporting every possible issues found with the feature → defects go to the dev plate and are analyzed for the root cause and fixes are provided post which retest will happen.

As we can see, the QE is only involved in reading the finalized requirements of that release, then preparing a test strategy document, testing the feature and retesting in case of issues found and fixed. This clearly shows the gap of communication of the QE being able to understand or empathize with the customer. How can the design thinking attribute of empathy be achieved in the test process? This is where I propose my solution to think design in testing, by refining toward a better standard of testing.

Let us understand how to empathize with the customer as part of the design thinking activity in testing. Consider the same example of product development: The requirement set comes up from the customer, stakeholders → ... → the development team then starts with the development of the requirements by following a lot of design, architecture discussions and reviews → **meanwhile, the test team might be held up with understanding the requirements before it is actually available for testing and would prepare the test data and environment for testing** → ... → defects go to the dev plate and are analyzed for the root cause and fixes are provided of which retest will happen. Here's what can be implemented as the *meanwhile* part of the above test process,

Empathize-

1. Firstly, once the architecture document or the product requirement document is ready and read across by the QE, it is necessary that QE should question or clarify the requirement with the Product Management to better understand and to start documenting a good set of test designs/cases.

Define-

2. Once necessary efforts are spent on creating a test design document, it is very important that this test document be sent across multiple level of reviews which aids to filling the gap with respect to the user point of view. So, what are these multiple levels of reviews that a test document must go through:
 - a. Dev Review: In the dev review process, it is possible that missing scenarios be given as feedback to the QE or even additional cases may be guided to the QE.
 - b. QE Lead Review: In the QE Lead review process, it is possible that refinement of test cases may be made, or the test cases may be broken down to certain level of granularity.
 - c. Product Management (PM) Review: The product manager shall review the test document and provide clarity if in case of misunderstood requirements.

Here, we can clearly see how the attribute of design thinking viz., empathy is closely related, since the PM is also one of the stakeholders of the product, and is an individual or a team who is in close association with the customer, it becomes easy that the PMs have a clear picture of what the customer wants and how good is the current test strategy affecting the usage points of the customer thus, helping the test team gain ample insights into high standard testing.

In all the three levels of review the individuals can provide their feedback as form of comments to respective areas that need reconstruction, also, this review process can become regular activity of the product development phase where in these activities can be recorded as tasks.

This also helps in making the development team to understand the test process and scenarios.

Ideate-

3. One of the better ways to get the quick benefit of fixes is that providing the developer with the test scenario along with the reported defect to be tested after the fix has been made in the local dev machine. This makes sure that the fix is correct, and the test team can utilize remaining effort on finding new issues rather than spending too much time retesting the issue.

4. It becomes necessary to set a definition of done be it only testing. A testing done criteria is necessary though it is not possible to say when to stop testing. May be manual testing of the listed scenarios along with automation completeness can act as a pair criterion to say testing is complete.

Prototype-

5. To understand how a customer might use an “asked” feature; as a next step, it is good to directly set the system at the hands of the stakeholders. In this case, provide the test environment with all data set up and configurations to the PM. A PM who is in a way the face of the customer may try out the delivered requirements and provide feedback or may be even report new issues those which were missed by the hands of the tester.
6. As part of the above point, it is good to have significant customer cloned environments so that any new changes can be tested as part of the customer clone to avoid customer defects. For example, there are customers who customize the product as per their needs and these may be led to a variety of a new set of defects that cannot be detected as part of common test environment set up. This is one of the ways to closely question, understand and resolve the various “Why’s” (needs) of a customer.

Test-

7. One of the possible ways to avoid chances of bug misses is that it is good if there’s a couple of days of deployment activity be held where in participants such as the Dev team, the QE team, and the PM team be involved and do a single round of testing on a simulated customer environment set up on pre-deployment of the release and post-deployment of the release to make sure last-minute surprises are avoided so that even in case of surprise bugs, it can be fixed, and deployed into production week ensuring quality release is going into the customers bag.
8. Coming to the other minor aspects on how to change the design in testing, it is good if every user story be associated with the test plan number or be attached with the respective test document so that it will be easy if any new QE member gets his/her hands on that story.

VII. IMPLEMENTATION

Coming over to the implementation part, as discussed in the proposed solution, the various stages of review can be done via a tool like Jira, which is well-known for issue tracking and project management. Since, this tool is widely used across industries for project management, it would be easy to conduct reviews of test cases via the tool which gives a clear-open view to the management as well and which provides a seamless way to manage timely feedbacks.

VIII. CONCLUSION AND FUTURE SCOPE

As technology keeps updating, the customer number keeps increasing, thus, newer agile process models are coming up to satisfy such trending needs. This paper presented a different software testing process model on how to include empathy into the testing cycle of SDLC by involving the design thinking strategy that aims to achieve high standard product delivery. The future scope aims at how the similar strategy of Design Thinking be involved into what is known as automation testing so that software testing via automation shall become more effective in delivering promising results that can be trusted upon.

IX. ABBREVIATIONS

- SDLC - Software Development Life Cycle
- DT - Design Thinking
- SPOC - Single Point of Contact
- PM - Product Manager / Management
- Q - Quality
- QE - Quality Expert

REFERENCES

- [1] Cprime, *What is Agile? What is Scrum?*, [Online], Available: <https://www.cprime.com/resources/what-is-agile-what-is-scrum/>
- [2] Wikipedia, *Design Thinking*, [Online], Available: https://en.wikipedia.org/wiki/Design_thinking
- [3] Reply, *Design Thinking*, [Online], Available: <http://www.reply.com/en/design-thinking>
- [4] David Terrar, Enterprise Irregulars, “*What is Design thinking*”, [Online], Available: <https://www.enterpriseirregulars.com/125085/what-is-designthinking/>
- [5] Aditya P Mathur, *Foundations of Software Testing*, 2nd Edition, Pearson, 2014

- [6] Don Norman, The Design of Everyday things, “*Design Thinking – The Double Diamond model for design*”, Published by Basic Books, Perseus Books Group, 2013
- [7] Roger S. Pressman, *Software Engineering: A Practitioner’s Approach*, 7th Ed., McGraw Hill International Edition, 2017
- [8] Ian Sommerville, *Software Engineering*, 9th ed., Pearson, 2014
- [9] Paul C Jorgensen, “*Software Testing: A Craftsman’s Approach*”, CRC Press, 2013, 4th Edition.

Authors Profile

Ms. Sushmitha S Prabhu, has 3+ years of IT Industry experience and currently employed as a developer at SAP Labs India Pvt. Ltd. Her areas of specialization are on front-end technologies such as HTML, CSS, JavaScript and backend topics such as Java. On a lighter note, she loves playing badminton



Ms. Zoya Kapoor has 9 years of IT Industry experience and is a Certified Life Coach, HR Consultant and a BITS Guest Faculty. She is a passionate and optimistic person who believes in optimizing the best of her abilities. She believes in performing best to her abilities also, develop all her skills, achieving excellent results in the work and the assigned area through the trust placed in her with an invaluable experience to get an extremely important step in her personal growth together with the immense pride.

