RIGA TECHNICAL UNIVERSITY

Padmaraj Nidagundi

SOFTWARE TESTING STRATEGY UTILIZING LEAN CANVAS MODEL

Summary of the Doctoral Thesis

RTU Press
Riga 2022

RIGA TECHNICAL UNIVERSITY

Faculty of Computer Science and Information Technology
Institute of Applied Computer Systems

Padmaraj Nidagundi

Doctoral Student of the Study Programme "Computer Systems"

SOFTWARE TESTING STRATEGY UTILIZING LEAN CANVAS MODEL

Summary of the Doctoral Thesis

Scientific supervisor
Professor Dr. sc. ing.
MARINA UHANOVA
Professor Dr. habil. sc. ing.
LEONĪDS NOVICKIS

RTU Press
Riga 2022

Nidagundi, P. Software Testing Strategy Utilizing Lean Canvas Model. Summary of the Doctoral Thesis. – Riga: RTU Press, 2022. – 30 p.

Published in accordance with the decision of the Promotion Council "RTU P-07" of August 25, 2021, No. 21-7.

https://doi.org/10.7250/9789934227325
ISBN 978-9934-22-732-5 (pdf)

DOCTORAL THESIS PROPOSED TO RIGA TECHNICAL UNIVERSITY FOR THE PROMOTION TO THE SCIENTIFIC DEGREE OF DOCTOR OF SCIENCE

To be granted the scientific degree of Doctor of Science (Ph. D.), the present Doctoral Thesis has been submitted for the defence at the open meeting of RTU Promotion Council on 21 March 2022 at 14:30 at online https://rtucloud1.zoom.us/j/93987854650 Riga Technical University.

OFFICIAL REVIEWERS

Professor, Dr. habil. sc. ing. Jānis Grabis
Riga Technical University

Dr. sc. ing Igor Lemberski
Ventspils University, Latvia

Assoc. Professor Dr. Dejan Jokic
International BURCH University, Bosnia & Herzegovina

DECLARATION OF ACADEMIC INTEGRITY

I hereby declare that the Doctoral Thesis submitted for the review to Riga Technical University for the promotion to the scientific degree of Doctor of Science (Ph. D.) is my own. I confirm that this Doctoral Thesis had not been submitted to any other university for the promotion to a scientific degree.

Padmaraj Nidagundi _____ (signature)

Date: _____

The Doctoral Thesis has been written in English. It consists of introduction, 4 chapters, conclusions, 45 figures, 23 tables, 10 appendices: the total number of pages is 175.

TABLE OF CONTENTS

1. GENERAL DESCRIPTION OF THE RESEARCH

Topicality of the Theme

Rapidly developing technology has brought many benefits to human life; technology is increasingly integrated within software today. Technology has now become an integral tool in addressing daily issues. Software development has moved towards the agile approach, as outlined by Black in [1], and whenever software is developed, it needs to be tested and made free from defects before transferring it to end-users. Well-tested software can ensure customer satisfaction. Software testing has a long history and many authors have contributed to this topic, as discussed in [2], [3], [4] and [5]. A test strategy is a framework that describes the software testing technique in the software development life cycle. However, in the field of software testing, the use of Lean Canvas in software testing process and visualization of the test strategy at different levels have not been sufficiently investigated, see [17] and [18].

According to World Quality Reports [20] and [21] published annually, there is a persistent skill gap in test strategy design in the software testing industry. The benefits of the test strategy design include development of testing processes in order to speed up circulation of documentation, to address test management issues and to reduce costs and time of software development. Software development companies continue considering the test strategy design practice as a separate process. The test strategy design practice is not always seen as too important and is often considered only as an aspect increasing operating costs. A clear and adequate process or test strategy is adopted according to the research needs, as discussed in [7]. The present research intends to propose the Lean Canvas visualized design tool using test strategy visualization that can serve these needs.

The object of this research is visualization of the software test strategy. Adopting Lean Canvas in strategy visualization process allows reducing the cost of the testing process, speeding up documentation circulation and addressing test communication and management issues that arise during the software product development process. It contributes to the reduction of costs in software development and optimisation of the testing process, as outlined in [19].

The research is valid because it proposes a new visualized test strategy for software development, which may be quickly adopted by a test manager, test team leader, tester, team member or company. In this manner, a visible test strategy is quickly created and managed, monitoring software development life cycles. If software testing is not done appropriately, the customer receives low-quality software, which leads to customer dissatisfaction with the end product. In some cases, organisations incur significant financial losses due to errors in the software.

According to [17], the software development industry requires an appropriate test strategy for the projects according to the scope and requirements of the project. Choosing the proper test strategy is a dominant factor in software test effort according to software test document IEEE 29119. In most cases, a software test strategy plan is a subsection of a test plan.

The research aims to develop a software test strategy design in terms of the development of a new visualized strategy. It is based on the components of applications in the developed software or in the process of development focusing on the functionality of the product.

The proposed strategy should primarily focus on visualization in the overall project. It also identifies the types of testing essential software components, which are the most significant advantages and the highest risk, respectively, thus marking a difference from the traditional testing strategies. The adoption of the test strategy design process provides a significant benefit to the software development life cycle. According to [20] and [21], software development companies continue viewing the 'test strategy design' practice as an inessential part of the process, and in order to address this problem, it is necessary to bridge the skills gap. In the investigation process, the author found that in some companies the strategy design process is long documented, not updated according to the changing requirements, or

is not adopted at all. The author of the research intends to propose the visualized software test strategy as a tool that can serve this type of company and promote the use of proper software testing practices.

Object and Subject of the Research

The object of the research is visualization of the software test strategy.

The subject of the research is the development of the visualized software test strategy within different software development life cycles and types of software testing.

Research Hypothesis

Software test strategy involves such stages as plan, design, execution, analysis and reporting within the software development life cycle (SDLC).

Thesis statements to be defended:

- A visualized software test strategy improves the test process, work visibility, test team resource handling and remote team communication in both functional and non-functional testing.
- The proposed approach will help reduce the lengthy documentation effort and create an effective agile test strategy in a short time in a more visible way.

The hypothesis to be proven within the Doctoral Thesis is as follows: To overcome the above-mentioned challenges, the visualized test strategy has been developed, adopted and evaluated.

Aim of the Doctoral Thesis

The aim of the Doctoral Thesis is to develop a visualized software testing strategy utilizing Lean Canvas model. Software quality improvement is still a significant challenge in different types of software development life cycles (SDLCs) for many organisations, and it gets more critical primarily when distributed teams work remotely and test complex products, according to [30]. The software testing process involves many activities, including different levels of testing and different resources. The author intends to explore and understand the possibilities for software testers to develop the visualized Lean Canvas approach and to examine its effect on the process of communication, collaboration, and circulation of documentation.

Tasks of the Doctoral Thesis

In order to reach the primary aim of the Thesis, the following tasks have been set:

- to explore the existing software testing processes and test strategies and to analyse the need for visualization in strategy creation;
- to identify the main advantages and disadvantages of the recent software strategies and to analyse the strategy of visualization;
- to develop a visualized test strategy based on different real-life projects and software requirements;
- to develop the criteria for evaluation of the proposed visualized test strategy;
- to implement the visualized software test strategy in software development projects and to collect feedback;
- to investigate and document the benefits and limitations based on the practical results, to collect recommendations from other project and team members;
- to define the scope of possible further development of the proposed strategy.

Research Methodology

The following methods have been used in the Doctoral Thesis.

This research starts with the analysis of the existing different software testing strategies and their documentation. Qualitative research method (exploratory and observation) is used to find the strengths and weaknesses of different ways of visualizing the software testing strategy. Six different design options for strategy visualization were analysed, and limitations were identified, which are listed in Section 2.1. Next, this research developed the software testing strategy visualization using the Lean Canvas testing activity building blocks in Section 3.2.

Requirement analysis was done to collect the system design, technology, system users, use cases to study and develop prototype system. In this research, the author collected the system design requirements, technology, system users, use cases to study and develop prototype system. From the collected requirements, the web portal was simulated in Section 3.3.

The quantitative and qualitative methods (survey-based questionnaires, online user reviews, use case method) are used to determine the strengths and limitations of the author's proposed approach to visualizing the testing strategy. Quantitative research methods are used to analyses the length of documentation.

Finally, the analysis of the results of the case study resulted in the recording of the amount of documentation (up to) in pages after the implementation of TABB and the amount of documentation (up to) in each software version in hours.

Many of the processes of theoretical calculations and graphical representation of the results have been obtained utilising a menagerie of software systems, including:

- Wordpress.com (PHP framework for developing a content management system);
- MySQL (storage of user information and content management system);
- Github.com (version control for source code maintenance);
- Atlassian.com (software development and collaboration tools for issue and project tracking);
- Google drawings (TABB creation, storage, and updating of building block templates for testing activities);
- Google forms (survey and data collection);
- Microsoft word (analysis of existing documents and creation of new documents);

Scientific Novelty

Scientific novelty of the Doctoral Thesis:

- A new approach to test strategy visualization has been developed for implementation of software test strategy with the Lean Canvas.
- A new approach – test activity building block (TABB) has been used for finding the right test metrics for the Lean Canvas visualized board using transformation models defined.
- The new strategy has been evaluated within different projects, and data have been collected.

Practical Significance

The author intends to implement the TABB strategy in projects and collect constructive feedback in order to understand the research impact. The author has collected criticism and statistical data for the improvement of TABB strategy.

- Criteria for the TABB strategy have been developed. It has been explained how to identify the lean test metrics for the Lean Canvas board.
- Experiments have been conducted with the aim to utilize the Lean Canvas test strategy in software development projects in software development companies.
- The collected practical results can be used in future by the software companies to improve software development and testing processes, mainly the test activity building block (TABB) strategy

on Lean Canvas.
- An open-source project has been created, where the integrated strategy models can be used.

Approbation of Research Results

The research results have been presented at 8 international conferences in the Czech Republic, Ukraine, Lithuania, Austria, and Latvia.

1. Symposium for Young Scientists in Technology, Engineering and Mathematics, Kaunas, Lithuania, 28 April 2017.
2. Environment. Technology. Resources. International Scientific and Practical Conference. Rezekne Academy of Technologies, Rezekne, Faculty of Engineering, Latvia, 29 April 2017.
3. IRES International Conference, Vienna, Austria, 26 November 2017.
4. 12th International Scientific and Technical Conference "Computer Science and Information Technologies". Lviv, Ukraine, 6 September 2017.
5. CSOC 2017, Faculty of Applied Informatics, Tomas Bata University in Zlin, Czech Republic, 27 June 2017.
6. RTU 58th International Scientific Conference, Riga Technical University, Riga, Latvia, 13 October 2017.
7. International Conference on Information Technologies, IVUS 2018, Kaunas, Lithuania, 27 April 2018.
8. RTU 59th International Scientific Conference, Riga Technical University, Riga, Latvia, 11 October 2018.

The results of the research have been published in the following publications:

1. Nidagundi P., Introduction to Investigation and Utilizing Lean Test Metrics. In: Agile Software Testing Methodologies. Int. Journal of Engineering Research and Applications. ISSN: 2248-9622, Vol. 6, Iss. 4, (Part 1) April 2016, pp.13-16. (INDEXED: Google Scholar).
2. Nidagundi P., Novickis L. Possibilities about the design lean canvas model and its adaptation in the agile testing. Symposium for Young Scientists in Technology, Engineering and Mathematics, Vol. 1853, SYSTEM 2017, Kaunas, Lithuania, 28 April, pp. 20-23. (INDEXED: SCOPUS, Google Scholar).
3. Nidagundi P., Novickis L. Introduction to Lean Canvas Transformation Models and Metrics. In: Software Testing, APPLIED COMPUTER SYSTEMS, 2016, pp. 30-36. (INDEXED: Google Scholar, Web of Science).
4. Nidagundi P., Lukjanska M. Introduction to adoption of lean canvas in software test architecture design. Computational Methods in Social Sciences, Vol. 4, No. 2, 2017, pp. 23-31(9). (INDEXED: Google Scholar, ProQuest, RePEc, DOAJ, EconPapers, Index Copernicus International).
5. Nidagundi P., Novickis L. Introducing Lean Canvas Model Adaptation in the Scrum Software Testing. Procedia Computer Science. Riga, Latvia. Vol. 104, 2017, pp. 97-103. (INDEXED: SCOPUS, Web of Science, ScienceDirect, Google Scholar).
6. Nidagundi P., Novickis L. Towards Utilization of Lean Canvas in the DevOps Software. Environment. Technology. Resources. Proceedings of the International Scientific and Practical Conference, ISSN 2256-070X, June 2017, Riga, Latvia. pp. 107-111. (INDEXED: SCOPUS, Google Scholar).
7. Nidagundi P., Novickis L. Towards Utilization of a Lean Canvas in the Testing Extra-Functional Properties, Software Engineering Trends and Techniques in Intelligent Systems. Czech Republic, Zlin, June 2017, pp. 349-354. (INDEXED: SCOPUS, Web of Science, Google Scholar).
8. Nidagundi P., Stepanova V. Survey on Software Test Strategy. Proceedings of the IRES International Conference, Austria, Vienna, November 2017, pp. 26-27. (INDEXED: Google Scholar, WORLD RESEARCH LIBRARY).

9. Nidagundi P., Novickis L. New method for mobile application testing using lean canvas to improving the test strategy. 12th International Scientific and Technical Conference "Computer Science and Information Technologies". Lviv, Ukraine. Sept. 2017. (INDEXED: IEEE Xplore digital library, SCOPUS, Web of Science, Google Scholar).
10. Nidagundi P., Novickis L. Towards Utilization of a Lean Canvas in the Biometric Software Testing. A JOURNAL OF MULTIDISCIPLINARY SCIENCE AND TECHNOLOGY, IIOABJ, 2017, Vol. 8, pp. 32–36.
11. Nidagundi P., Uhanova M. Software application security test strategy with lean canvas design. International Conference on Information Technologies. Lietuva, Kaunas. April 2018, pp. 50–53. (INDEXED: SCOPUS, Google Scholar).

Outline of the Doctoral Thesis

The Doctoral Thesis consists of introduction, four chapters addressing theoretical and practical issues, conclusions, bibliography, and appendices. The volume of the Doctoral Thesis is 175 pages, it contains 43 figures and 24 tables, there are 199 titles in the bibliography.

The Summary of the Doctoral Thesis consists of introduction, five chapters addressing theoretical and practical issues, conclusions, bibliography, and appendices.

Chapter 1 provides a general description of the research, i.e., states the problem to be solved, defines the goal, objectives and thesis statements to be defended, as well as presents the

research methodology and structure of the Doctoral Thesis

Chapter 2 is devoted to the main concepts, strategy types and standards.

Chapter 3 includes the visualized software test strategy design, strategy visualization and introduction to Lean Canvas design.

Chapter 4 presents the development and visualized test strategy generation. TABB (Test activities building block) is introduced, explanation and system design use case are developed. The prototype system design overview, adoption of a new strategy considering the importance of system design and its limitations is presented.

Chapter 5 evaluates approbation and test activity building block strategy case studies, presents experiments overview and analysis of case study results.

Conclusion and Scope of Future Research summarizes the research results, the obtained findings and directions for future research.

2. MAIN CONCEPTS

Definition of the test strategy is as follows: A test strategy is a framework that describes the software testing technique in the software development life cycle.

It is designed to inform project managers, team members, testers, and developers about some essential issues of the testing process. In software testing process, the software test strategy is adopted either proactively or reactively [28].

a. Proactive strategy adaptation: In this process, the test strategy created in the early stage to discover and fix software defects before the software structure is created.

b. Reactive strategy adaptation: A process software testing strategy is not created until user interface design and coding are completed.

Defining the strategy approach and quality assurance objective ensures a seamless communication between stakeholders, quality assurance specialist, and developers in order to create a good quality software product. Having a good understanding of the strategy is the first step to deciding on the strategy. At the initial stages of software engineering, good documentation creation is the most important basic practice to be implemented during software development.

Considering the documentation for software test strategy, in the initial stage it looks small, and in the later stages, it needs to be changed more frequently according to the changing requirements and scope of the project. Moreover, a tester starts testing and exploring and thinks about the discovery of more information over time. Respectively, the test strategy grows in volume. Over time it becomes very difficult to manage such a large document.

According to [29] and [30], the current industry trend is agile process manifesto, which focuses on building the working software over comprehensive documentation. In the author's point of view, it is necessary to think about the possibilities of visualizing the software test strategy itself. The visualized strategy is supposed to highlight the risks in testing and possible components of the test strategy document, as well as to increase the impact of team communication on reducing the testing costs and documentation effort.

Test strategy means a specific technique that a software tester is going to follow when considering an application for testing. The author analyses the existing software test strategies. The author created Table 2.1, which presents the strategy types, risk factors, business goals, necessary skills of the testing team, objectives, regulations, and visualized test strategy.

Table 2.1

Strategy Types and Standards

Strategy type	Risk factors	Type of testing	Skills required from the testing team	Scope and objectives	Regulations
Analytical	Yes	-	-	-	-
Model-based	-	Yes	-	-	-
Methodical strategy	-	-	-	-	Yes
Process- or standard-compliant strategies	-	-	-	-	Yes
Dynamic strategy	Yes	-	-	-	-
Consultative or directed strategy	No	-	-	-	-
Regression-averse	Yes	-	-	-	-
Reactive test strategy	No	-	-	-	-
Standard-compliant test strategy	-	-	-	-	Yes

The author of the Thesis suggests approaching the problem-solving in a visualized way. The author addresses such problems as the most significant risks, what the product owner wants to know and what goal is to be reached [31]. In this way, the strategy offers a solution to a complex problem of how to meet the information needs of the stakeholders and how to do that in an efficient way. There are many aspects to be considered on the way to a better software test strategy.

3. ASPECTS OF TEST STRATEGY VISUALIZATION

The context of the research problem is as follows: software testing becomes very important as well as challenging for any software development company in an attempt to deliver error-free software to the end customers. The traditional strategy in testing software applications was adopted from long-established companies. According to [6], [7], [8] and [9], many research articles and industry surveys prove that software testing needs to be improved and that it is necessary to rethink the testing strategy, test management, tools and process adaptation.

3.1 Software Test Strategy Visualization

According to [23], [24] and [25], software development projects have a prompt access to project relevant information, which means that they have the power to improve the programming and testing speed, gain a better understanding of the business domain, seizing opportunities and gaining a competitive advantage in the marketplace. But until that point, information comes in the form of documentation, which, alone and without proper analysis, may not mean much. In practice, depending on the amount of documentation, the interpretation itself can become quite time-consuming or provide an extremely complicated result.

In order to facilitate comprehension, one can focus on business data, process and documentation. In the process of ethical decision making, visualization is essential for any company that wants to get excellent results. Visualization techniques consist in trying to accurately imagine the content of the text by forming mental images that may be enriched by the evocation of sounds and other sensations. Visualization strategies based on drawing the content of the text or representing it can also be considered.

The need for strategy visualization is straightforward: a good diagram can communicate information immediately in a memorable way. Visualization strategies improve understanding, help capture significant moments in the text and combine as many tangible details as feasible. Visualizing a diagram, users are invited to express their sentiments, visualise the text and make it their own, communicate decisions in an effective way. As a result, actions are facilitated using visualization, which allows users to understand the relationships between software testing and results, interact with visualized Lean Canvas board and generate new ideas and discussion.

In this condition team members, the test manager, team leader, and tester read a single page document and scrutinize information about the under-investigation product or the project, tools, process and strategy. Visualising the test strategy within any project, the overall testing process is simplified.

3.2 Lean Canvas Design

In this research the author also analysed different existing design strategy possibilities for strategy visualization, such as fishbone diagram, strategy map design, empathy map design, Lean Canvas design and business process model and notation design. The author saw more benefits of adopting Lean Canvas as a base model for further test strategy visualization research [17].

According to [10], [12], [13] and [11], Lean Canvas is an adaptation of the Business Model Canvas by Alexander Osterwalder made in 2008, the model was created by Ash Maurya in the Lean Start-up. It is used for strategic management, lean start-up templates, and documenting the existing business models. Figure 3.1 provides an example of the standard Lean Canvas template. It is a single-page document that visualizes different parts of the product or business, such as value proposition, infrastructure, customers and finance section.

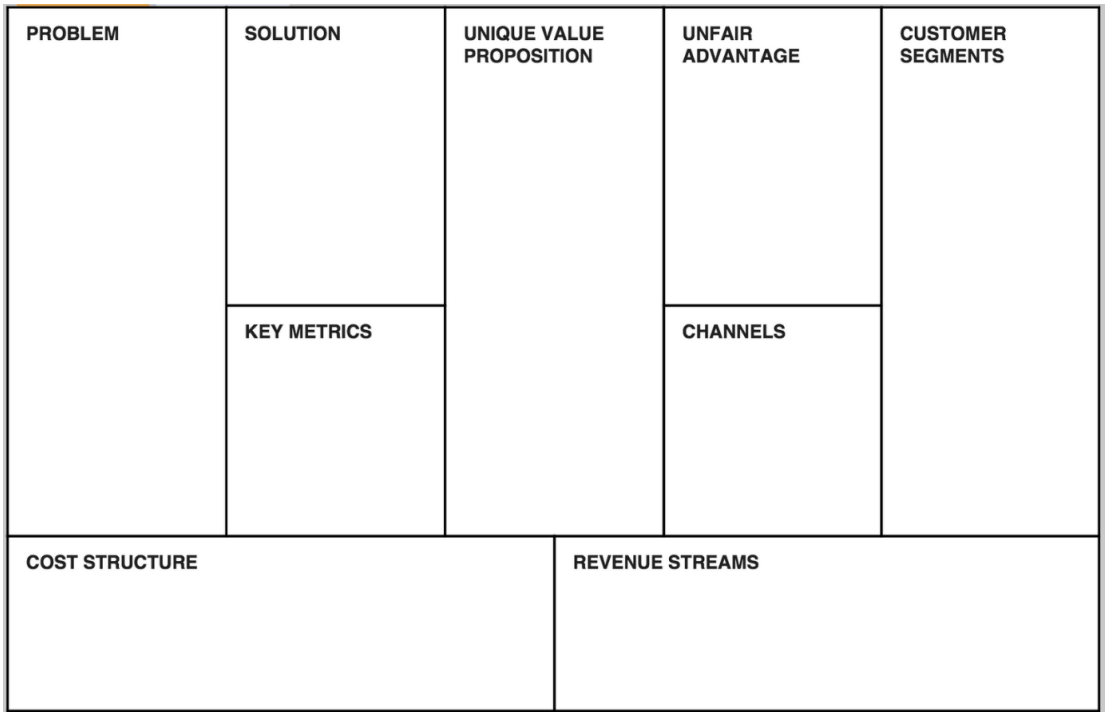

Fig. 3.1. Example of a standard Lean Canvas template.

In the broad view, it is also possible to use Lean Canvas in software testing [18], [19]. The Lean Canvas model may be adapted for the scrum software testing process that may directly impact the improvement of the software quality [17]. The author wanted to investigate whether the existing Lean Canvas model with nine blocks is sufficient for test strategy visualization or further changes are needed. In this process, the author explores user experience with Lean Canvas from a person's point of view, in our case, 'Software tester' is the user.

In terms of the user, experience design is not about interface behaviour. It is about human behaviour and adapting technology, accordingly. In the first step it should be uncovered what a designer is. A designer is an architect, a theoretical builder who creates fictional artefacts, which later become real products.

Marsh [14] claims that the team makes sure that the digital product is constructed according to the needs of the user and requests of the stakeholder. User experience design (UXD) is the designing process the objective of which is to design digital systems, interfaces and applications to offer the most efficient and straightforward user-friendliness. Thus, user experience design embraces theories of many disciplines, such as user interface design, usability, accessibility, information architecture and human-computer interaction.

Personas are fictional characters that have general characteristics of the persons who create the test strategy researched by the author. Before proceeding to the design of the Lean Canvas test strategy directly, it is necessary to identify the needs of persons. In our case they are the software tester, test manager, team member, and test leader are focused on first and foremost. The author used the agile approach to create persons; an agile user experience practitioner creates persons collaboratively. Grant and Mittal [14], [16] discuss that the person is a typical or standard user (the famous archetype), a fictional representation of possible target users, and it is used to set priorities and guide interface design decisions.

In the process of investigation of Lean Canvas adaptation for test strategy design, the author proceeded towards the user experience design. In this study, the author came to the conclusion that the

existing nine sub-blocks are not sufficient for strategy visualization. To better understand the problem, the author collected the persons' point of view. In the next step, the author re-evaluated the entire solution with Lean Canvas design and evaluation technique. Thus, re-evaluating the ideas from the person's point of view, learning the basics of user experience design and user interface design allows formulating suggestions and guidelines to develop a better-visualized test strategy of lean canvas.

4. DEVELOPMENT OF VISUALIZATION APPROACH

In the software development life cycle, the team may encounter different types of waste, this waste comes from different sources. If the team does not address waste, the project might take more time than planned. Waste may have direct or indirect impact on the development and test life cycle, in this situation, software development life cycle waste identification gains momentum (Fig. 4.1). Classification of waste further allowed to find the primary key meta title for the Lean Canvas board. The author suggests thinking critically to move towards more lean thinking with development and software testing before lean design canvas itself.

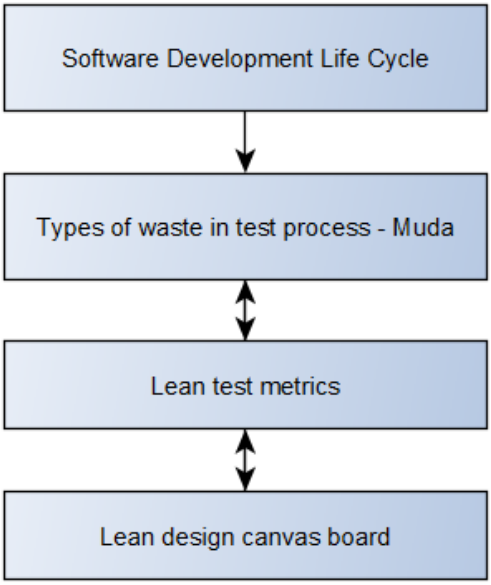

Fig. 4.1. Software development life cycle waste identification.

4.1 Test Activities Building Block

It is important to have appropriate titles of blocks of the canvas, each block focusing on a different factor. The titles guide users to get meaningful information quickly from the observation. In this section, creation of blocks, sub-blocks, titling and subtitling process are explained.

The three main steps to design Lean Canvas test activities building block include:

Step 1: In this step, the primary key meta title for the board is identified. A user can use the existing test strategy document or brainstorm with team members to understand the critical meta title for the board (e.g. Testing deliverables).

Step 2: In this step , the key meta title is split into smaller ones. Using the lean principles, the critical meta title is found and added on the Lean Canvas board. Later the meta description is added.

Step 3: In this step, the meta title or meta description is modified. The meta title or meta description is modified according to the software development life cycle.

The author described each step-in detail in visualized Lean Canvas test strategy design.

Visualized lean canvas test strategy design in steps:

Step 1: Identify the current type of software development process.

In this step, the software development process is identified (e.g.: agile, waterfall, lean).

Step 2: Identify the type of software testing.

In this step, the type of software testing is identified (e.g.: regression test, performance test).

Step 3: Use any existing test strategy or test documents.

In this step, investigate and use any existing test strategy or test documents or run a brainstorm session with the team in order to create one (e.g. test plan document, test strategy document).

Step 4: Identify the primary key meta title for the board and use empty canvas with the board and start adding sub-blocks (e.g. test levels, testing scope, test deliverables, risk).

Step 5: Split the key meta title into smaller ones and name the appropriate key meta title names for blocks and sub-blocks (e.g. manual, automation).

Step 6: Start filling the short meta description in sub-blocks (Figgie 4.2), (e.g. testing scope in sub-blocks as a scope of testing, out of scope).

Step 7: Visualized test strategy is ready to be used.

Step 8: In case it is needed, after each sprint or after development phase, re-review the existing test strategy and modify the meta title or meta description.

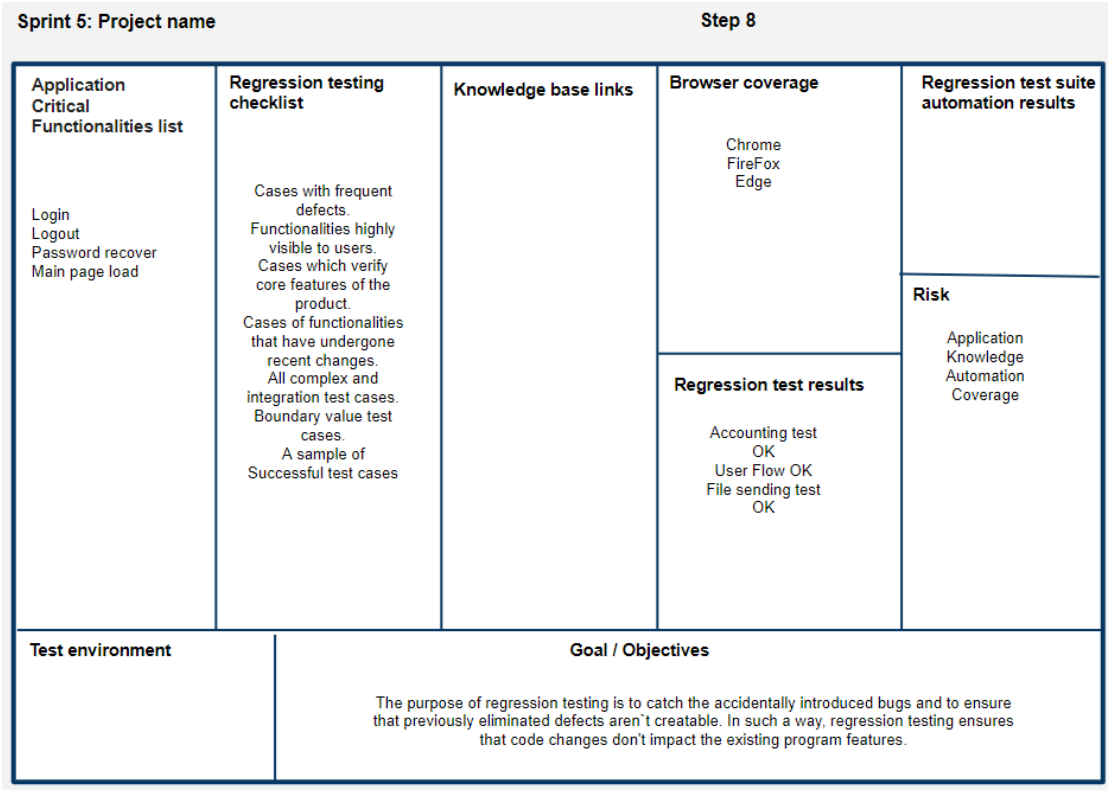

Fig. 4.2. Writing titles and filling description.

Considering the above-presented figure, the titles of the first block feed the transformation process and output feedback is transformed into the input. This concept is used to identify the right title for a lean canvas box.

Test process waste classification for TABB (Test activities building block) for Lean Canvas board (Fig. 4.3).

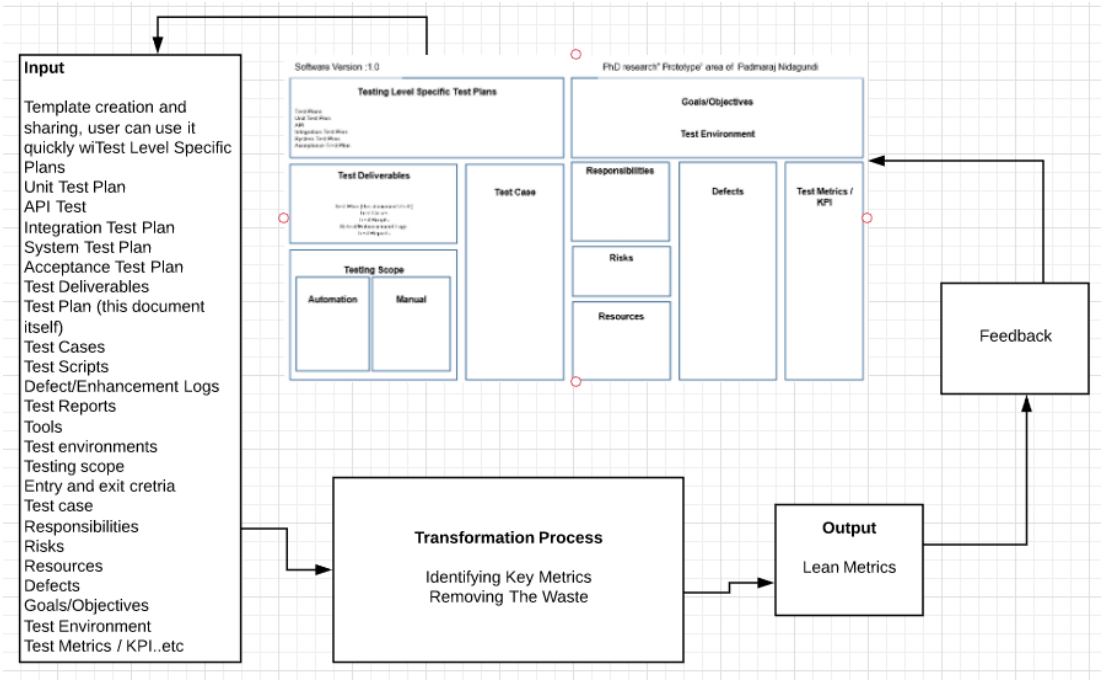

Fig. 4.3. Test activities building block (TABB) strategy for the Lean Canvas blocks and sub-blocks.

Input: Collection of critical terms used in the existing test strategy or test process, if there is no historical evidence available for a test strategy, then the team can get these from the brainstorm session.

Transformation process: In this phase, adopting the lean principles, the undesired meta title or meta description needs to be removed. The aim is to keep important titles only for the board [27]. While using the transformation process, users need to keep in mind 8 lean principles. Also, the user can receive feedback from the team on what they view as an essential title that needs to be visualized further.

Output: Collected key metrics is considered as a title, sub-titles or short description for the visualized Lean Canvas board.

Feedback: It is a collected key metrics proposed, further these key metrics titles are reused in the lean canvas board.

The preceding section presented the description of the TABB (test activities building block) for the Lean Canvas blocks and sub-blocks' lifecycle used to explain and visualize the Lean Canvas test strategy design in eight steps that allows understanding how to use it.

Formal Specification of TABB

The author bridges the gap between informal models and a formal model using the unified modelling language (UML) class diagram (Fig. 4.4).

In the first step, an empty board is created and the user adds an appropriates meta title based on the strategy document or brainstorming with team (e.g. testing deliverables). In the later stage, the user can add more meta titles or remove them. The user can add a description to the meta model, as already explained in the visualized Lean Canvas test strategy design in steps:

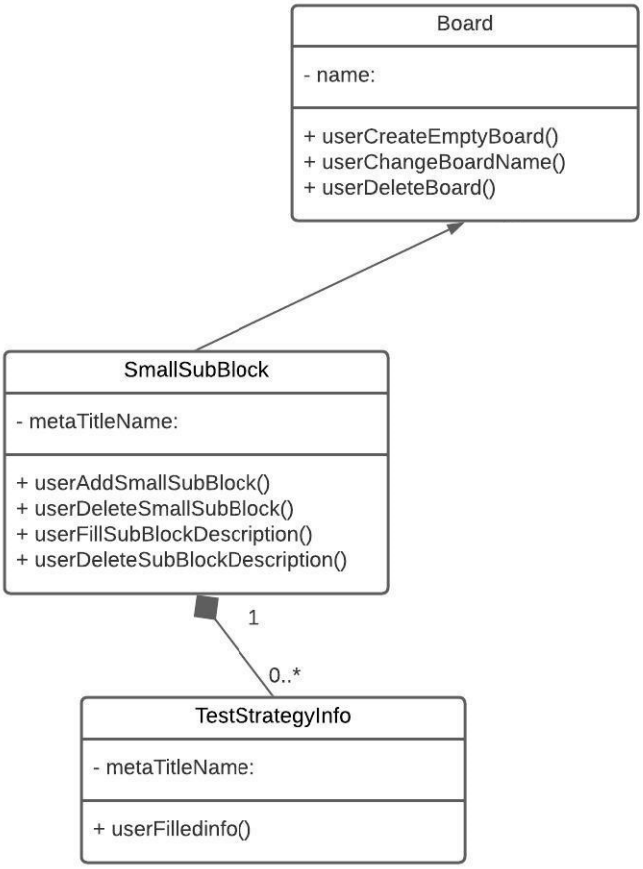

Fig. 4.4. Formal approach to TABB using UML class diagram.

As a result of investigation, it is further possible to develop a conceptual solution (Fig. 4.5). Formal methods have proven efficient in the development of an explanatory system.

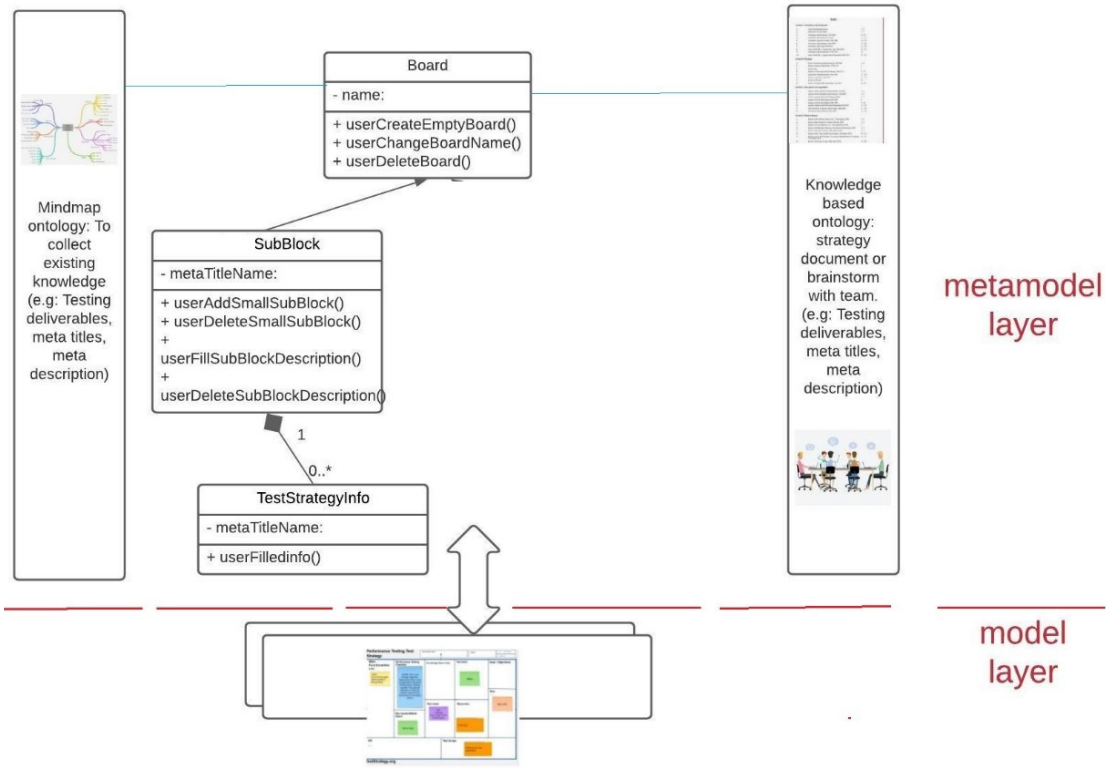

Fig. 4.5. Conceptual solution for a graphical representation of TABB generation.

As demonstrated above, knowledge comes in meta format from different sources, it depends on the user, and at a later stage information is put into the TABB model format. Considering all inputs discussed above, the author explains that the prototype system is implemented as Web application and described in the next section.

4.2 Prototype System Design

The study was conducted based on the experimental evidence of the pilot project. Identifying the appropriate technologies and building the test strategy, a web portal with ready-to-use strategy template is applied for the testing process. A number of visualized test strategy templates is available at www.teststrategy.org

The pilot project was created in four steps:

Step 1: This step provides essential explanations of pilot project possibilities. In this step, considering the hypothesis test strategy, visualization possibilities are investigated.

Step 2: Pilot project activities and possible solutions are developed. In this step, a number of use cases are identified. The pilot project user flows are developed at www.teststrategy.org for test strategy visualization.

Step 3: Advantages and loopholes of the pilot project are identified. In this step, strengths and limitations of the visualized software test strategy are identified after implementation of activities.

Step 4: This step provides a brief analysis, summing-up, and tips to move forward.

The author identified the system design requirements for building TABB. In the process of building the first system prototype, the author received several feedbacks. In the later stage, these feedbacks were

used for improvement of the system and TABB. In this process, some limitations were also identified. The triangulation method demonstrated that a set of questions should be focused on the users, such as testers, developers and test managers, to get feedback on the visualized test strategy. These questions are asked by individual users and adopters of the www.teststrategy.org.

The process of exploring the strengths and limitations of the visualized test activity building block strategy required application of the exploratory research method to develop better understanding of a relatively new research field. This method basically involves two steps: a) analysis of the comments and feedback from online users; b) multiple-case study done within live projects. The author used different methods of data collection and used triangulation as a technique to analyse the results (Fig. 4.6).

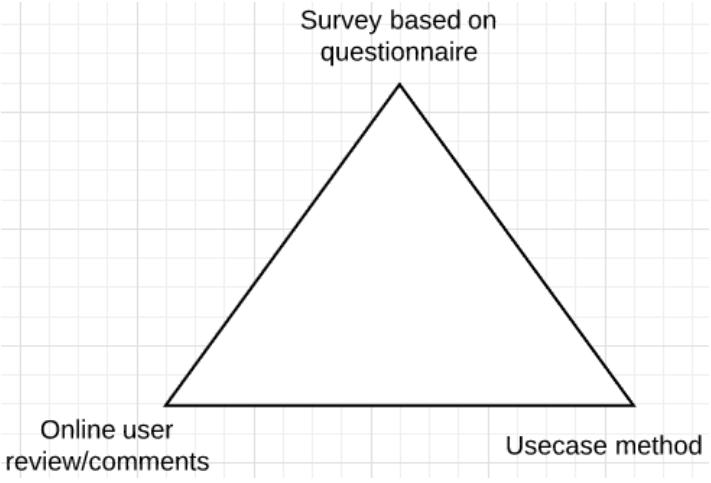

Fig. 4.6. Quantitative and qualitative triangulation of the visualized test strategy.

In the first step, comments and feedback given by online users are collected and analysed. Most users are software testers; these users are early adopters of www.teststrategy.org. Table 4.1 provides the overview of quantitative and qualitative effort.

Table 4.1

Quantitative and Qualitative Triangulation of the Visualized Test Strategy

Quantitative and qualitative triangulation of the visualized test strategy			
Survey based on questionnaire	Quantitative research for prototyping and implementing	Selected member group discussion at the initial stage	2 groups participated
		Selected member group discussion after the visualized Lean Canvas has been implemented	4 groups participated
		Questions and answers of the survey	33 individuals participated
Online user review and comments	Qualitative research for design improvement	Different feedback received at www.teststrategy.org.	15 independent feedbacks considered for design
Use case implementation	System design and improvement considerations	A number of users used ready TABB templates developed and stored at Gdrive www.teststrategy.org	Many designs implemented at www.teststrategy.org

In Step 2, a multiple-case study done with live projects was used to analyse possible use cases and the experience of the interviewees with the visualized test activity building block strategy. In the overall process, the triangulation method was used to collect the results.

4.3 Adoption of a New Strategy

According to [22], successful adoption of a new strategy depends on the purpose, design/methodology, findings, practical implications, originality/value, and key values. In this research, the author focuses on all of them. The author suggests using the proposed steps as a guideline for starting new or running the existing software development or test projects. A user can adopt sample TABB ready-to-use templates proposed by the author available at the portal teststrategy.org (Fig. 4.7).

- Step 1: Agree or propose that your team visualizes the test strategy using the lean one-page template.
- Step 2: Investigate the existing test strategy document or test process document to get an overview. If the team does not have any, ask questions or brainstorm with the team members to clarify.
- Step 3: Use the ready-to-use template from teststrategy.org or GitHub resource.
- Step 4: Rename the block and sub-block meta title or meta description according to your project and the type of testing.
- Step 5: Share 'Test Activity Building Block Strategy Using Lean Canvas – Visualized Test Strategy' with the team to update it when the project lifecycle changes.

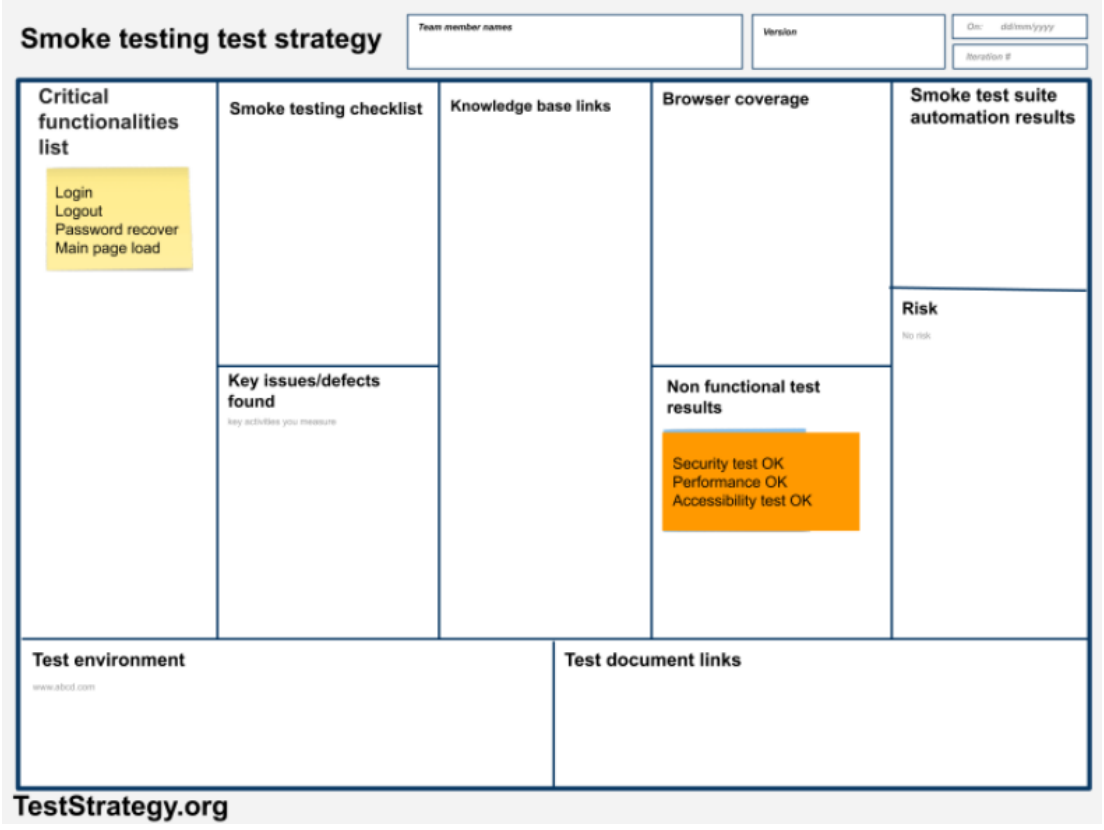

Fig. 4.7. Sample template from teststrategy.org.

In this research, the author wishes to describe the new proposed test activities building block strategy more profoundly to further finetune it. Thus, Chapter 4 concludes with the analysis of the following strengths and limitations. Table 4.2 presents the strengths and limitations of the visualized software test strategy.

Table 4.2

Strengths and Limitations of the Visualized Software Test Strategy

Strengths	Limitations
An apparent strength is the recognizing of the centrality of value. The purpose of constructing a visualized test strategy model is to capture and deliver value to the software test process.	A clear consensus was identified in the absence of external factors, such as software test process maturity, regulatory compliance for special software projects. The need for a skilled team, team and work relationship and detailed estimation of test effort in the visualized test strategy.
During the online user survey and processing respondent comments, the visual representation, usefulness of the template in the test process, usability and simplicity of the test strategy have been mentioned as strengths.	Individual building blocks were investigated, and the repeating limitation was identified. All interviewees mentioned the confusion about the building blocks, which are not sufficient to add a larger set of information
The strength of the visualized test strategy is that it functions as a communication tool between the teams, test managers, individual testers, remote team members and partners. Because of the visualized format, the structure and simplicity of the test strategy contribute to better communication between different disciplines.	The last limitation is focused on the interaction of teams and the value of the creator of the visualized test strategy. It should be taken into account that in agile testing, all team members should be involved in the visualized test strategy design process.
The visual representation of the test strategy is well focused on internal factors.	Disregard of external factors, such as process maturity, skilled team, team and work relationship and estimation of test effort.
The coverage of different dimensions of a visualized test strategy, such as risks, goals/objectives, testing level specific test plans, test metrics, key performance indicator.	The visualized test strategy is based on building blocks of different levels of abstraction. This results in the emphasis solely on building blocks, such as test case, main functionality list, long text descriptions.

5. APPROBATION AND TEST ACTIVITY BUILDING BLOCK STRATEGY

In this research, the author has worked closely with two software development companies – a medium- and a large-size company. The author has found a platform to implement TABB within the ongoing software development projects. The author selected a medium-size company which deals with the development of software products for the end customer. At the company, a number of software development projects were developed using different tools and different development methodologies. The author proposed an idea of testing strategy design. In this way, the author collected results and valuable feedback, which were documented in the section. The author also got the opportunity to work in software development company. It is one of the leading cloud software development product-based companies in Europe. All products development life cycles are agile . The company itself is very big, the author found a legacy project among new product development projects for which the author proposed an idea, which was accepted, and received feedback.

5.1 Evaluation of a TABB

The test activity building block (TABB) strategy on the Lean Canvas for test strategy has been evaluated in two software development companies within a number of agile processes-oriented projects. In this process, 22 projects are evaluated and in the Thesis outcomes of 6 projects are documented. In the case study analysis process the author found that each company has used different technologies, tools, testing process, and unique business domain software.

5.2 Scope of the Case Study

The scope of the case study covers the testing activity and adoption of building block strategy on Lean Canvas in companies. Although the scope of the project included different business domains, different types of software applications and their development of life cycles, the study focuses on different possible domains to cover different types of testing done in the software development life cycle (Table 5.1).

Table 5.1

Type of Project, Business Domain, Type of Test Strategy Created and Type of Development Methodology

No.	Type of project	Business domain	Type of test strategy created	Type of development	Documentation size (up to) in pages (WWWWWH test strategy)	Documentation effort (up to) in each software release in hours
1	Micro-service	Procurement	Test automation	Agile – Kanban	14	4
2	Web application	Entertainment	Performance testing	Agile – Scrum	12	4
3	Web application	Entertainment	Security testing	Agile – Scrum	7	4
4	Web application	Ecommerce	E2E AI testing	Agile – Scrum	9	4
5	Web application	TAX declaration	Legacy testing	Waterfall	22	8
6	Web application	Ecommerce	DevOps testing	DevOps testing	14	4

* Documentation size (up to) in pages (WWWWWH test strategy) – page size goes higher and/or lower depending on the description written in the document.

** Documentation effort (up to) in each software release in hours – documentation effort in hours go higher and lower depend on time allocated in the project planning session.

5.2.1 Security Testing Case Study

This sub-section highlights the adoption of the TABB strategy in one of the projects where security tests are carried out (Table 5.1). This project is a web-based application following the agile Scrum model. In this project, the visualized test activity building block strategy is formulated for the web-application security testing.

Considering the earlier investigation of a project that adopted security testing, several questions were asked to the team members of the group. The answers were recorded for further test strategy development (Table 5.2).

Table 5.2

Security Testing Observations, Questions and Project Details

Project details	Project title: EveryHair Team size: 10 Type of SDLC: Agile – Lean Type of testing: Security Testing Team members: Analyst, Developer, QA, Architect		
Questions	Q1: What kind of testing is done? Function testing, UAT	Q2: What kind of challenges for web application security does the test team face? Injection Broken authentication Sensitive data exposure XML External Entities (XXE) Broken Access control Security misconfigurations Cross Site Scripting (XSS) Insecure Deserialization	Q3: Who is responsible for software quality? Testers and partially development team
	Q4: What kind of documentation does your team draw up? Test document Security test report	Q5: Does the project have a test strategy in the document or visualized form? No	Q6: The team would like to have a visualized one-page test strategy. Yes
Observations	EveryHair is a user engagement solution for salon users. Project technologies: Azure, C#, Progressive web, AngularJs, SQL, Git Test management: High level test document, Manual testing Test strategy: Test document and release checklist		

Considering all the above inputs, analysing the existing document and adopting the TABB allow creating the visualized security test strategy (Fig. 5.1). The visualized test activity building block strategy is created for the user, it should be followed in the way the author described in Section 4.1.

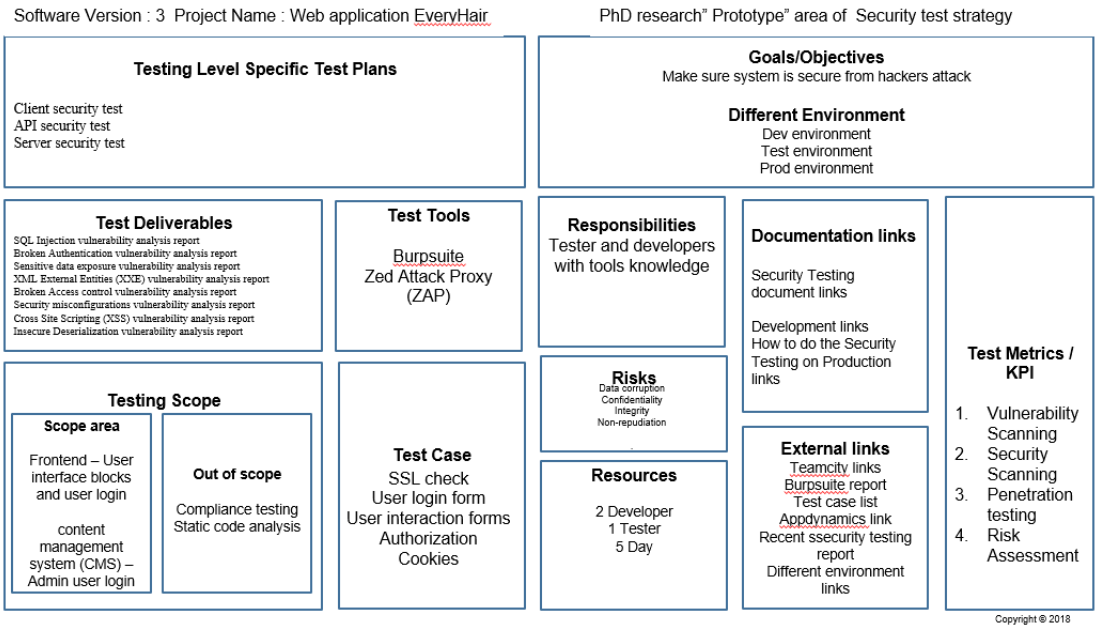

Fig. 5.1. Visualized security test strategy.

A visualized test activity building block strategy for security testing solves many issues:

- appropriate titles for the sub-blocks make it easy to understand for all internal team members;
- different types of the security test are identified;
- test deliverable reports and task are identified;
- the testing scope is recognised very clearly and out of scope issues are addressed;
- the security test tools for testing coverage are identified for the project;
- a test case is recognised and organised according to the priorities;
- the goal of the security test, different testing environment and human resources needed are identified;
- relevant documentation links and external links are mapped;
- KPI is used to monitor the progress and consequences of testing;
- lightweight documentation strategy fits in this project;
- related internal teams can also use security testing test activity building block strategy;
- a non-technical team member can also use the test activity building block strategy for communication or gathering the results.

5.3 Analysis of Case Study Results

This section presents six illustrative case designs to show how to adapt and apply the visualized test activity building block strategy. The cases cover different types of software projects, business domains, technologies and types of software testing. In the initial phase, several questions were asked to the team before starting the test activity building block strategy. In parallel, the existing test and development documents were reviewed to collect the project information. In addition, the visualized test strategy is created and feedback is collected and discussed in this section. Table 5.3 presents the analysis of the results of the empirical study considering the essential comparison of the type of testing, documentation size [26], communication and visualization.

Table 5.3

Analysis of Results of Empirical Study

No.	Type of project	Documentation size (up to) in pages (WWWWWH test strategy)*	Documentation size (up to) in pages after adopting TABB	Reduced documentation pages in percentage	Delta pages	Documentation effort (up to) in each software release in hours**	Documentation size (up to) in pages after adopting TABB
1	Microservice [4.4.1]	14	5	64 %	9	4	2
2	Web application [4.4.2]	12	7	42 %	5	4	2
3	Web application [4.4.3]	7	3	57 %	4	4	2
4	Web application [4.4.4]	9	4	56 %	5	4	2
5	Web application [4.4.5]	22	13	41 %	9	8	4
6	Web application [4.4.6]	14	6	57 %	8	4	2

* Documentation size (up to) in pages after adopting TABB – page size goes higher and lower depending on the description written inside the document. From the above collected data it was noticed that the minimum pages are reduced comparing to the WWWWWH test strategy pages.

** Documentation effort (up to) in each software release in hours – documentation effort in hours goes higher and lower depending on the time allocated in the project planning session. From the above collected data it was noticed that the documentation effort minimum hours are reduced comparing to the WWWWWH test strategy.

The adoption of TABB implies the improvement expressed according to variance in 5.0 pages and 0.6 hours. The reduced documentation pages average 53 % and percentage variance – 0.8.

The author's research on the creation of a one-page visualized test strategy (TABB) does not lead to the commonly noticed consequences relating to the following test strategy and test management issues:

- Quality documentation. Visualization test strategy does not have impact on document artifact, coordination, control, delivery, or support of an item required for quality assurance purposes.
- Communication. A one-page test strategy simplifies the inter-team communication and communication within a geographically distributed team.
- Documentation effort. Documentation time and effort is shortened.

CONCLUSIONS AND THE SCOPE OF FURTHER RESEARCH

The aim of the Doctoral Thesis is to develop a visualized software testing strategy utilizing Lean Canvas model.

On the way to achieving this aim, the following results have been obtained:

- Test process waste classification has been performed to create better meta titles.
- In Ontology, mind maps have been analysed for knowledge management system. In the next step, mind map extraction has been used for creation of meta titles.
- For systems design, use cases have been collected and online web-based application has been developed, online web-based limitations have been considered.
- To get better insights, both primary and secondary research data have been analysed for TABB.
- Adoption of TABB steps has been outlined.
- The existing test strategies WWWWWH and STD (story, development, testing) adopted by the industry have been documented.
- The overall results of approbation allow stating that TABB visualized test strategy has been successfully adopted and investigated in functional and non-functional projects.
- The TABB test strategy proposed by the author has been compared with other testing strategy.
- Analysis of different case studies has been conducted for test automation, performance testing, security testing, E2E testing, legacy software development and DevOps projects; the results have been evaluated and documented.
- The developed methodology and methods, combining experiments, case studies and expert interviews, have been comprehensively assessed.

Within the Doctoral Thesis, the following conclusions have been made:

- The adoption of TABB approach reduced the documentation pages to average 53 % and the time to develop documentation approximately to 50 %.
- The main building blocks of security testing are goals/objectives, test environment, risks, testing scope, testing level-specific test plans, external links and resources.
- TABB is adaptable in test automation, performance testing, security testing, E2E AI testing, legacy testing and DevOps testing. It is adaptable in both functional and non-functional testing.
- TABB visualized test strategy is adaptable in Agile (Kanban, Scrum), Waterfall, DevOps and other methodology projects.
- TABB adaptation has been analysed and evaluated. It does not lead to the consequences related to commonly noticed test strategy and test management issues such as communication, visualisation, internal team conflicts, software defects, documentation effort.

The results of the Doctoral Thesis provide an opportunity for conducting further research in the following areas:

- Extension of reusability of TABB, including the evaluation of different software development projects, such as embedded software, firmware etc.
- Development of tools or plug-ins to integrate TABB in test reporting tools.
- Development of a more simplified online solution at teststrategy.org and advancement of the visualized test strategy creation using machine learning and artificial intelligence. In this process, the user submits necessary project information and gets back several ready-to-use visualized test strategy models.
- Creating awareness about the 'visualized test strategy' model in the software testing community to increase its adaptation. Collecting constant feedback from the community and applying this feedback in better and new model creation.

BIBLIOGRAPHY

1. Black, R.: Agile Testing Foundations: An ISTQB Foundation Level Agile Tester guide. 2017, pp. 12.
2. Spillner, A., Linz, T., Schaefer, H.: Software Testing Foundations. 2014, pp. 23.
3. Graham, D., Fewster, M.: Experiences of Test Automation. 2012, pp. 32.
4. Watters, A., Pinkster, I., Janssen, D.: Integrated Test Design and Automation. 2000, pp. 34–76.
5. Black, R., Veenendaal, V., Graham, D.: Foundations of Software Testing: ISTQB Certification (3e). 2012, pp. 49.
6. Test Process Improvement. – Internet / https://www.qualitestgroup.com/assets/test-process-improvement.pdf.
7. Black, R.: Pragmatic Software Testing: Becoming an Effective and Efficient Test Professional, 2016.
8. Robinson, J., Nanda, V.: Six Sigma Software Quality Improvement. 2011.
9. Fleurey, F., Haugen, Q., Johansen M F.: A Survey of Empirics of Strategies for Software Product Line Testing. IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops, Berlin, Germany, 21-25 March 2011.
10. Maurya, A.: Running Lean: Iterate from Plan A to a Plan That Works (Lean Series), 2012.
11. IPC Group.: Eat. Sleep. Create. Entrepreneur's notebook Lean Canvas Business Ideas Journal, 2018.
12. Lin W.: Lean Start-up Canvas Notebook, 2018.
13. Mirabelli V.: The 1 Page Project: A Practical Guide to Using the Lean Project Canvas to Focus Your Projects on What Matters, 2019
14. Marsh J.: UX for Beginners: A Crash Course in 100 Short Lessons, 2016.
15. Grant W.: 101 UX Principles: A definitive design guide, 2018.
16. Mittal N M.: Software Design Patterns 2.0: UI and UX Design Principles and Tips
17. Nidagundi, P., Novickis, L.: Possibilities about the design lean canvas model and its adaptation in the agile testing. Symposium for Young Scientists in Technology, Engineering and Mathematics, Volume 1853, SYSTEM 2017, Kaunas, Lithuania, 28 April, pp. 20–23.
18. Nidagundi, P., Novickis, L.: Introduction to Lean Canvas Transformation Models and Metrics in Software Testing, APPLIED COMPUTER SYSTEMS, Year 2016, pp 30–36.
19. Nidagundi, P., Novickis, L.: New method for mobile application testing using lean canvas to improving the test strategy, 12th International Scientific and Technical Conference "Computer Science and Information Technologies", Lviv, Ukraine, September 05-08, 2017.
20. World Quality Report – Sogeti. – Internet / https://www.sogeti.com/explore/reports/world-quality-report-2017-2018/.
21. WQR – Sogeti. – Internet / https://www.sogeti.com/globalassets/global/wqr-201819/wqr-2018-19_secured.pdf
22. Graner, M., Mißler-Behr., M: 22. Key determinants of the successful adoption of new product development methods, European Journal of Innovation Management 16(3), 2013.
23. Visualizing the Complex Software Development Process. - Internet / https://www.cmcrossroads.com/article/visualizing-complex-software-development-process.
24. Visualize Strategy. – Internet / http://udleditions.cast.org/strategy_visualize.html
25. Tan, K., Platts, K.: Strategy visualisation: knowing, understanding, and formulating.
26. Penella S.: Overcoming Challenges to Good Test Documentation. – Internet / https://www.stickyminds.com/article/overcoming-challenges-good-test-documentation.
27. Nidagundi, P., Novickis, L.: Introduction to Lean Canvas Transformation Models and Metrics in Software Testing, APPLIED COMPUTER SYSTEMS. 2016, pp. 30–36.
28. Proactive Vs Reactive: Which Approach is Better to Attain Quality. – Internet / https://www.qualityze.com/proactive-vs-reactive-approach-better-attain-quality/
29. Rubin, K. S.: Essential. Scrum: A Practical Guide to the Most Popular Agile Process (Addison-Wesley Signature): A Practical Guide to The Most Popular Agile Process (Addison-Wesley Signature Series (Cohn)), 2012.

30. Chen, N.: std 829-2008 and Agile Process – Can They Work Together? Year 2013.
31. C. Unger-Windeler, J. Klünder, and K. Schneider: "A Mapping Study on Product Owners in Industry: Identifying Future Research Directions," 2019 IEEE/ACM International Conference on Software and System Processes (ICSSP), May 25-26, 2019. Montreal, Canada, 2019, pp. 135–144.

Padmaraj Nidagundi was born in 1985 in Vijayapura, Karnataka, India. He obtained his Bachelor's degree in Information Science and Engineering from Visvesvaraya Technological University in 2010 and a Master's degree in Computer Engineering in 2014. Since 2010, he has been a working at different software development companies. He is currently a researcher with Riga Technical University. His most successful field of activity in previous years has been software development and testing, where he gained significant international work experience. In recent years, he has been a member of teaching staff and participated in research projects. His research interests are software development, quantum computing, cybersecurity, and artificial intelligence.