

Introduction to adoption of lean canvas in software test architecture design

Padmaraj Nidagundi^{1*}, Margarita Lukjanska²

¹ Riga Technical University, Kaļķu iela 1, Riga, Latvia.

² Politecnico di Milano, Milano, Italy.

Abstract

The growth of the software dependent businesses, as well as the use of electronic devices in daily life, brings new challenges requiring the software to work error free all the time, to achieve this goal software needs to be sufficiently and effectively tested during various development phases. Most software development companies make great efforts in testing; it is even more difficult to reach the error-free software goal. Different software development methodologies (e.g. traditional waterfall, agile) brought in a new dimension for both - development and testing - introducing new technologies and tools. In software test automation the test architecture design plays a key role in managing written test cases and effectively executing them. Having the more effective software test automation architecture design in test process saves resources, efforts and reduces the technical depth.

This paper provides the new dimension and possibilities of using lean canvas in the design of the software test architecture.

Keywords: *software testing, test automation, lean canvas, test architecture design, software verification, software validation.*

1. Introduction

Nowadays the software development process accelerated with the adaptation of various new tools, and software testing gains significant value in early verification and validation of the built software. Few decades before software development and testing were more time consuming and expensive process. In recent years the focus changed to agile software development and testing that aims to deliver product fast. Various software applications also grew large and more complex, nevertheless, the software quality requirements and elimination of any bugs is still a topical issue. From software development to software testing the architecture design plays a key role to build high-quality software. Software development adopted different types of development methodologies and each mythology has their own property and contribution to software development process. Software development, testing, and delivery to production needs run in parallel, all these steps influence the entire development and testing process directly. In recent year`s software development also adopted many engineering design processes. The engineering design process is a methodical series for developing functional products and processes. As discussed further the engineering design process has a many stages in the same way as software development..

Engineering design process steps such as 1) ask, 2) imagine, 3) plan, 4) create, 5) improve can be applied in software development process, as well [3]. Comparing the engineering design process to software development process - both ideally share the same value, namely, building the usable product/software. In software development software architect [8] is a key responsible person for the framework design, project management, design, quality, and process and documentation. Software test architect is responsible for the developing the test architecture to test planning, test development, test interpretation and test harness creation.

Lean canvas principles are novel to both software development and engineering design practices, specifically, applying them at the architecture level. The main aim of this paper is to shed some light on the possible adoption ways of the lean canvas in software test architecture design.

A. Scientific Novelty of Paper

Lean canvas is used for evaluation of the business models on the white board with empty titled boxes. It is a one-page lightweight document that helps to plan product development from its start to end phase and evaluate marketing fit.

*Corresponding author: padmaraj.nidagundi@gmail.com

Our core contribution is to identify possibilities for adapting lean canvas principles in software test architecture design..

Specifically:

- Design process for the software test architecture Scope for the lean software test architecture design
- Analysis of the complex system design for test architecture design
- Application of the lean principles in the software test architecture
- Identification of the software test architecture design optimization
- Design of the lean software test automation architecture
- Lean test metrics in software test automation architecture
- Lean canvas transformation models for test automation architecture

B. Structure of Paper

This paper is divided into three main sections and conclusions. Section 2 Paper begins with problem formation and in next. Section 3 design process and engineering analysis. Section 4 lean canvas principles for software development, and the related research. Section 5 Scope lean software test architecture design.

2. The Problem Statement

Software test architecture plays a key role in achieving effective software testing and related management activities. Not having properly designed test architecture framework could cause a delay in test automation, prevent or delay complex test cases testing and increase overall the software testing costs. The most common test automation architecture challenges are:

- Lack of proper test framework
- Lack of test environment
- Lack of test automation tools implementation
- Ineffective test automation techniques
- Failure to select test design, test data creation, release process techniques
- Lack of the right test architecture
- Ambiguity in test strategy
- Difficulties in handling custom built test automation tools
- Selecting the right tools and technologies for test automation

3. Related research

3.1 Principles and elements of the science of design

Design play key role in product success or failure in most of area and design bring the significant value to the product. Most of the companies and industry doing a lot of research and building the new tools for better design [1]. Effective design can bring effective product and that drive the business and bring more money into it.

Steps in design process

Defining the problem - It is very important to define what is problem user-facing or for whom we developing the product. Once we clearly define the problem then we can work around to solve the problems.

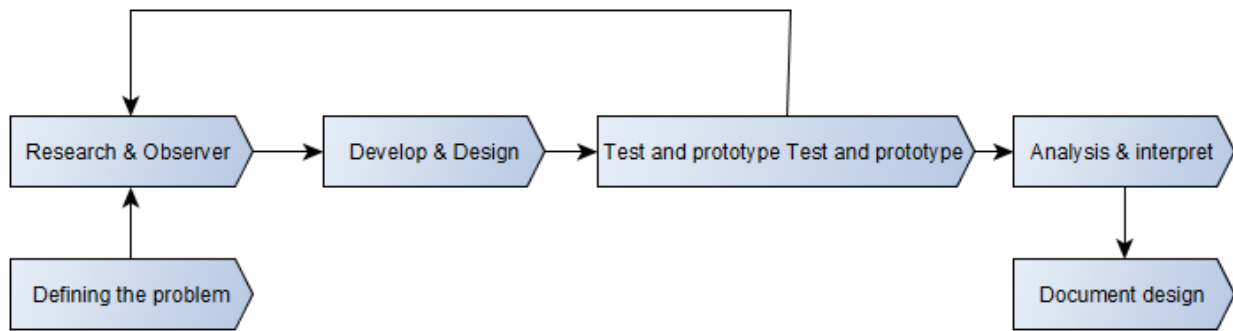


Figure 1: Steps in design process

- Research & Observer - In this step basic to an in-depth analysis done with research data or sample data. Data abstraction can help to set up a milestone for the next step.
- Develop & Design - In this step we use data to make the prototype and set a research framework so we work under that. First model development helps to know the early level success and failure of the product.
- Test and prototype - In this step product design is it tested with real or early stage users to check and give a feedback for the prototype.
- Analysis & interpret - In this step, continuous small feedback is given to developing a design stage for the to improve the design.
- Document design - Once a design is finalized and documentation is generated for the further references.

As we view in the above figure in test and prototype bring the validation and evaluation methods. The science of design brings new models and new opportunity solutions for the business to change their existing products with time and increase the customer stratification.

3.2 Iterative improvement of design

Make an observation, make a hypothesis, create and define an experiment to prove or disprove the hypothesis, Perform the experiment/test, Interpret the results of the experiment and find the result. Go back to the first step.

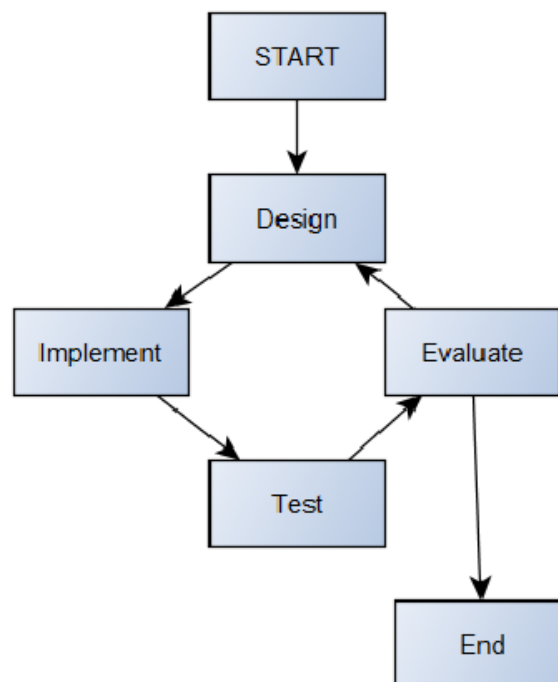


Figure 2: Iterative improvement of design steps

As we view in the above figure iterative improvement of design [1,2] we take initial design or a first model and start in design process, implement and basic design and test it in front of real or early user of the product, this process bring some intermediate results or observation collecting those data can further feedback to design to improve or use those data and finalize the design process. The end product is developed in such way that user can use it in an easy way and get maximum benefit from the product. Iterative improvement of design gives the constant feedback so the user can find the optimized design to the end users. One side iterative improvement of design adds a value to existing product and one side also brings new ideas of future implementation.

The design of science can be archived to multiple disciplines and for multiple purposes, my research is in this article is about how we can use the design in the software testing architecture. As we know the architecture play a key role in the building any computer based system, with software development to software testing architect bring the great value for the system to work it in an appropriate way. My research also towards the finding the lean architecture design advantages in test architecture. As we know software testing play key role in validating and verifying the software in early stage and reduce the overall cost. It's important to well-tested software need to deliver the end customer to improve the software quality..

3.3 The engineering analysis

The engineering analyses is a very basic and important tool for the performing and practicing engineering tasks in daily routine life. The analysis is used for the creations, decision-making, and problem-solving. As from history the scientist discovers the new things for humankind and engineers develop that idea and make it real to its users.

Engineers also make decisions some of the critical once those can affect the root or the system architecture and bring the very high impact. Engineers made a decision about the design [2,4], manufacturing, maintenance & expected conditions where critical decision need to be taken to go one step further more. Engineers solve the problem related to the design ambiguity, manufacturing fault, cost control, quality issues. An engineer can also make task creation, decision-making, and problem-solving.

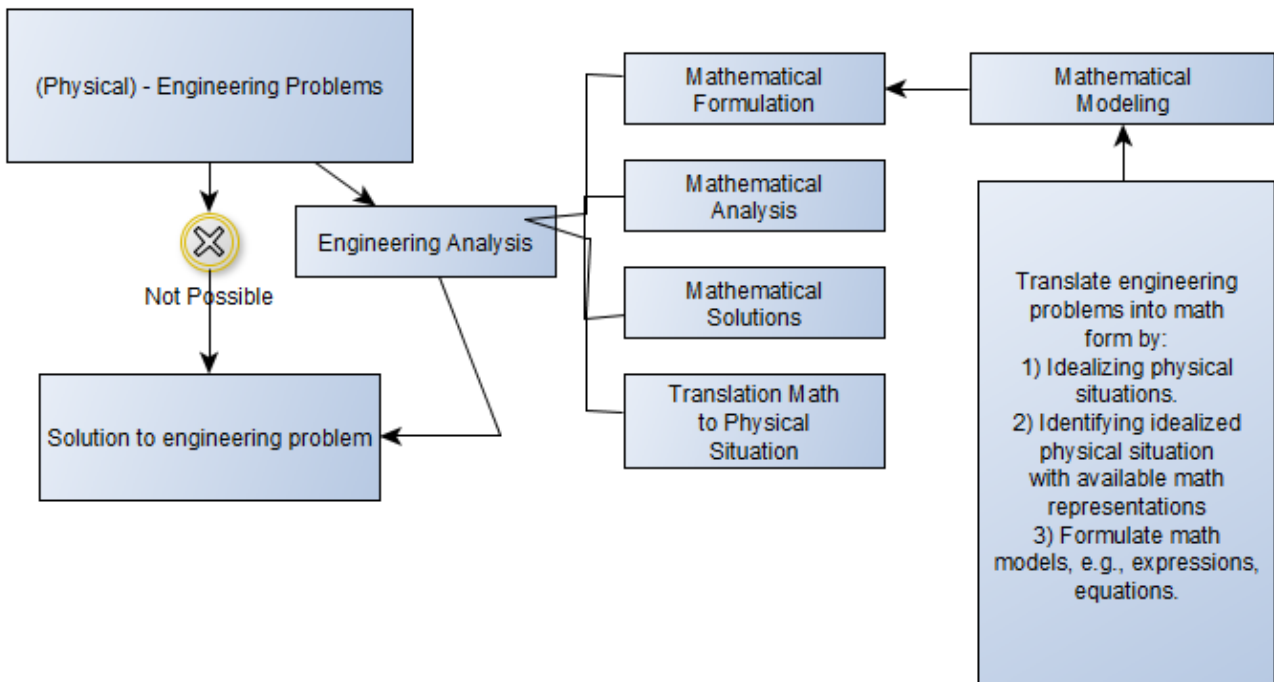


Figure 3: Model for the engineering analysis by mathematical modeling

As we seen in above figure the engineering analysis is core is mathematics [2.3], math plays a key principal role as a servant to engineering design and engineering practices. Mathematical modeling is a practice involving the translation of engineering items to mathematical forms.

3.4 Stages of engineering analysis for design

There are four stages in it, the first stage is indenting the real core physical problem and gathering the more specification about the problem. In second stage idealization of actual physical problem mathematic demission analysis. In later stage, mathematic modeling and analysis are done. At the end and in the last stage the results will be applied. Engineering analysis gathers the information that can be further used for detailed design.

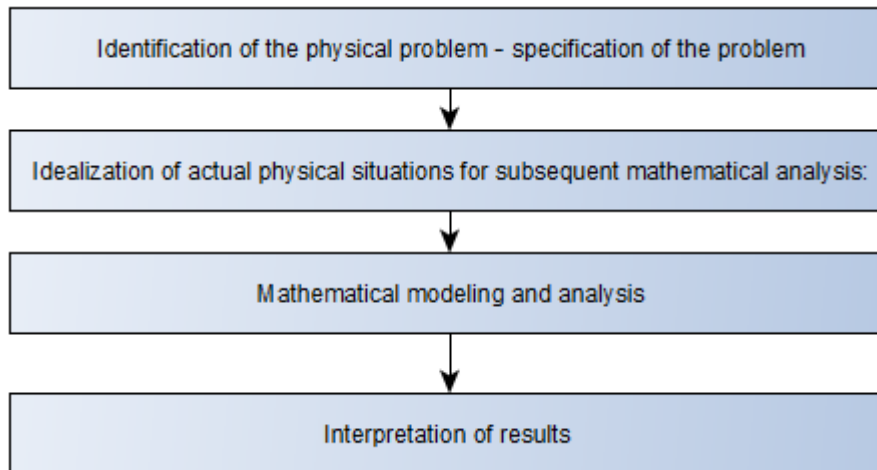


Figure 4: Stages of engineering analysis

4. Lean development and software architecture and test architecture design approach

Lean term grows from decades ago from manufacturing to lean software development. Lean mean elimination of the waste in the Japanese language it called as a “Muda”. Lear core principle says whatever we do it need to bring the value and customer willing to pay for that action.

Lean philosophy derived from the Toyota production system in 1990 in Japan from manufacturing industry. Lean work on 7 core principles 1) Transport 2) Inventory 3) Motion 4) Waiting 5) Overproduction 6) Over processing 7) Defects. The lean philosophy also focuses on the lean leadership who drive the lean team. Lean leadership address continues improvement principles 1) Challenge 2) Kaizen 3) Genchi Genbustsu..

Nowadays software development also adopted the lean development principles also called as lean software development [6]. The lean software development focuses on thinking all team members on its seven principles.

1. Eliminate waste
2. Increase Feedback
3. Delay commitment
4. Deliver fast
5. Build integrity in
6. Empower the team

Let`s identify the lean software waste to understand more

Table 1. Identify the lean software waste to understand more

ManufacturingWaste	Software Waste
Inventory	Partially developed software / code
Overproduction	Extra features
Extra processing	Software releasing
Motion	Task switching
Transportation	Handoffs
Waiting	Delays
Defects	Defects / Software bugs

4.1 Software test architecture

The Institute of Electrical and Electronics Engineers proposed IEEE-Std-1471-2000, put forward Practice for Architectural Description of Software- Intensive System, [IEEE00] – to start a basic conceptual framework and vocabulary can be used later during the design of software architecture. It will help to provide detailed guidelines for representing a software architectural description, and also to encourage sound architectural design practices. Software architecture plays a key role in making software better for the end users and it drives the business value.

Software architecture begins from the requirement architecture

1. Understanding the software requirement, technical environment details, and constraints
2. Understanding the problem domain / business domain
3. Identifying the possible objectives (e. g. Algorithms, technologies, best practices)
4. Identifying the and subsystem the solutions (e. g. reusable components)
5. Selection of appropriate solution (for reuse or future implantation)
6. Assign solution for main and subsystems
7. Identify and perform a risk analysis
8. Document the architecture design and subsystem interconnection with user interfaces
9. Review stakeholder needs and fulfilling the business objective

Software architecture significantly contributes the making system design more systematic this will help to build the subsystem with it in an easy way..

Core advantages of having better software architecture

1. Software architecture enhances the software reliability
2. Increase the performance
3. Enable software evolution
4. Software maintenance cost will reduce
5. Bring value to the agile software development
6. Risk minimization
7. Efficient software development
8. Accessible knowledge and decision

4.2 Test automation architecture design approach

As we understood how much software design an architecture is essential in same way software test architecture is very important to test the developed software. A software can be tested in fast manual or automated way if we have the efficient test architecture. Software testing broadly classified as a Manual testing and Automated testing.

Manual software testing architecture: In practical clear software test environment / clone environment test architecture itself can be used for testing purpose.

Test automation architecture: Its method of automating the human testing effort to test the software and reduce the regression cycles time and cost. This also in involves design the test framework, coordinating the implementation of test tools, creating the test environment for the test automation process. In the big picture, test automation process needs to be run in parallel to it. Test automation needs to support the development in all of its development life stages. One side this test automation architecture supports the test coverage and one side enhances the maintain flexibility.

There are much different to test automation, widely used GUI and API driven. In GUI software UI is tested and behaviors are recorded in negative and positive testing. In API testing software is validated with the help of tools.

Unit testing: when software complexity grows it's become more important to test different components. Unit test confirms that all sub-component of software working normally in different circumstances.

Test automation architecture support to test software as an OS independent, data drive, customizable reporting, easy to debugging, support the version control, run in distributed environment. Now a day's number of tools available those come with own architecture environment.

4.3 Identifying the scope for the lean software test architecture design

The growth of importance of developing and delivering error-free software given the new dimension for the test automation, it's very important nowadays to have a proper test automation environment and architecture to make test process more efficient. The rapid growth of agile software development also pushed software testing new dimension with new cutting edge tools [4]. Continuous integration and continuous deployment tools helping the team to send their code to test environment in parallel. One side deploying code continues and another side testing the code with automation continues same time.

In complex software development and a number of a team working on the different environment where the number of developer team developing and testing team working in parallel to test the code in such time it is very important to keep things simple and lean. As we know already lean principles and lean software development, not it's important to think lean software test architecture. As we know the software architecture it's a system design that shows the major components and their relationship. Software architecture also directly linked different IT and business components.

4.3.1 Goals of software architecture

The basic goal of the software architecture is to recognize the right requirement's those are linked with the structure of the application. The software architecture also helps to identify and reduce the risk and bridge the gap between business and technical requirements.

Other goals

- Show us the clear structure of the system, but hide all implementation details.
- Give an overview of all use-cases and scenarios.
- Address the different requirements for the different stakeholders.
- Show the clear subcomponents and their relationship in the system.
- Bring all team members on the same goal and reduce the goal of ownership.
- Clearly showcase system internal and external components, its help to scale the architecture.

Limitation with software architecture

- Lack of tools sometimes tools not fit with each other
- No standardize of the design or represent architecture
- Lack of the standard analysis method to predicts that current architecture full fill all requirements
- Lack of awareness how to implement the architecture itself
- Lack of understanding the requirements from stake holder by the architecture
- Lack of understanding design, analysis process for all architecture itself

Importance of design expertise for the software architecture.

It is very important software architecture have a clear view of design as the full system. A person needs to be expert in software design with different approaches, architecture design must full the integrity of the system design. Architecture design needs to be flexible for the scaling of the system.

4.3.2 Scope lean software architecture design

With software architecture, it is important to have a software test architecture for the testing purpose [7]. Effective test architecture will bring the fast results and easy to maintain in future. With the growth of cloud technologies, it's important to re think now how we can adopt the lean principles in the lean software test architecture. Lean architecture helps to build and deliver clear, consistent, complete set of goal-oriented test architecture.

Lean architecture could help us to bring the

- Simplified concept of system design
- System designs with least number of layered decomposition
- Build the architecture design customer value focused
- Identify the architecture level waste and add value stream
- Improving architecture consistency and flow with other subsystems
- Change or Improve the architecture based on demand

- Improve the architecture with continuous improvements
- Support the different test tools and programming languages

Lean architecture business agility

- Moving software application architecture from one data center to another in term of the upgrade, maintenance and cost effective.
- Improving dynamic allocation of the servers and storage so it provides the platform to add and run other related and sub comments in the smooth way.
- Managing the complex computing scenarios through different dynamic allocation such as locally stored vs stored in cloud web.
- Easy to manage the remote servers and applications who adopted the lean test architecture

Lean test architecture design considerations

- Virtualization
- Storage Virtualization
- Network Virtualization
- Service Orchestration
- Tools and Technology Support

Lean architecture is able to bring the number of opportunities and advantages than another way of architecture design process. Lean also eliminate the waist in the architecture level design and process. The growth of technology also increases the complexity of the development environment in such situation it is very important we think about lean principle adoption in development and test architecture design process.

5. Scope lean software test architecture design

As we noticed above we have already data from lean architecture design optimization, now can use that data to build own system architecture design.

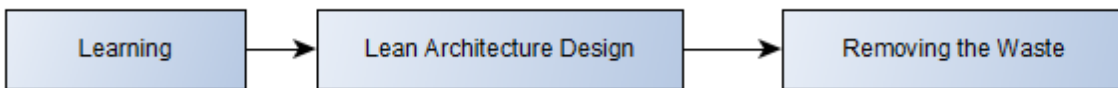


Fig 5: Steps for lean test automation architecture design

Table 1: Indemnification of the lean test automation architecture design components

Learning	Lean architecture design	Removing the waste
System components	Use lean canvas	Identifying un used system component
Automation Frameworks		Identifying us used framework and libraries
Testing tools		Identifying un used testing tools
Domain knowledge		Identifying best fit knowledge
Types of tests		Identifying non automated test cases

6. Conclusion and future research

The above study explains the all new possible approach for the investigating more adoption of lean canvas in software testing architecture design to design test architecture in more efficient way.

In order to do more research, it is needed to accomplish the following activities.

- Need to do an experiment to identify the lean test automation architecture design metrics;
- Need to define the test design architecture criteria, that evaluate design architecture;

- To carry out more appropriate investigation lean test automation architecture design;
- Need to study if there is any need of the design tool for the test automation architecture design;
- Need to develop the canvas design tool that collect the all test automation architecture design components;

The new possible approaches proposed in this article are focused on the possibilities of finding that how we can use lean canvas for the test automation architecture design. The authors aspiration is that new approach will generate more new ideas, and some of the new ideas will be extracted from current article. In test automation design architecture, it's an endless search how lean canvas can be adopted in the design process.

References

- [1] Duffy A. H. B. and O'Donnell F. J. "A Design Research Approach. In: Critical Enthusiasm – Contribution to Design Science", 1999, pp. 33-40.
- [2] Goel A. K., et al. Functional Explanation in Design. In: Proceedings of IJCAI-97 Workshop on Modeling and Reasoning about Functions, 1997, pp. 1-10.
- [3] The Engineering Design Process, <http://www.eie.org/overview/engineering-design-process>.
- [4] Nidagundi, P., Novickis, L. Introduction to Lean Canvas Transformation Models and Metrics in Software Testing. Applied Computer Systems. Nr.19, 2016, 30.-36.lpp. ISSN 2255-8683. e-ISSN 2255-8691. Pieejams: doi:10.1515/acss-2016-0004.
- [5] Padmaraj Nidagundi. Int. Journal of Engineering Research and Applications, ISSN: 2248-9622, Vol. 6, Issue 4, (Part - 1) April 2016, pp.13-16.
- [6] David, five principles of lean building design, <http://www.architecturefordevelopment.com/2010/09/five-principles-of-lean-building-design/>.
- [7] Jeff Sutherland, Jean Tabaka, "Incorporating Lean Development Practices into Agile Software Development", System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference, Year: 2007, Pages: 273 - 273, DOI: 10.1109/HICSS.2007.267
- [8] Duties, Skills, & Knowledge of a Software Architect Background, <https://www.sei.cmu.edu/architecture/research/previousresearch/duties.cfm>