PA2552 - Software Testing

Lean Testing Principles

Christoffer Bohman
MScEng: Game Engineering
Blekinge Institute of Technology

Contact:
chbh22@student.bth.se / student@skyh1gh.dev

Revisions and git history:
GitHub

February 10, 2026

Notice of AI usage:

Scopus "AI Query Builder" provided by Blekinge Institute of Technology has been used for the purpose of building database queries in search of relevant articles. These queries were manually checked and verified by the author.

1 Introduction

Testing is an essential part in most large-scale projects, whether that would be in industrial manufacturing plants or software development. The practice of ensuring functionality, reliability and safety of crucial mechanisms is a foundational requirement for any kind of development. There are multiple ways of evaluating different types of metrics for a variety of use cases, some wider known types include: stress testing, performance testing and smoke testing [1].

The purpose of this technical report is to collect, analyse and summarise relevant information touching on the subject of lean software testing. More specifically, this report aims to questions: "What lean testing principles are there?" and "How do they relate to testing?"

2 Methodology

The research method employed by this report is to search the common internet for information that may be relevant for the questions outlined in the introduction (section 1). Various data gathering methods have been utilised, namely the Google Search Engine and the Scopus Journal Database have both been used for said gathering. The vetting of the search results was mainly comprised of filtering the contents based on keywords like "agile testing", "lean testing", "software development", "game development" and "software engineering".

3 Results

Lean testing is based on principles that closely align to the commonly known principles and values of agile and lean where the purpose is to maximise efficiency, adapt to change and minimise waste [2]. In addition to this, a core perspective that is a result of the lean way of thinking is the idea that the value of the product along with customer satisfaction should be the central focus [2]

Table 1: Abstract and Concrete Lean Testing Principles

Abstract	Concrete
Provide continuous feedback	Efficiency / Effectiveness
Deliver value to the customer	Minimisation
Enable face-to-face communication	Test data generation
Have courage	Execution
Keep it simple	Maintenance
Practice continuous improvement	Values (-illities)
Respond to change	Purpose
Self-organize	Automation
Focus on people	
Enjoy	

Table 1: Tabularised form of the reinterpreted lean testing principles (not lean principles) outlined in the PowerPoint slides 6 & 9 written by Alégroth and Nass. [3]

A three-way case-study between waterfall, agile and agile with dynamic QC has been carried out that compared the efficacy of testing within these methodologies [4]. The method Tommy et al. [4] employed in their research yielded a result where the method of doing tests at the end of a production cycle (waterfall) generated a detected defect count of slightly above twenty. In contrast, agile and agile with dynamic QC reached counts of slightly below eighty and around a hundred and ten respectively. In addition to this the test case count is around one hundred for waterfall, one thousand two hundred for agile and nine hundred for agile with dynamic QC [4].

4 Analysis

The results of my data collection show that lean testing adds an additional layer to the development phase. Instead of pushing the testing to the very end causing a large backlog of tests, they are done continuously with the value of the product and the customer in mind. The data also shows that there is a gap in efficacy between waterfall and different implementations of agile and lean testing, showing it is more beneficial to test continously. Dynamic QC as defined in [4] furthers this effect by adding testing as a main focus of a project.

5 Discussion

At a broader glance, this report not only shows that lean testing can be boiled down to abstract and concrete principles, but also that the lean testing methodology can have a much greater impact when combating functional defects in projects. What are the implications of these findings? It can can be argued that the importance of these principles provide a general and usable framework that speeds up the process by for example extracting the purpose of the testing. This can be further linked to the ideas of practicing continuous improvement through test assessment and usability. Additionally, by keeping it simple, a layer of assessment criteria can be derived that try to weed out unnecessarily large and complex tests that don't perform as good as they potentially could.

These patterns can be recognized in, for instance, unit testing. A unit test, as commonly known, should be small and simple in order to test a smaller unit of a large codebase. Keeping it simple is fairly self-explanatory as unit tests should encompass simpler units and problems. However, depending on the team that has been organised, the approach to the writing of the tests may be varying. In our small game test, our approach to unit testing was the usage of the Google Test framework where each person each had the responsibility of covering their own code. This way of testing required us not only to enable face-to-face communication, but also forced us to have courage and put the breaks on the project if the tests didn't satisfy our definition of done. The result of this was having to respond to change as sometimes test cases broke, and certain discussions had to be held on how certain mechanics should work and why.

6 Conclusions

In summary, lean testing is a testing equivalent of the lean methodology with principles that are based around customer satisfaction and value generation. Lean testing does not necessarily imply a certain type of tests, but rather the way of working with tests. Some argue that this way of interacting with testing not only increases defect detection, but may also lead to fewer, more precise tests to be maintained.

Some considerations to take into account may be the increased workload of continously executing, adding, changing and updating tests to a codebase which in turn could also impede progress in a few situations. The general idea though, is that with the right design, architecture and a mindset to reduce waste, these disadvantages will diminish considerably.

References

[1] S. Susnjara and I. Smalley, “What is Software Testing? — IBM — ibm.com,” https://www.ibm.com/think/topics/software-testing, [Accessed 2026-02-02].

[2] R. Jain, “Lean Testing — What it is, Principles & How to Do? — testsigma.com,” https://testsigma.com/blog/lean-testing/, 2024, [Accessed 2026-02-02].

[3] E. Alégroth and M. Nass. Pa2552: Software testing. Blekinge Institute of Technology. [Accessed 2026-02-02]. [Online]. Available: https://bth.instructure.com/courses/7004/files/1496336?wrap=1

[4] R. Tommy, M. Mhaisekar, S. Kallepally, L. Varghese, S. Ahmed, and M. D. Somaraju, “Dynamic quality control in agile methodology for improving the quality,” 2016, Conference paper, p. 233 – 236, cited by: 4. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84966570081&doi=10.1109%2fCGVIS.2015.7449927&partnerID=40&md5=02d52ded3d20e725332b0d62b43060af