

# AGILE TRANSITION WITH MORE FREQUENT DELIVERIES AND HOW THAT AFFECTS TESTING

Göran Kero<sup>1</sup>

<sup>1</sup> ICT Services and System Development (ITS), University of Umeå, S-90187 Umeå, [goran.kero@umu.se](mailto:goran.kero@umu.se)

## Keywords

Continuous delivery, Agile, Test

## 1. SUMMARY

One central part of transitioning to agile software development is being able to (and being allowed to) more frequent deliveries. This creates both great opportunities but also some challenges.

In the Ladok3 project, the current frequency of deliveries to production is every other week. This can be compared to the current Ladok system where new deliveries have been made four times a year.

This paper focuses on how more frequent deliveries affect the challenge of testing the system properly. It begins with a description of the concept of continuous delivery and how it affects testing, followed by an experience report on how the Ladok3 project meets this challenge, both how it has been met until now as well as challenges seen for the future.

## 2. INTRODUCTION

Ladok is a national student information system for higher education in Sweden, owned by 38 universities and university colleges through a consortium.

The ongoing Ladok3 project, where the next generation of Ladok is being developed, is a large agile project that uses Scrum as development method. The project consists of product owners and domain experts from many of the universities and university colleges and development teams located at Umeå University, ICT Services and Systems Development (ITS).

One big change made in the development process of the new Ladok system, within the Ladok3 project, is an increased frequency of delivery compared to the existing Ladok system that is being replaced.

## 3. TESTING WHEN FREQUENCY OF DELIVERY IS INCREASED

Increased frequency of delivery does impact how testing can and should be done.

### 3.1 CONTINUOUS DELIVERY

The concept of continuous delivery has been a central part of the agile movement ever since the Agile Manifesto was published. The first of the twelve principles behind the Agile Manifesto (2001) says:

“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”

The main benefit from continuous delivery is the early feedback that is given by the use of implemented features in production very shortly after it is written by the developer. The possibility to fix bugs and make changes quickly based on the feedback received is also enhanced and new releases be delivered to production with very short delay.

### 3.2 FREQUENCY OF DELIVERIES IN LADOK

The current Ladok system that is being replaced with the new system developed within the Ladok3 project has had a delivery frequency of four times a year, with patches added when needed in case of emergency. This is what the universities and university colleges are used to. It is also a matter of accepting a new release as well as be aware of changes made in the new delivery. Striving towards

more frequent deliveries does not only affect the Ladok3 project and operations but also the end users and system owners and their business. This is also a great challenge, to help the stakeholders get comfortable with more frequent deliveries.

### 3.3 HOW DOES MORE FREQUENT DELIVERIES AFFECT TESTING?

One major challenge when transitioning to agile development methods and more frequent deliveries is how the system can be tested. It is obvious that it does not work to have a test phase in the end of the development cycle like in a traditional waterfall project, there is just not time enough for it. The testing puzzle must be solved differently.

### 3.4 THE NEED FOR AUTOMATION

Making complete (manual) system tests every release is nearly impossible in every software project, but when the delivery frequency increases to every other week or higher, it is not even worth trying. One solution very popular in Agile methodologies is through increased automation of testing. This, though, must also be handled with care since creating automated test not only takes development time but also a lot of maintenance time since changes to the code affects existing tests.

### 3.5 THE AGILE TEST PYRAMID

Mike Cohn introduced the agile test pyramid in his book *Succeeding with Agile* (2009). The pyramid is an illustration on how test automation should be distributed between the layers of software, with more emphasis on the low (unit) level and less on the higher (e.g. user interface) levels.

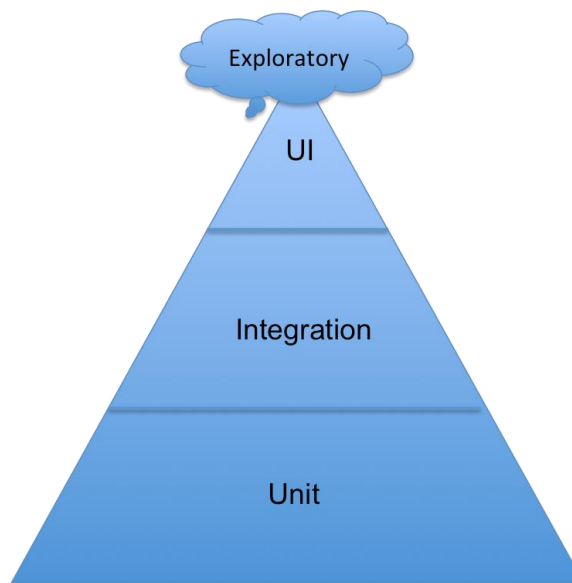


Figure 1. The test pyramid

Unit tests are the least expensive type of tests and the speed of running them is very high. Automated tests at the user interface level is much slower to run and they also cost a lot more to build and maintain. Tests on higher levels also mean increased amount of possible combinations to test which makes it very hard to keep a good test coverage. The same philosophy of test automation is also used in the concept of TDD (Test-driven development).

Manual testing is even slower but can be a good complement to the automated tests since it can add other kind of values that automated tests can't, since human users can identify other types of issues that could possibly be problems. In the Ladok3 project, this philosophy is adopted.

### 3.6 EXPLORATORY TESTING AS A COMPLEMENT

With the automated tests being the testing base, the Ladok3 project uses exploratory testing as an addition in order to uncover other kinds of bugs and risks, like user experience (UX) issues.

Exploratory testing is an approach that has become increasingly popular during the last decade. Compared to the traditional scripted test methodologies, exploratory test focuses on human skill and ability to learn instead of processes. Elisabeth Hendrickson (2013) defines Exploratory testing as

“Simultaneously designing and executing tests to learn about the system, using your insights from the last experiment to inform the next.”

In the Ladok3 project, exploratory testing is performed within the development teams, by the developers and testers included in the teams. Using Scrum as a development method, with a sprint length of two weeks, the goal is to be done with all exploratory testing necessary within the sprints, which means no test phase after a sprint is finished. This results in that the software should be possible to release at the end of every sprint.

### 3.7 A CHANGED MINDSET AND A LOOK INTO THE FUTURE

One of the greatest challenge when moving towards more frequent deliveries is to change the mindset, both within the project as well as all stakeholders outside the project. Agile software development with frequent deliveries requires trust and a robust system that can be released frequently with high quality. Being able to trust the code base with its automated tests together with quality built in through the development process is a prerequisite of frequent deliveries. At the same time, delivering more frequently does also mean smaller increments of software each delivery, which means less risk of breaking the software. Making the delivery process faster also means that bugs in production can be fixed much faster and a new released deployed into production.

Increasing the delivery frequency is an ongoing work with the new Ladok system, where every part of the process is being trimmed to make that possible, including the development team processes, the continuous integration environment and the operation- and deployment processes. It is also a process of helping both project member and other stakeholders get comfortable with more frequent deliveries. Perhaps in the future it will even be correct to call it continuous delivery. One major key to being able to deliver more frequently is to gain trust about the concept by delivering working software.

## 4. CONCLUSION

Increasing the frequency of delivery to production does affect testing a lot. It makes it more important that quality is built in during development, especially with a good base of unit tests, since the time for testing after development is decreased and of less value. At the same time, the feedback loop is shortened, making it possible to discover issues earlier and deliver fixes more rapidly. It is an ongoing challenge for the Ladok3 project to be able to deliver new software more frequently.

## 5. REFERENCES

Agile manifesto website (2001). Principles behind the Agile Manifesto. Retrieved March 1, 2017, from <http://agilemanifesto.org/principles.html>

Hendrickson E (2013). *Explore it! Reduce Risk and Increase Confidence with Exploratory Testing*. Dallas TX: Pragmatic Bookshelf

Cohn, M (2009). *Succeeding with Agile*. Addison-Wesley Professional.

## 6. BIOGRAPHY



Göran Kero studied Computer Science and Economy at Luleå University of Technology and holds a bachelor degree in Economy and Computer Science. He has over 15 years of work experience as a tester and software developer, first in the telecom business, Mobilaris AB and Telia Mobile, for 3 years and since 2004 at Umeå University, ICT Services and Systems Development (ITS). In recent years his primary focus has been on quality, test methodologies and software development processes, with a role as Senior Test Architect in the Ladok3 project. Göran is a regular speaker, with experience from local, national and international events, including EUNIS 2014 Congress.