POOR QUALITY

CASE STUDY

20

Huitale – A Story of a Finnish Lean Startup

Marko Taipale

Huitale, Hiilikatu 3
00180 Helsinki, Finland
marko.taipale@huitale.com

Abstract. We in Huitale have implemented a lean product development process. As a result Huitale has a workflow that is predictable within acceptable variance. We can change the direction of the business at any given time but stay grounded in what we have learned. We can adjust the product roadmap visibility according to our business needs. In addition we have achieved exceptional quality. In past three years we have had two production bugs. The implementation of lean development process requires discipline and experience.

Keywords: lean product development, agile software development, Kanban, Scrum, eXtreme programming, lean startup.

1 Summary

There is on-going buzz about lean startups [1], agile and lean software and product development in the industry [2]. However there is very few descriptions how software product companies have applied the lean principles in order to match the demand for their product and improve the development lead and cycle times. We in Huitale have implemented a lean product development process. For past few years we have been able to predict the lead times for new features and adapt our business model according to the market needs [3]. In addition we have built capability for releasing our software everyday for thousands of consumers with extraordinary quality: we have had two bugs in past three years. We have noticed that it takes discipline and experience to implement continuously improving process.

2 Background

Huitale Ltd is a provider of lean and agile consulting and a product development company. Last three years Huitale has been developing a national consumer portal Nextdoor.fi to enable individuals and companies to buy and sell house holding services online. Currently the portal is having over 30 000 unique visitors every month and 2 000 active users. The people working for Huitale have several years of experience in agile software development, lean product development, Scrum, Kanban and eXtreme programming practices and the company is continuously improving its processes by experimenting new ways of working.

P. Abrahamsson and N. Oza (Eds.): LESS 2010, LNBIP 65, pp. 111–114, 2010.

3 Product Development

Our product development process [3] starts by a signal from development saying that there is only two Minimum Marketable Features [4] left in product development queue. This signal initiates a discovery of three themes from the customer development [5] channels, competition analysis and internal innovation. We spent two hours time box on figuring out which themes are worth of investing the upcoming month. Once the themes are discovered we start working on features by brainstorming them for each theme. We limit the number of features to seven in order to keep the buffer small but still big enough to ensure the flow for next month. The brainstorming session including formulating of MMFs takes two days and it ends by putting the items into Product Qeueue in prioritized order.

Once we have the items in the queue and there is room in the downstream for upcoming work we start defining the highest priority MMFs as READY. The definition of ready is:

- No immediate blocker for developing the MMF
- If MMF has impact on language content the content is available. It might not be final.
- If MMF has impact on look & feel the content is available (GUI layouts, graphics etc.). It might not be final.
- If MMF has impact on emails sent to the users the email templates are available. They might not be final.
- If MMF has impact on reports the expected changes are communicated.
- If MMF has impact on third party services/integrations the expected changes are listed. List might (and usually is not) final.
- We have set measures for MMF to decide later on if we should keep the feature or drop it

At any given time we can have two items defined READY to be pulled by to development. The Work-In-Process limit for development is also two. Developed items must pass our Definition of Done to get done. Our definition of done is:

- MMF can be released
- code has unit tests and automated acceptance tests [6] that execute successfully
- code passes tests in CI environment
- code is refactored and the design is simple
- code passes automated QA checks (checkstyle, pmd/cpd & whatnot)
- new feature is documented (if applicable, sometimes for third parties)
- peer review is done (if applicable, sometimes for critical paths)

Once a MMF is done it will be released to production within next 24 hours with help of our Continuous Deployment [7] environment.

4 Teams

We have moved from single Product Owner model to a Problem and Solution team model as part of process improvement. Problem team is lead by CEO and consist

CTO, Sales and Marketing Director and User Experience Designer. Problem team uses tools like customer development, business reports and metrics to find out what problem we should be solving for our customers and users. Solution team is lead by CTO and consists of developers and User Interface Designer. There is no separate "testers" or "operations". These roles are carried out by the solution team members as part of their development roles. The mission of solution team is to build whatever required by the problem team and provide insights, data and feedback to the problem team and customer development process.

These teams overlap for a reason. The idea is to avoid communications via documents and encourage rich communication between all the parties. We have also involved subcontractors for various roles in the teams. Both the teams are at least somewhat distributed.

5 Results

Our workflow is predictable within acceptable variance and we can change the direction of the business at any given time. We can adjust the product roadmap visibility according to our business needs.

We have continuously done daily releases of Nextdoor.fi to production for past three and half years. The public beta consists of minimum set of features and was developed in 120 man days from which 70 was used to developing the features and 50 for user interface. The environment for Continuous Deployment and operations were developed during the product development incrementally by the solution team. The lead time from idea to production is currently 8 days on average. Smallest MMFs take less than 3 days to pass through the workflow.

The site has now over 30 000 unique visitors per day, over 2 000 active users and it has proven to deliver value for its users. Nextdoor.fi is still released to production every day. The production system has 24/7 system and application monitoring with daily backups and automated system rollbacks in case of unsuccessful deployment. The business reports and metrics for the whole and each feature are generated on every 24 hours.

Continuous Deployment and Integration environment executes over 550 automated acceptance tests through the Graphical User Interface to the database and integration points. The overall test coverage is over 80% for all the code including custom maintenance tools and over 85% for all the application code. During past three years we have gotten two production bugs from which one was reported by the users and both were reported by the application monitoring. Both bugs were fixed within less than one hour from the time they were reported.

We have no employees or owners working on operations nor do we have single tester.

6 Lessons Learned

According to our experience discipline is the key factor on executing, implementing and improving the lean product development process. Visualizing the workflow and measuring both lead time and cycle times is very beneficial on finding subjects for improvement. Applying tools like root cause analysis [8] helps finding actionable issues. For a product company it is also important to have mechanisms to validate the product vision and have metrics to figure out if your product is actually valuable to anybody.

References

1. Ries, E.: What is lean about lean startup, `http://www.startuplessonslearned.com/2009/12/what-is-lean-about-lean-startup.html`
2. Wikipedia: Lean Software Development, `http://en.wikipedia.org/wiki/Lean_software_development`
3. Taipale, M.: The Huitale Way, `http://huitale.blogspot.com/2010/03/huitale-way-our-value-stream-map.html`
4. Scotland, K.: The anatomy of an MMF, `http://availagility.co.uk/2008/07/07/the-anatomy-of-an-mmf/`
5. Blank, S.: Customer development for web startups, `http://steveblank.com/2010/02/25/customer-development-for-web-startups/`
6. Koskela, L.: Acceptance TDD Explained, `http://www.methodsandtools.com/archive/archive.php?id=72`
7. Ries, E.: Continuous deployment in 5 easy steps, `http://radar.oreilly.com/2009/03/continuous-deployment-5-eas.html`
8. Wikipedia: Root cause analysis, `http://en.wikipedia.org/wiki/Root_cause_analysis`