

Chapter 57

Improving Lean, Service– Oriented Software Development at Codeweavers Ltd

Paul Shannon
7digital Ltd, UK

Paul Barrett
Codeweavers Ltd, UK

Neil Kidd
Codeweavers Ltd, UK

Chris Knight
Codeweavers Ltd, UK

Sam Wessel
Esendex Ltd, UK

ABSTRACT

Following a successful adoption of lean and agile practices, the development team at Codeweavers Ltd has furthered its approach to service-oriented software development for the motor finance and insurance industry. Through iteratively inspecting and adapting their processes over the last twelve months, the team members have seen a change from their Kanban-style single piece flow to multiple work cells developing with separate swim lanes on a work in progress board and within fixed length iterations. The arrival of strong competition to their market led to a positive shift towards customer service and interaction with increased attention on lean planning and agility. This chapter reports on improvement in software craftsmanship with a focus on quality, largely achieved by the use of service-oriented architecture, combined with increased use of mocking for unit-testing. The perspective taken is from software team members, and in the chapter, the developers chart their own observations, improvements, and failures over the course of a year.

INTRODUCTION TO CODEWEAVERS AND THEIR AGILE ADOPTION PROCESS

Codeweavers Ltd is a software solutions provider based in Staffordshire, UK. The company has around twenty-five employees, including a development team of around twelve developers.

Smaller teams of two or three are responsible for infrastructure, design, sales and client support. Since 2001 the company has delivered applications and web services for the automotive financial and insurance industries, and in that time has grown to become the market leader in its specialised field. Its client-base includes leading car manufacturers and dealer groups.

In late 2007 the company began a gradual journey of agile process adoption, in response

DOI: 10.4018/978-1-4666-4301-7.ch057

to a chaotic working environment and a period without profit. At this time, developers at Codeweavers were divided into two teams of four, working on separate applications with specialised knowledge and no shared code base. Each developer worked alone from his own to-do list. Planning and development were unpredictable, code quality was poor and deadlines were often missed. Almost half of the developers' time was devoted to fixing bugs and making small, ad hoc changes.

In response to these problems, the teams adopted Scrum. They each compiled their to-do lists into a backlog and began using planning poker to estimate tasks. They began working to fixed-length sprints, holding daily stand-up meetings and weekly retrospectives. They also began implementing Test Driven Development and Pair-Programming in an attempt to raise code quality. To assist with these changes, the company hired an Agile Coach, who introduced work in progress boards to replace sprint management software, and a continuous integration server to automate builds and unit test runs.

As a result of the improvements to planning and development gained from these actions, the developers' focus shifted to reducing specialisation and knowledge silos. This was accomplished by merging the two teams together and swapping pairs more frequently, which helped to share knowledge across all the developers, and helped the whole team to agree on common standards and domain language.

By mapping the value stream through the company the team was able to identify bottlenecks on which they could swarm. Responding to one such bottleneck in deploying releases, the developers began grouping tasks into marketable features of predictable sizes. These were specified with criteria for their acceptance and only deployed when all developers were satisfied with their quality. These changes noticeably improved the quality and speed of development.

A brief bottleneck in development led the team to begin recording causes of wasted time, increasing the frequency of stand-up meetings and encouraging developers to stop the whole team when a problem was encountered. This action increased communication, reduced downtime, and improved the flow of value through the business.

By Spring 2010, the business was transformed both financially and culturally. Codeweavers Ltd was a vastly more relaxed, sociable and productive environment, with a self-organising development team and a healthy profit from regular feature releases with dramatically fewer bugs. Details on the company's transformation from chaos to Kanban via the Scrum, and the approach used, are provided in Rutherford et al. (Rutherford et al., 2010).

FURTHER IMPROVEMENTS IN SOFTWARE CRAFTSMANSHIP AT CODEWEAVERS

Codeweavers' Understanding of Agility and Leanness

Codeweavers, and in particular the development team, experienced a positive feedback loop by reporting on the way the company inspect, adapt and continually try to improve their practices and processes.

This chapter emerged in the same vein and reports on further improvements, new observations and additional changes to the software development process. It covers the overall increase in code quality, and the changes to practices that helped foster these improvements.

Codeweavers Ltd arrived at the following understanding of agility Highsmith(Highsmith, 2002):

Agility is the ability to both create and respond to change in a turbulent business environment. Agility is the ability to balance flexibility and stability.

12 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/improving-lean-service-oriented-software/77752

Related Content

System Integration Using Model-Driven Engineering

Krishnakumar Balasubramanian, Douglas C. Schmidt, Zoltán Molnár and Ákos Lédeczi (2009). *Designing Software-Intensive Systems: Methods and Principles* (pp. 474-504).

www.irma-international.org/chapter/system-integration-using-model-driven/8246

A Multi-Methodological Approach to Study Systems Development in a Software Organization

Paivi Ovaska (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 2285-2306).

www.irma-international.org/chapter/multi-methodological-approach-study-systems/29506

Self-Modifying Code: A Provable Technique for Enhancing Program Obfuscation

Chandan Kumar Behera and D. Lalitha Bhaskari (2017). *International Journal of Secure Software Engineering* (pp. 24-41).

www.irma-international.org/article/self-modifying-code/201214

Experiences with Cloud Technology to Realize Software Testing Factories

Alan W. Brown (2013). *Software Testing in the Cloud: Perspectives on an Emerging Discipline* (pp. 1-27).

www.irma-international.org/chapter/experiences-cloud-technology-realize-software/72224

VORDi*: A Combined Approach for Requirements Engineering of Software as a Service

Farida Kherissi (2021). *International Journal of Information System Modeling and Design* (pp. 1-20).

www.irma-international.org/article/vordi/288553