

From Agile to Continuous Development in the Healthcare Domain - Lessons Learned

Extended Abstract

Walter Stocker

Siemens Healthineers

Forchheim, Germany

walter.stocker@siemens-healthineers.com

ABSTRACT

Starting in 2006 our organization in the healthcare domain began a shift towards agile-like development rather than a strict interpretation of the "V-model". But this was only the beginning of an ongoing journey. Today modern software engineering is characterized by much faster development cycles than those used in traditional models. Our business demands that we shorten release cycles and support many different deployment scenarios. Due to the nature of our products, it is also necessary to address the relevant regulatory requirements which make this task even more challenging.

Over the years, we have made a number of improvements in the spirit of "continuous improvement" that have helped us reap the benefits of a more continuous way-of-working. These include organizational, architectural, and process-based improvements as well as explicitly focusing on the necessary culture in the organization. This experience report is a summary of our learnings and hopefully will provide something for everyone that helps on their own journey.

KEYWORDS

Agile, Continuous Delivery, DevOps, Software Architecture, Best Practice Sharing, Software Engineering

ACM Reference Format:

Walter Stocker. 2018. From Agile to Continuous Development in the Healthcare Domain - Lessons Learned: Extended Abstract. In *ICSE-SEIP '18: 40th International Conference on Software Engineering: Software Engineering in Practice Track*, May 27-June 3, 2018, Gothenburg, Sweden. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3183519.3183552>

1 INTRODUCTION

Siemens Healthineers has several thousand software engineers worldwide and does software development in a very broad range of, mostly embedded, software-intensive systems in the healthcare domain. As part of this company our department delivers Computed Tomography (CT) post-processing applications (see Figure 1) to support doctors in their diagnosis based on images created by CT scanners (see Figure 2).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ICSE-SEIP '18, May 27-June 3, 2018, Gothenburg, Sweden

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5659-6/18/05.

<https://doi.org/10.1145/3183519.3183552>



Figure 1: CT Post-Processing Application (Example)



Figure 2: Computed Tomography Scanner (CT)

2 FROM AGILE TO CONTINUOUS DEVELOPMENT

In 2006 we decided to do a rather disruptive change. Up to then all of our applications were based on C/C++. This changed significantly when we created a platform implemented in a new technology

(.NET C#). In addition, there was also the wish to deliver more value to our customers and improve their experience with the software. Therefore new user interface concepts were added and everything was directed towards supporting a continuous way-of-working centered on the needs of the customer. This disruptive change triggered an ongoing journey from a very strict interpretation of the "V-Model" towards a more agile-like approach.

Achieving agile for products that are in a regulated environment is significantly more challenging, due for example to the importance of maintaining the high quality attributes, adherence to relevant standards and norms, approach to get regulatory approval, and mix of hardware, software, and services. The talk will outline some of those challenges we have faced and describe the experiences we had.

In the beginning there was no huge upfront plan to become fully agile. With the changes in technology and features it was not possible to develop software like before. We started by approaching each single problem and improving the way we work in a step-wise manner. This habit of continuous evolution is still in the DNA of our department.

The talk presents several examples of those improvements in different areas: built in resiliency (safety-net), architecture, code base and process.

The initial trigger for the first improvements was the need for a better safety-net when it comes to the integration of the software. Therefore we started by building in resiliency with automation (night-run tests) and several guidelines. In addition, an automated process of a stacked system-integration was implemented. When it comes to architecture we worked on making it an integral part of our software development. This included advertising the importance of strategic design and architecture to project- and product-managers. Nowadays the architects have their own resource corridor that can be used for architectural work. In the code base we introduced different improvements starting from XP-practices (like code review, pair programming) and clean code to the introduction of component responsibilities. We also aimed to change the process from a strict V-model towards a more agile-like approach.

In the future we strive to introduce continuous development aiming towards faster releases and getting closer to a DevOps approach. This will be quite challenging because it includes the whole organization and has huge cultural impact. For us this is the next logical step to deliver even more value to our customers. The talk will also present this part of the journey where we have just started.