

A Case Study of the Implementation of Agile Methods in a Bioinformatics Project

Xueling Shu¹, Andrei Turinsky², Christoph Sensen², and Frank Maurer¹

¹ Computer Science Department, University of Calgary,
Calgary AB T2N 1N4, Canada
{shu,maurer}@cpsc.ucalgary.ca

² Sun Center of Excellence for Visual Genomics, University of Calgary,
Calgary AB T2N 4N1, Canada
{aturinsk,csensen}@ucalgary.ca

Abstract. From July 2005 to August 2006, a bioinformatics project experienced a substantial transformation by adopting Scrum and some XP practices. The paper reveals project risks, previous challenges faced by the team and results from this one-year exploratory case study. The paper presents a discussion of the lessons learned from the perspective of both the project manager and the on-site agile advisor, and recommendations on speeding up the adoption process for other projects.

Keywords: Agile Methods, Lessons Learned, Java 3DTM, Bioinformatics.

1 Introduction

Ethnography recommends collecting data from “participant observations, interviews, documents, and informal contact with participants” over an extended period [1]. Through embedding agile researchers into a bioinformatics development team during an approximately one-year case study, we developed a deep understanding of the implementation strategies for agile methods from long term observations.

The observed team has been developing a software tool that allows medical researchers to view a complete image of disease mechanisms in a 3D environment. Several project risks existed: technical novelty, limited resources, ineffective interpersonal communication, and reluctant user support. One of the goals of the project is *commercialization* of leading edge bioinformatics research software. Therefore, high software quality and solid management practices are essential.

Some of the earlier projects in the lab faced various challenges. Developers struggled with debugging of unstable research prototypes that focused on proof-of-concept demonstrations, facing substantial requirement churn. Acceptance of some papers that presented new software tools was delayed when peer reviewers tested the software and found bugs in it. Overall, the management was not satisfied with progress and software quality.

2 Result

The adoption of Scrum and XP practices brought substantial improvements in two aspects: management process and intra-team communication. User stories now

describe features more clearly. Developers pick the stories by themselves rather than being assigned the stories, which helps the team form a self-organization culture. Test automation saved considerable amount of effort and improved software quality. Product backlog facilitated the management of specification obsolescence. Story-oriented meeting style improved the meetings efficiency considerably. Task coordination became a team-wide effort. With the problems being cleared away step by step, the morale has also improved both professionally and personally.

3 Lessons Learned

The on-site agile advisor must be a proactive gentle(wo)man. Disagreements continued throughout the duration of the project. The resolution lies in being a proactive listener with patience and actively yet carefully analyzing the environment.

The on-site advisor should try to benefit from her technical expertise. Demonstration of skills tells people that you really can help them solve problems and build up the team confidence in the advisor's suggestions.

"Down, and touch the ground". Benefits of a new idea will not be accepted easily by simply talking about theories from books. For example, we showed the benefits of test automation by actually writing test code for the project. After the lab wide seminar on test automation, some developers either started coding tests for their own project or were willing to learn how to write the code.

The project manager should focus on most tangible agile improvements, especially at the beginning. One of our biggest concerns was to ensure that the suggested changes are both timely and relevant to the project deliverables, and are demonstrably valuable. In our case, these turned out to be the introduction of automated development-support tools for automated testing, Java code-standard checking, and iteration planning by the on-site agile advisor.

The project manager should try to pair the agile advisor with another developer. Such pairing may overcome the steep learning curve for the advisor, and also broaden the social support base for the advisor within the team, making subsequent changes easier to introduce.

The biggest problem we observed is the advisor was considered "just a student", instead of an outside expert. For other agile practitioners, we would strongly recommend start higher-up in their teams: add a process consultant as an advisor or as a PM so that the power structures are actually set up to impact a change faster.

Reference

1. Myers, M.: Investigating Information Systems with Ethnographic Research. *Communications of the AIS*, vol. 2(4) (December 1999)