

Repo Margining System: Applying XP in the Financial Industry

Piergiuliano Bossi
Banca IMI S.p.A.
+39 02 77512104 Italy
pgbossi@hotmail.com

Francesco Cirillo
XPLabs S.R.L.
+39 0761 659037 Italy
francesco.cirillo@xplabs.com

ABSTRACT

In the financial industry, the ability to rapidly develop software flexible enough to follow a continuously evolving business is a must. The existence of a work environment which is usually stressful prevents developers in this field from delivering quality software on time and within budget. Like gold prospectors in the West, our team – named Klondike after the legendary Gold Rush – searches for the vein: developing “*good, fast and cheap*” software in an enjoyable way. This paper shows significant results from an actual XP project.

INTRODUCTION

The Repo Margining System was chosen as a project during the Sun Education “*SUNBIM*” Program due to its limited impact, relative complexity and the team’s poor domain knowledge (typical in financial engineering software development).

PROJECT OVERVIEW

The system allows a trader to request, via browser, a series of reports regarding risk connected to repo portfolios of various kinds and to evaluate margins that must be observed by contracting parties. Through the capture of relevant quotes, the system provides margin determination based on proprietary formulas.

Project Organizational Structure

The team is made up of 7 people. The work area, an open space, was rearranged in such a way as to make team communication more effective. The project is being developed in Java, and the team is using a class browser, a refactoring tool and a tool for automated testing. Moreover, the team has developed: a framework for the implementation of blocks; testing, metrics and integration tools; a worksheet for tracking and planning.

XP PRACTICES

The following practices have been fully applied: The Planning Game, Small Releases, Metaphor, Simple Design, Test-First, Refactoring, Pair Programming, Collective Ownership, Continuous Integration, Coding Standards, and Open Workspace.

The customer has not been on-site, but always available on request.

As for the 40-Hour Week, from a 50 hour average workweek at the beginning of January 2001, the team has now gone to 45 hours a week, 14 hours of which are devoted to study.

LESSONS LEARNED

Refactoring – Significant attention (on average, 50% of development effort) has been devoted to increasing

system simplicity and communicativeness. This has allowed the team to implement changes or new functionalities required by the customer - while maintaining the same high level of quality - in less time and at lower cost than otherwise thought possible.

Pair Programming – This practice has almost eliminated the introduction of system defects (1 defect observed out of 4 iterations) and has produced shared knowledge of the system within the team.

The Planning Game – The team has noticed that the customer is very much focused on the delivery of value. The customer was satisfied even when the team delivered part of what had been committed.

Tracking and planning – Tracking on the basis of 30 minute periods has provided concrete and continuous feedback, and has offered the team a way to learn how to make better estimates.

Metrics – Measurement of certain variables (such as the progression of testing and application methods, the ratio of the number of delivered source instructions to the number of classes, etc.) has, among other things, disclosed and/or confirmed the need for refactoring.

Testing – The availability of a complete test suite has given the team the confidence to make changes to the system.

RESULTS

So far, the team has completed 4 iterations in 7 weeks. The system went into production at the end of the 3^d iteration and it has been released twice.

The goal of developing “*good, fast and cheap*” software while having a good time has not yet been achieved, but the team has taken a significant step in that direction. At the moment the team feels it’s developing good software while working on the project with pleasure and enjoyment. The team has also improved its estimating and control capability. In addition, in 3 iterations out of 4, all the required user stories have been delivered, in all cases without overtime. Lastly, the positive effect on cost reduction deriving from the minimization of defects introduced in the system, although not measurable at the moment, is expected by the team. Nevertheless, the team still feels that it’s not that fast and therefore not that cheap.

Having said this, the team believes that by acquiring experience, and enhancing its skills and tools, it will also become fast and cheap.