Improving Responsiveness, Bug Detection, and Delays in a Bureaucratic Setting: A Longitudinal Empirical IID Adoption Case Study

Caryna Pinheiro, Frank Maurer, Jonathan Sillito

University of Calgary
Calgary, Alberta, Canada
{capinhei, frank.maurer, sillito}@ucalgary.ca

Abstract. This paper empirically studies a group of projects in a large bureaucratic government agency that adopted iterative and incremental development (IID). We found that a project that followed IID since inception provided substantially better bug-fixing responsiveness and found bugs earlier in the development lifecycle than existing projects that migrated to IID. IID practices also supported managerial decisions that lead to on-time & on-budget delivery.

Keywords: Iterative and Incremental Development, Longitudinal Study, RUP.

1 Introduction

With the ascendance of Agile Methods it has become hard to find articles related to process improvements without hitting the words "Iterative and Incremental Development" (IID). Incremental means "add onto" and Iterative means "refine" [1].

This paper reports on a case study that was conducted to understand the adoption of an IID approach by a group of IT projects in a large bureaucratic government agency. The improvement efforts were motivated by concerns identified by the business clients of those projects, primarily: poor bug-fixing responsiveness and delivery delays. The focus of this work was to understand how effectively the process changes have been able to deal with these concerns. To this end, we explore the practices in the context of a new project in comparison with existing projects that migrated to IID, both quantitatively and qualitatively.

An industrial investigation of IID practices merits examination for various reasons. IID is often seen as the innovative response to traditional engineering practices [2]. Many believe that IID practices have hit the mainstream of application development [6]. Others believe that *"we're not there yet"* [7]. Perhaps these contradictory views are related to the fact that much of the evidence about Agile adoption has been decontextualized [4]. In addition, Dingsoyr *et al.*'s roadmap [3] calls for more industrial collaboration through Action Research and for more knowledge on how Agile principles, such as IID, work in different contexts. In our research, a governance model faced with strict conformance to rules and regulations resulted in formal decision-making processes. This culture and context led managers

to adopt IID through a methodology with a higher degree of "ceremony and formality", the Rational Unified Process (RUP) [5].

The IID adoption strategy we present in this paper followed a "go all in, but iterate first with some public display" [10]. IID adoption efforts were motivated by several "smells" [9] identified by the business clients of those projects (poor bug-fixing responsiveness and delivery delays). The adoption was top-down and mandated that all projects in the organization had to follow the new RUP practices, but the adoption path for existing projects was limited (discussed in a previous paper [14]).

2 Case Study

We collected data from a suite of three pre-existing complex projects in the largest IT program that migrated to IID practices **(Projects Transitioned)**, and from a suite of four new applications developed using IID practices since inception **(Project New)**. Table 1 presents some organizational context following Eckstein's five dimensions to categorize the largeness of a project [11].

Table 1. Jutta Eckstein five dimensions of largeness.

People	900 employees in total. Ten IT programs with 191 IT professionals.
Scope	Real-time validation of complex government rules and regulations during on-line submission of data, with a user base of over 11,000 users.
Money	The projects under study had budgets of over $1.5 million dollars.
Time	One to three years for a first production release.
Risk	Innovative and unconventional systems. Specially, one of the web applications is considered a "state of the art" government application.

Due to external industrial factors, such as the Alberta oil and gas boom, development was fast paced and many releases were rushed or delayed. Fixed delivery dates forced unreliable acceptance testing schedules on the business clients. IT project teams quickly evolved from small groups of four to six developers to increased teams of over 15 developers. In less than three months the existing projects under study grew to 40+ team members. The result was poor software quality, low team morale, and loss of trust between IT management and business clients. Business clients communicated the need for better quality, better stability, better responsiveness, and more reliable testing schedules to management.

The introduction of process improvements in real life is a complex problem that involves many simultaneous factors. IT Management made a small number of process changes over a period of approximately 13 months to Projects Transitioned (discussed in a previous paper [14]). In this section we provide further contextual information about the process changes in the new project that followed IID since inception (Project New), to avoid decontextualization [4]. Specific practices adopted by this new IID project included: short iterations, iteration planning, scheduled iteration testing, iteration end demos, risk management, early prototyping, and external focus groups for user acceptance tests.

The reviewed RUP execution state used by Project New included: the iterative RUP lifecycle, Rational Tools, Role sets, and selected work products (approximately 13). The team experimented with iteration length during the Inception and Elaboration phases. At the end of first Elaboration phase, the team decided to follow three-week iterations as more become known about the project. Transition iterations were an exception; they were seven weeks long. The team also started to hold daily stand-up meetings during the Elaboration phases and the developers implemented unit tests to prototype and evaluate the risk of project tasks. Later in the construction stages the team stopped developing and maintaining these unit tests. Manual testing occurred during the last week of an iteration. During the last week of Construction iterations code was delivered to a staging environment - the "Sandbox." A dedicated tester and the business analyst used this staging environment to test and validate the builds. The team prepared demos for the business clients at the end of each iteration. These demos presented the progress of the iteration to business clients for feedback by showing a working version of the system in so far.

2.1 Methodological Approach

The data presented is longitudinal. It extends over three years of data from Projects Transitioned and over two and a half years of data from Project New.

The Goal Question Metric approach (GQM) proposed by Basili *et al.* [8] was adopted in order to formalize the research goals and to find appropriate measurements to answer them. The question: "How long did it take to fix bugs?", resulted in two metrics: bug-fixing responsiveness in days (quantitative data) and subjective views of business clients (qualitative data). This question and metrics were applied twice: once for (Projects Transitioned) and once for (Project New).

Quantitative Data. Bugs were grouped based on priority. Mixing all the bugs together would lead to a less realistic representation of bug fixing responsiveness as higher priority items would most likely be worked on first. Bug priority was used instead of bug criticality because according to interviewees, a clear definition of a critical bug was not available until the later stages of the RUP adoption. We define bug-fixing responsiveness as:

Bug-fixing responsiveness (in days) = The number of days between when the bug was submitted and when it was closed. (1)

Only bugs logged from the RUP Transition forward were measured for Projects Transitioned to allow a fair comparison of the affects of the IID adoption. Bug reports that did not include any action related to a developer analyzing and/or fixing the bug were excluded. As a result, we only included 958 bugs from Projects Transitioned and only 318 bugs from Project New in our analysis.

Qualitative Data. Data was gathered using field notes based on interview sessions with the project manager, technical lead, business analyst, and one developer from Project New. Three to four sessions were conducted with each of the participants

for approximately 20 minutes each time. The questions were designed based on qualitative interviewing techniques [12] with probing questions [13].

3 Results

On-time & On-budget Delivery. The risk list document and iteration demos provided a more tangible way to manage and negotiate expectations. Business clients bought into the idea of having a subset of the project delivered first instead of prolonging the delivery timelines. Based on items of this risk list the business clients agreed to break Project New into two phases at the end of Elaboration Iteration 2. Although the project took longer than it first envisioned, both phases of Project New were delivered on-time and met the deliverables milestones.

This government agency has a set budget given to the IT Department at the beginning of each fiscal year. The budget for Project New was set during Inception. The interviewed business clients and the project manager stated that this project was on-budget. The interviewees indicated that no overtime was required or imposed to the project team.

This was confirmed by the overall perception of this project in the organization that this was ***"the first IT project to be on-time and on-budget in six years."***

Better Bug Detection. A core goal for software is to deploy bug-free software. Bugs found before deployment are a sign that the overall process is working. As such, we analyzed where in the development process most bugs were being logged for both Projects Transitioned and for Project New. The bugs reports were grouped based on the staging environment where they were discovered. The ascending order of staging environment is: Dev (Development), Test (Testing) and or Sbx (Sandbox), UTE (External User Testing Environment), Act (User Acceptance), Prod (Production).

Figure 1 shows that during the waterfall days (Pre-RUP), 40% of all bugs were found after a production release for Projects Transitioned. A total of 47% of all bugs were found in the two latest staging environments (Act and Prod). During the RUP transition, the percentage of bugs found after a release dropped to 34%, and after the Partial RUP adoption, the overall numbers dropped to 25%. We do see an improvement in the amount of bugs found after a release (from 40% to 25%).

For Project New, Figure 1 shows that only 7% of all bugs were found after a production release. Close to 9% of all bugs were found in the latest staging environments. An interesting difference is that for Project New an external user testing environment (UTE) was set up for external focus group participants to test the application. 30% of the bugs reported were found during such testing.

Bug-fixing Responsiveness. Bugs were grouped based on priority. The arithmetic mean (average) in days, median, standard deviation, minimum (min), and maximum (max) number of days that it took to fix a bug were calculated for each group. Figure 2 illustrates the results. The values from Projects Transitioned and Project New are intercalated to facilitate comparisons. The averages and median values are substantially lower for the project that followed IID practices since inception.

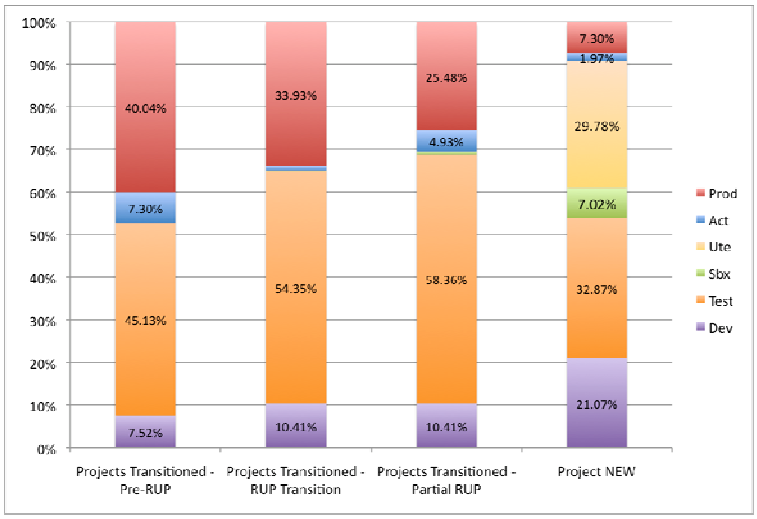

Figure 1. Bugs breakdown per staging environment.

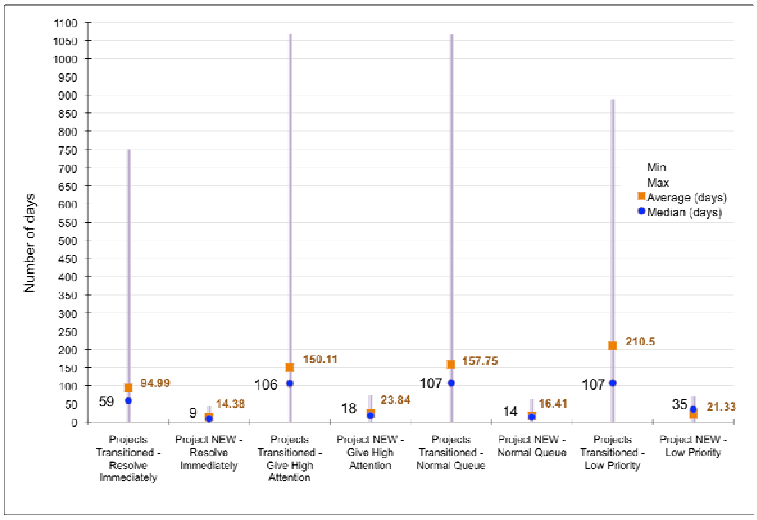

Figure 2. Bug-fixing responsiveness - day to closure.

The data was not normally distributed. The standard deviation was always higher than the averages. This variation in time to fix bugs within the same priority was not a surprise. The level of effort required to fix a bug is not always associated with its business impact and priority.

A quick turnaround to address an issue showed respect and concern towards the business clients. The business partners felt they could trust the team to fix issues

in a timely manner. To validate this sentiment, bug-fixing responsiveness was also measured relative to the iteration end dates. Project New's bugs labeled as "Resolve Immediately" were, on average, addressed within the iteration they were found. "Give High Attention" and "Average" bugs were addressed within the next iteration. Unfortunately that was not true of Projects Transitioned where due to the high volume of bugs introduced before the concept of iterations was adopted, existing bugs competed against new ones in the queue to resolution. In average bugs were fixed well after the end date of releases in which they were found.

4 Conclusion

Through the analysis of bug-fixing data and qualitative data gathered during interviews, we discovered that IID practices allowed a new project to: provide support for managerial decisions that lead it to be the first project in six years to deliver on-time and on-budget; avoid quality and stability issues; provide substantially better bug-fixing responsiveness than projects that migrated to IID. IID practices in both new and existing projects allowed the teams to detect bugs earlier on in the development lifecycle.

References

1. Alistair Cockburn. Incremental versus iterative development. On-line: http://alistair.cockburn.us/index.php/Incremental versus iterative development.
2. C. Larman and V.R. Basili. Iterative and incremental development: a brief history. Computer, 36(6):47–56, June 2003.
3. T. Dingsoyr, T. Dyba, and P. Abrahamsson. A preliminary roadmap for empirical research on agile software development. In Agile, 2008. AGILE '08. Conference, 83–94, Aug. 2008.
4. Philippe Kruchten. Voyage in the agile memeplex. Queue, 5(5):38–44, 2007.
5. C. Larman. Agile & Iterative Development, A Manager's Guide. Addison Wesley, 2004.
6. S. Ambler. Agile adoption rate survey. On-line: www.ambysoft.com/surveys/agileMarch2007.html.
7. Hakan Erdogmus. Agile's coming of age ... or not. Software, IEEE, 24(6):2–5, 2007.
8. H. Rombach V. Basili, G. Caldiera. The goal question metric approach. Encyclopedia of Software Engineering, 2:528–532, 1994.
9. Amr Elssamadisy. Patterns of Agile Practice Adoption. Lulu.com - InfoQ, 2007.
10. Mike Cohn. Patterns of agile adoption. On-line: http://www.agilejournal.com/content/view/734/111/.
11. Jutta Eckstein. Agile Software Development in the Large - Diving Into the Deep. Dorset House, New York, 1983.
12. H. J. Rubin and I. S. Rubin. Qualitative Interviewing: The Art of Hearing Data (2nd Ed.). SAGE, 2005.
13. The question man, What to avoid, What to do. On-line: http://www.a jr.org/article printable.asp?id=676.
14. Caryna Pinheiro, Frank Maurer, and Jonathan Sillito. Improving quality, one process change at a time. In 31st International Conference on Software Engineering (ICSE 2009), volume 31 of ICSE Companion 2009, pages 81–90. IEEE, 2009.