

Implementing an enterprise system at Suncorp using Agile development

J.A. Couzens

Suncorp, Brisbane 4000, Australia.

Email: james.couzens@suncorp.com.au

Abstract

Suncorp is a large financial services organization currently undergoing a significant shift in development process from a heavyweight waterfall process to a more Agile iterative method. This paper describes a large financial services company's experiences using Agile when implementing a major system replacement. Lessons learned from the process are compared to those described in the academic and professional literature as well as correlated with the Agile principles and concepts from named Agile methods.

Keywords: Agile, experience report, insurance, claims, Suncorp, Guidewire, ClaimCenter.

1 Introduction

This paper describes Suncorp's experiences and lessons learned implementing a core system replacement using an Agile development method. Development is generally described as Agile if it follows the values and principles of the Agile Alliance [1], [2] and implements, at least in part, a recognised Agile development process such as Extreme Programming, SCRUM or Crystal [3].

While Agile processes have their critics [4], [5] and some express concern over its manner of adoption [6] it is generally accepted that some form of software development agility is necessary to cater for a highly changeable business environment and evolving requirements [7], [8]. While Agile is seen as a solution for managing this change, many organisations struggle adopting Agile practices in a consistent way, challenged by its implementation and making many mistakes in the transition [9]. Suncorp is no exception in this respect.

2 Context

The Suncorp Group is a diversified financial services company tracing its banking roots to 1902 and insurance to 1916. It is now Australia's sixth largest bank and third largest insurer [10]. Growth is predominantly through merger, starting with the merger in 1996 with QIDC and Metway Bank to form Suncorp Metway. AMP's insurance interests were acquired in 2001, followed by the merger with the Promina Group

in 2007. This brings the Suncorp Group to its current position in the ASX20 [11] with sixteen thousand staff, seven million customers and over eighty billion dollars in assets [12].

Agile Software development methods have been used at Suncorp since 2003, predominantly in small niche developments or those that involved external software companies. The first formal adoption on a project was in 2004. Agile as a major initiative started with the arrival of Jeff Smith as CIO in March 2007, with an emphasis on it being a management approach rather than just software development [13]. In an effort to embed Agile Suncorp has invested heavily in its Agile program, implementing an Agile change programme, Agile coaching and creating an Agile curriculum to develop the skills required across different disciplines [14].

The ClaimCenter application [15] is an insurance claims management system from Guidewire Software being implemented at Suncorp to modernise the claims process. The initial implementation was in support of personal home claims. Currently underway are projects to implement the solution for claims in worker's compensation, personal motor, commercial motor, commercial property and compulsory third party. The ClaimCenter project was the first enterprise size implementation at Suncorp to use Agile methods. The implementation using Agile has been used as a showcase of the success of Agile methods for large system implementation at Suncorp [14].

3 The ClaimCenter Project

3.1 Project 1 - The Claims Business Model Programme (CBMP)

The Claims Business Model Program started at Suncorp as the Claims Cost Reduction Programme. The initial intent was to identify process improvements that would result in reduced leakage, paying claims that should not be paid, and lower handling costs. It was intended to have minimal impact on IT, however as the review continued two things became apparent, that most of the changes would require system support and that the current systems were not sufficiently flexible enough to support the changes in a cost effective way. After an

expedited product selection process Guidewire ClaimCenter was selected as the system in which the new process would be implemented.

The implementation of ClaimCenter commenced in August 2006. At the recommendation of the vendor an Agile delivery method based on Scrum using 4 week sprints was used for development directly associated with the ClaimCenter product. Originally it was planned to have seven development sprints, however the final number of sprints was nine. After completion of the development sprints there was a period of integration and acceptance testing before the system went live in July 2007.

ClaimCenter required integration with a large number of external systems and required co-ordination with as many as thirteen development teams including one external vendor and one offshore development team. The methods used by teams developing integrated systems was dependent on the prevailing method used by the responsible team. Methods varied from waterfall with long release cycles for mainframe systems to the use of Extreme Programming on two week iterations. In addition it was decided that a service oriented architecture (SOA) approach to integration would be used for customer, policy, payments, general ledger, receipting and claim interfaces with the intent to build them as much as possible for reuse.

3.2 Lessons learned from the first ClaimCenter project

In general the project was considered a success from both business value and IT implementation perspectives. It delivered significant business capability and was delivered close to on time and budget. There was strong business representation and a very broad stakeholder group providing input to requirements. Given it represented Suncorp's first attempt at delivery of an enterprise system using Agile techniques there were however many aspects that could be improved.

Some integrated systems had long release cycles or offshore development so requirements were provided to them as early as possible. However in many cases the ClaimCenter stories in the current sprint did not include these requirements. This resulted in a mismatch between what was produced by the integration teams and what was eventually required by ClaimCenter. This necessitated significant and expensive rework on the integration side. A stand-up was introduced across all integration teams resulting in improved communication and a better understanding of the integration dependencies. This confirmed the agile principle of "the most efficient and effective method of conveying information to and within a development team is face-to-face conversation" [2].

Learning: Give stories with an integration requirement a high priority as these represent areas of high risk due to distributed, independent teams.

Learning: Make the Agile team adapt their schedule and priorities rather than more rigid processes, the cost should be lower.

Learning: Just because a team is working using a non-Agile method does not mean they cannot benefit from Agile techniques.

These learnings in part support the concept from both Scrum and Extreme Programming of prioritising the backlog so that stories generating the highest value to the customer are done first [16], [17]. There is however a question over whether this would be categorised by the customer as generating high value. They are certainly areas of high risk, one of the reasons they were commenced early. The project team needs to take on a more active role of advocating why such areas of risk also represent high value. Schwaber [18] describes how application infrastructure items were prioritised by the customer above application functionality seen as enablers. This is similar to how the integration components for ClaimCenter can be viewed and should have been explained.

There was a significant scope creep during the life of the project. Additional requirements were added to the backlog but the expectation was that the backlog would still be implemented in the original timeframe.

Learning: Manage the backlog well. Make sure the business understands what the backlog is. Make it clear what can be achieved. If there is a hard end date make it clear where in the backlog the current end date is so it is clear what will not be delivered.

Again, both Scrum and Extreme Programming discuss managing the backlog and the need to negotiate with the customer to ensure that the content of each iteration is achievable [16], [17]. Requirements will necessarily change and grow over time especially when using an iterative process [19]; managing the backlog effectively ensures the most valuable changes can be included in the end product.

There was an emphasis placed on delivering demonstrable functionality over delivering to a definition of done leading to shortcuts to get a demonstration working. It is unclear whether this was driven by a legacy attitude to present a more positive picture than was actually the case or a new attitude to demonstrate that Agile was working. This poor definition of done also led to a false view of the teams velocity. Stories for a sprint were estimated without considering a complete software engineering process or what it meant for a story to be complete.

Learning: There needs to be a definition of done for a story card so that it is clear when it is complete.

Learning: Estimation should be inclusive of all the tasks that go into producing working code; analysis, design, test case development, coding, code review and test execution. It is not just writing the code.

Schwaber [18] puts a practical slant on the definition of done. He refers to an *increment of potentially shippable product functionality* where the definition of done requires that outputs from a sprint should be available to implement immediately. For Schwaber "done" includes analysis, design, coding cleanly and legibly, unit testing, test automation, code review and refactoring. All these activities must be included in estimates.

3.3 Project 2 – Joint Venture Household and Imaging

The intent for the joint venture household and imaging project was to leverage the code base and infrastructure implemented for the first project. Initial discussions were held with the joint venture in December 2006 with the project commencing around April 2007 five months prior to the full rollout to Suncorp and GIO home claims. A decision was made to delay the joint venture development sprints until the first ClaimCenter project was fully in production. As a consequence solution design and a significant amount of analysis and story card elaboration was complete before the first development sprint.

After the first project went into production an initial refactor was done to prepare the code to support multiple legal entities and brands with the first of five three week development sprints starting in November 2007. At this point all stories were identified and the backlog had been assigned to different sprints. Based on feedback from the first project a clear definition of done was agreed and the team understood what was required to consider a story complete. In February 2008 half way through the fifth development sprint a new and critical requirement was identified that necessitated an additional two week development sprint. During development a significant effort was put into developing an automated test suite that could be configured to drive smoke, regression, performance and volume testing [20]. This had the result of reducing rework in subsequent sprints allowing the inclusion of additional scope without impacting time or budget and reducing duration of system testing.

Integration leveraged the SOA and other integration points from the first project. Integration related stories were given a high priority and were elaborated early, before the commencement of ClaimCenter development. This allowed integration development to commence before ClaimCenter development reducing the risk of future change. ClaimCenter development aligned completion of integration stories with completion of development in integrated systems allowing early system integration testing to occur.

3.4 Lessons learned from the second ClaimCenter project

The second ClaimCenter project was considered a greater Agile success than the first and has been used as an example of how Agile is improving project success at Suncorp [14]. There were a number of factors that contributed to this success: the development team particularly the lead had previous Agile experience; the team was building on a previous implementation benefiting from an existing code base; existing relationships with integration partners and a substantially reusable SOA reduced the complexity of the integration environment; and there were benefits of lessons learned.

There were however further lessons to be learned. Getting ahead in terms of story card elaboration in some cases became a burden for business analysts and subject matter experts. When they were required to revisit the elaborated story cards several months after their initial completion they were often in completely different subject domains.

Learning: There is value in recency. Don't get too far ahead as there is often difficulty revisiting the decisions and thought processes that went into elaboration months in the past.

The Agile Manifesto [1] values *individuals and interactions over processes and tools* and *working software over comprehensive documentation*. What was learned was that additional upfront time spent to complete more comprehensive documentation did not remove the need to spend time face to face. From the Lean Programming perspective the additional time spent up front would be an example of waste that could be eliminated, producing unnecessary inventory [19]. Performing elaboration closer to the time required will result in less documentation and a clearer link between the business need and software.

The late addition of requirements, while it was able to be catered for, would have been identified if the correct stakeholders had been identified earlier in the project. Given the long lead time before development commenced there was ample time to identify the stakeholders, especially given the missed equivalent stakeholder group in Suncorp was a the project sponsor for the first project.

Learning: Stakeholder analysis is a key precursor to identifying stories. It is not good enough just to have business users on the project, there has to be representation and active involvement of all stakeholder groups.

Ambler [21] in his Agile modeling method refers to the practice of active stakeholder participation and establishes a broad definition of a stakeholder; not just direct users of the system but also managers of users, senior managers, support staff and so on. Stakeholders need to be involved, not just passive observers, and

make contributions to the collective solution based on their perspective.

The business used a traditional project structure with many levels of management oversight and control. There were defined sign-off points that did not necessarily fit with an Agile approach and in many cases the sign-off cycle had long delays. The velocity of the project depended on quick sign-off for stories so in most cases the stories would be implemented without sign-off. This did result in a higher level of requirement churn, not because the requirements changed, but because the people with the knowledge of the requirement were not involved enough in the original specification.

Learning: A decision maker from the business with authority to make sign-off decisions needs to be available to give input into stories and turn around decisions at the project's velocity.

One of the four basic values of the Agile Manifesto is that of *customer collaboration over contract negotiation* [1]. The assumption behind this value and backed by the principle of working daily with the customer [2] is that customer representatives are both knowledgeable, or have access to knowledge, and are empowered make decisions.

4 Conclusions

The experiences at Suncorp highlight some of the issues faced when using Agile methods both for large system implementations and in large organisations. As identified by Lindvall et al [22] and Fitzgerald, Hartnett and Conboy [23] large organisations will tailor the process to suit the way the organisation works. This was also evident at Suncorp with, in the first implementation, a modified Scrum process followed with the second implementation adapting the process further based on lessons learned. It is also apparent that existing organisational practices and structure needs to be considered along with how the particular software development fits within the wider project context [22], [24]. Berger in looking at a bureaucratic, hierarchical government organisation found that an Agile project was at odds with the organisational culture [25]. Individuals were reluctant to make decisions without ensuring there was a higher authority taking responsibility. This caused blowouts both in project costs and timeframes. While not severe enough to cause project failure this was also the case in Suncorp with existing project management and control practices hampering the velocity that could be achieved. Consideration needs to be given to how Agile practices fit with Suncorp's more traditional project control processes and business hierarchy.

Changing and unstable requirements are cited as one of the key benefits of implementing Agile methods [22], [7]. In the first ClaimCenter project there was a significant churn in requirements, both from the

perspective of changes to existing requirements as well as increased scope. Possibly due to poor understanding of Agile by the business, Agile was seen as a way to increase the scope of a project without impacting time, budget or quality. This belief was unfortunately not corrected by the IT project team. The second project also experienced changes in scope. It managed the changes better by managing all change through the backlog and continuously discussing the impact to the backlog with the business stakeholders, allowing them to identify the highest priority items.

Ambler [26] points out that difficulty with stakeholder identification and management in large organisations and projects is one that affects both traditional and Agile projects. It impacts Agile in particular due to the need to maintain velocity through quick decision making. Stakeholder management was a particular issue at Suncorp. As noted in some cases stakeholders were not consulted until late in the process, sometimes not until the system was being implemented into production.

It is difficult to be certain whether using Agile methods for implementing ClaimCenter had a positive, negative or neutral impact on Suncorp's ability to deliver. Given Suncorp's past history in delivering large and complex systems my belief is that it had a positive effect. Certainly the overall perception from the team involved, both business and technology was that Agile added value. In reflection some of the pitfalls of the first project may have been avoided if there had been deeper Agile knowledge and advocacy in the team. However there may be truth to the saying that a lesson learned hard is a lesson learned well.

The use of ClaimCenter to pilot large scale Agile implementation pre-empted a change in method that would happen at Suncorp anyway and our experiences and learning have placed Suncorp in a better position to understand the subtleties involved when introducing Agile into a large traditional IT organisation.

References

- [1] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham et al, Manifesto for Agile Software Development, <http://www.agilemanifesto.org/> (accessed 12/12/2008)
- [2] Principles behind the Agile Manifesto <http://www.agilemanifesto.org/principles.html> (accessed 12/12/2008)
- [3] L. Lindstrom and R. Jeffries, *Extreme Programming and Agile Software Development Methodologies*, Information Systems Management, 21:3, 2004
- [4] D. Longstreet, *The Agile Method and Other Fairy Tales*, <http://www.softwaremetrics.com/Agile/Agile%20>

- [Method%20and%20Other%20Fairy%20Tales.pdf](#)
(accessed 12/12/2008)
- [5] G. Keefer, *Extreme Programming Considered Harmful for Reliable Software Development*, <http://www.stickyminds.com/sitewide.asp?Function=edetail&ObjectType=ART&ObjectId=3248&tth=DYN&tt=siteemail&iDyn=2> (accessed 12/12/2008)
- [6] C.J. Neill, *The Extreme Programming Bandwagon: Revolution or Just Revolting?*, IT Professional, IEEE, September | October 2003.
- [7] L. Cao and B. Ramesh, *Agile Software Development: Ad Hoc Practices or Sound Principles?*, IT Professional, IEEE, March | April 2007.
- [8] P. Meso, and R. Jain, *Agile Software Development: Adaptive Systems Principles and Best Practices*, Information Systems Management, 23:3, 2006
- [9] L. Gurses, *10 Mistakes in Transitioning to Agile: Slow down the transition in order to go fast*, Dr. Jobb's Journal, December 2006; 31, 12
- [10] <http://www.suncorp.com.au/suncorp/about/about.aspx> (accessed 12/12/2008)
- [11] http://www2.standardandpoors.com/portal/site/sp/en/au/page.topic/indices_asx20/2.3.2.8.0.0.0.0.2.3.0.0.0.0.html (accessed 12/12/2008)
- [12] <http://www.suncorp.com.au/suncorp/about/history.aspx> (accessed 12/12/2008)
- [13] <http://www.zdnet.com.au/insight/software/soa/Suncorp-CIO-Jeff-Smith-Q-A/0.139023769.339293040.00.htm> (accessed 12/12/2008)
- [14] <http://www.zdnet.com.au/news/software/soa/Suncorp-takes-agile-leaf-from-Apple-s-book/0.130061733.339286413.00.htm> (accessed 12/12/2008)
- [15] Guidewire ClaimCenter, http://www.guidewire.com/our_solutions/claimcenter (accessed 11/3/2009)
- [16] K. Schwaber, *What Is Scrum*, http://www.scrumalliance.org/resource_download/227 (accessed 28/2/2009)
- [17] Iteration Planning, <http://www.extremeprogramming.org/rules/iterationplanning.html> (accessed 1/3/2009)
- [18] K. Schwaber, *Agile Project Management with Scrum*, Microsoft Press, 2004
- [19] M. Poppendieck, *Lean Programming: Part 1 of 2: Assembly-line production techniques apply to software, too*, Dr Dobb's Portal, <http://www.ddj.com/architect/184414734>, May 01 2001 (accessed 1/3/2009)
- [20] <http://www.astqb.org/educational-resources/glossary.php> (accessed 12/12/2008)
- [21] S.W. Ambler, *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*, John Wiley & Sons, 2002.
- [22] M. Lindvall, D. Muthig, A. Dagnino, C. Wallin, M. Stupperich, D. Kiefer, J. May, T. Kahkonen, *Agile Software Development in Large Organizations*, IEEE Computer, IEEE, December 2004
- [23] D Karstrom and P. Runeson, *Integrating agile software development into stage-gate managed product development*, Empirical Software Engineering, 2006 11
- [24] B. Fitzgerald, G. Hartnett and K. Conboy, *Customising agile methods to software practices at Intel Shannon*, European Journal of Information Systems, 2006, 15
- [25] H. Berger, *Agile development in a bureaucratic arena - A case study experience*, International Journal of Information Management, v 27, n 6, December, 2007
- [26] S.W. Ambler, *Scaling Onsite Customer*, Dr. Dobb's Journal, January 2008; 33, 1