

# Agile Project Experiences – The Story of Three Little Pigs

Craig Smith  
Suncorp  
[craig.smith@suncorp.com.au](mailto:craig.smith@suncorp.com.au)

Paul King  
ASERT  
[paulk@asert.com.au](mailto:paulk@asert.com.au)

## Abstract

*Over the last few years we have had the good fortune to aggressively apply the agile practices on a number of projects with great success. These successes, however, have not been achieved without challenges and lessons learnt along the way. This experience report specifically highlights examples from three different software development projects of varying sizes within this period and within the same organization. This is the story of three little pigs, where in all cases the pigs were well and truly committed.*

## 1. Introduction

Agile practices when applied with consistency, understanding and enthusiasm from the project team, almost certainly result in a successful outcome. However, throughout the life of any given project in any given organization, there are always unanticipated challenges to be overcome and lessons to be carried over to future work.

In this paper, we examine three very different projects using an analogy based loosely on the children's fairy tale "The Three Little Pigs" [1]. In our rendition, the little pigs have been sent out into the world by the organization to seek their fortune in the form of a successful project outcome. No pigs were eaten in this curly little tale but certainly technical success, organizational success and personal success<sup>1</sup> were all in jeopardy due to the huffing and puffing emanating from numerous wolves.

The project made of straw is an example of a medium-sized project using Scrum processes. Some aspects of this project were in quite good shape but other aspects were a little fragile. Some strengthening of its agile techniques was needed to stop it being blown over.

The project made of sticks is an example of a small agile project that was part of a larger waterfall program of work that delivered major benefits early and proved that small projects can run using agile techniques also resist being blown over.

The project made of bricks is an example of a successful project using XP techniques to the extreme with the entire team committed to the process, but with the wider community still maturing to the agile process, the strength of the team and practices meant it could not be blown over.

The analogy of the pig in this paper also draws its inspiration from the fable of "The Chicken and the Pig" [2], whereby in all of these projects the challenge was to make all of the project members pigs and to be accountable for the outcomes of the project.

## 2. The Project Made Of Straw

This project was a pilot for a component of a new processing system being introduced to replace a number of large legacy insurance processing systems. Another component of this system had already been successfully delivered to production by a separate project team. The project mentioned here was based around the same processing system, but was a pilot for new functionality for both the vendor and the organization. The vendor had already heavily promoted the Scrum methodology [3] and via their onsite representatives introduced the project team to these techniques as an approach to delivering the pilot functionality. The non-vendor team members were mostly new to agile.

The project team was geographically dispersed between two offices in two different Australian cities (Brisbane and Sydney), with the main business users located in an office in a remote third city (Perth).

Senior management decided that agile best practices learnt in other projects should be applied to this project, hence our involvement. Senior management also recognized that some features of this project (such as the geographical dispersion) made it a little fragile

---

<sup>1</sup> See Figure 1-1: Types of Success in [9]

but that if agile was to become more mainstream throughout the organization, it would need to be applied across many different kinds of projects and we should begin the process of learning how to stretch our agile knowledge to cope with such projects.

We have described this as the project made of straw, as it commenced with a number of factors that made it fragile, but with some strengthening throughout the life of the project it's construction made it resilient enough not to fall over.

## 2.1. Key Challenges

### **Geographically and technically dispersed team:**

For reasons of vendor resourcing, ongoing support and graduate training, the key members of the project team were dispersed across multiple cities. The team was also technically dispersed with all of the senior technicians based in one location and the project management, vendor representatives and graduate technicians based in the other location.

To combat this, team members traveled between the various locations on a regular basis, with the whole team being together for a couple of days every couple of weeks. Team communication was further enhanced using the standard Scrum techniques of a daily scrum, sprint planning and sprint reviews along with a traditional project team meeting.

Early on in the project, we introduced the concept of pair-programming, but due to the nature of the development work (an emphasis on configuration over coding), the dispersion of the team members and the lack of an agile coach in both environments meant that this technique was dismissed by members of the team. Ultimately, this evolved to a concept of buddying, where a senior technician would be paired with a graduate technician for the duration of the sprint for the purposes of mentoring and review. The buddying concept gave the partial benefits of information sharing, as two technicians were aware of the storycard and its resulting solution, and it gave the technician (especially the junior ones) a point of reference when they needed help. Ultimately, buddying was not pairing, which meant that it was difficult to predict in planning how long the buddying would take outside of normal reviews, and in some cases the buddy spent more time getting up to speed on issues and assisting with them than they would have if they had paired in the first place.

**Daily sitdown:** The project team had been doing daily scrums religiously since the beginning of the project, however due to the geographical dispersion, these were done via a telephone conference call.

Interestingly, we found that many of the values of a standup meeting were lost when a telephone is involved, as whiteboards cannot easily be referred to and people tend to sit down around the telephone. This was a practice that was hard to fix, as the habit was already entrenched and it was difficult to enforce the standup behavior to team members on the other end of the telephone.

**The four-week syndrome:** As the vendor had initiated the Scrum methodology at the beginning of the project, it had followed the book [3] to the letter and introduced the concept of a four-week sprint. It was clear, however, that these iterations were not working as effectively as they could have been.

The final sprint planning session was held traditionally on the first day/s of the sprint, and involved the entire team going through a workshop of the storycards on screen and estimating as they went. We discovered that whilst the team had agreed on a goal for the sprint (usually based around required functionality), there was no clear goal in relation to velocity or time available [4] in the iteration, and the time spent by the full team discussing and estimating cards was not fully productive for all team members.

Our first recommendation was to cut the sprint length from four weeks to two weeks. This was initially met with a lot of skepticism from the project team because in their belief they had trouble getting ready for a sprint in the space of four weeks, so two weeks would be even harder. Ultimately, the team (especially the business analysts who were feeling overwhelmed) realized that the smaller number of requirements and storycards required each week actually made their life easier and less stressful.

The notion of the XP Planning Game [5] was introduced to try and provide early warning, especially to the analysts, of the size and requirements of the sprint. Estimation was also completed prior to the start of sprint by a subset of the technical team on a rotational basis each sprint.

**Releasing before we are done:** Another outcome from a combination of the initial four-week sprint and the project release plan was that the team were releasing the code at the end of the third week and using the fourth week to work on deployment, test code and backlog. The result was that the technical team were leaving tests to the end which was affecting quality, no urgency to complete backlog because it would be picked up in the fourth week and deployment activities were not time boxed and prioritized.

The introduction of the two-week sprint allowed this issue to be rectified by focusing on completing the assigned storycards in the two week sprint period (including related test code and high priority backlog

items) and completing a storycard to release the prior sprint in the next sprint.

## 2.2. Lessons Learnt

**One chance to influence:** With a team that is new to agile techniques, and even more so when the team is dispersed, new techniques such as test-driven development and pair-programming are hard to introduce if the team does not have appropriate mentors and coaching. Furthermore, if the concepts are introduced but not reinforced with this technique initially, it is very difficult to enthuse the team to give them a second chance.

**Challenge the different:** The emphasis from senior management to try and harness the learnings from past agile projects was a key aspect of this project, however the challenges with the project discussed above overshadowed the resulting outcomes. However, the work to introduce virtual desktops, Linux operating systems for development (in a traditionally Windows environment), iteration and collaboration tools, deployment tools and continuous test execution (to name a few), are all outcomes that have made projects following on in the organization stronger, as the project was able to set a precedent where roadblocks had stood before.

**Technology helps, but collocation is better:** The introduction of video conferencing as part of iteration planning sessions, the use of virtual workstations so team members could view and share desktops to view code and the introduction of XPlanner [6] to manage sprints, Jira [7] to manage issues and Confluence [8] to share team information aided team communication. However, it was clear that when the team was most productive when they were together, and as a result the future phases of this project will be conducted with a collocated project team.

## 3. The Project Made Of Sticks

This project was a small enhancement to a backend batch processing system that is used by the organization as part of a larger mainframe-based legacy application. The small enhancement was requested as part of a larger program of work that was being undertaken by the organization to transform the processing of a recently combined entity onto the same processing platform. The larger program of work was being run, mostly, in a traditional waterfall fashion, but the project management were keen to see the outcomes of the agile techniques used in this small and relatively isolated piece of work.

Initially this piece of work had been estimated and specified using a traditional specification and estimation document. The size of the change meant that this specification was relatively small and the anticipated size of the change was expected to be around 25 days. Two developers were assigned to the project, primarily as a mentoring exercise for both the technology (Java) and the agile methodology and coding techniques. A hybrid agile approach was followed.

We have described this as the project made of sticks, as the project was small and resourced correctly but its role in the larger project made it initially fragile in the eyes of some of the project management and the state of the legacy codebase gave some doubts to the project team. However, with the use of agile techniques and some hard work, its construction stood up to the challenge.

### 3.1. Key Challenges

**The busy customer:** A business customer was assigned to this piece of work, but unfortunately due to multiple commitments with the larger waterfall project was unable to be co-located with the development team or be fully and directly available for the entire iteration.

Use of Instant Messaging was piloted within the organization on this project and proved to be relatively successful in dealing with this situation, especially when quick questions needed to be asked and being able to know when all of the parties were available to talk. Storycard signoff and review sessions were held as required, and at least once per iteration.

**Start waterfall, finish agile:** As this project had originally started with a design and estimation document, we first rewrote all of the specification back to storycards. This enabled the development resources to understand the requirements as well as set the basis for the iteration plan.

Ultimately, as a development team, we decided to write the cards on the behalf of the business customer from the specification and take them through a review process rather than ask them to rewrite them again. The team was not fully comfortable with this approach at first, however as the project had been through multiple requirements gathering cycles already and the number of requirements was very small, we decided that the risk for this approach was low in this instance.

Storycards were initially written on “stickies” but transferred into XPlanner [6] so that remote customer and project management could see visible progress. A project wiki was also initiated to track information and for handover at the end of development.

### 3.2. Lessons Learnt

**Just because you're small doesn't mean you drop the ball:** One of the key outcomes from this piece of work was the ability to demonstrate that regardless of the size of the project, agile practices can be followed. There was a belief by some staff in the organization that following agile principles meant you needed a large overhead, such as a dedicated scrum master and delivery lead.

Also, with a team size of two developers, some felt it was too small a project to adopt pair programming. However the development team felt this was the way to proceed. Indeed, it meant that the original mentoring aim for the project was met, while also ensuring quality and understanding of the code. This turned out to be an important decision because the original codebase had limited test coverage and the domain knowledge of the product and related codebase were limited. As a result, not only were the goals of the project met within the target timeframe, but the test and checkstyle coverage of the code were improved as well.

Also, the two developers brought different experiences to the project from a technical and agile perspective, so experiences were shared on both sides and small issues were usually resolved relatively quickly. Iteration Management was also managed by one of the developers, and only required a couple of minutes per day plus a small amount of time to prepare and facilitate the weekly meetings.

**Short project equals short iteration cycles and big progress:** We decided to run one week iterations due to the short development time frame, small number of storycards and the availability constraints of the business customer (so that review of progress and storycard signoff was happening at least once a week). Iteration Kickoff and Retrospectives were combined into a one hour session mid-week to accommodate the availability of the wider project management.

For each iteration a goal was specified, and the outcome of that goal was demonstrated at the end of the iteration (this was usually the simple demonstration of a file format that was growing to meet the specifications over time). This technique clearly showed the project management that progress was being made (despite it being small), and as a result the original doubts about the change being delivered reliably and on-time were eliminated (and their focus diverted to other larger project issues).

## 4. The Project Made Of Bricks

This project was a large enterprise initiative that started in 2005 and ran for two years across three project phases. The objective was to produce a central document generation service that could be used by the entire organization for the generation of documents to be output initially via print, email or facsimile format and later into PDF and mailhouse ready format.

This project was driven out of a small agile team that had already been involved in the support and maintenance of a smaller, similar designed and more integrated documentation solution. The lessons learnt from maintaining this existing application and the issues that had been realized in both a technical and organizational sense, meant that the core development team were keen to not repeat the mistakes of the past and ensure that the new service was of high quality and fit for purposes.

We have described this as the project made of bricks because when the project commenced, the core team consisted of a small but highly committed team of agile developers who were very XP-focused and aware, architects and designers who were knowledgeable and focused on delivering an open and best practice solution for a document generation system and project management who were cautious but open to the agile methodology proposed. This made for a strong base from which to start a project.

### 4.1. Key Challenges

**Honesty leads to trust:** As this project was fairly early in the organization's uptake of agile, no tools were in place for reporting to management, so as a result, much more information was flowing to management than they were used to. Initially they were ready to spring into action and remedy any perceived problems (for example, when iteration burn-up charts were showing a 10% drop in velocity when the programming was ramping up or justifying how an extreme agile approach was improving our outcome). Over time, as the team demonstrated its ability to work through its own issues internally, they moved to a role of assisting and sheltering the team rather than reacting to spikes in the metrics.

**Bring your own screwdriver:** Given the nature of a large organization, the team was required to relocate several times throughout the life of the project. Each location presented its own opportunities and limitations. Squeezing into meeting rooms, legacy cubicles farms and desks curved in such a way to make pairing difficult, a level of ingenuity (and rudimentary carpentry) was required to 'knock' the team space into

shape. Installing the chimney and pot of boiling water certainly raised some eyebrows!

By choosing accommodation that was suitable for a large team but was somewhat isolated, we improved the ability of the team to remain focused at the expense of making it harder to stay in touch with and communicate with other staff within the organization.

**Be careful what you promote:** The majority of the development team felt very strongly about particular XP practices such as pair programming and test driven development, but found it difficult to articulate the benefits to management and senior development staff at the time.

The team learnt over time to be careful about drawing too much attention to these practices and instead focused on achieving measurable goals such as code delivery and low defects. These metrics ultimately enabled the team to better articulate the benefits of their practices. Staff rotation through the project also allowed the agile message and many of the practices to start propagating throughout the organization.

## 4.2. Lessons Learnt

**Build it and they will come:** With the commitment to an XP methodology up front and the clear enthusiasm for the development team to follow the XP practices to the extreme, a clear benefit was the high level of expertise of candidates that applied for positions throughout the project. In most cases, these candidates came via word of mouth from conversations with members of the team. The result throughout the project was a team that was “the place to be”, both within the organization as well as the wider community, and as a result a relatively low-turnover of staff was the result.

When interacting with our third party software vendors we took time to explain our approach to agile, and as result they were excited about many of the practices and took on board many of our ideas.

**Turning the dials to eleven and doing XP to the extreme:** From day one of the project, the basic XP practices were followed by the team resulting in a very innovative environment with in-built continuous improvement. Throughout the entire life of the project the development team achieved the lofty goals of 100% code coverage from unit tests, all production code paired and test-driven, minimal design up front (but with an appreciation for when such design made sense), customer focused outcomes, daily development pair-programming rotation, full continuous integration, retrospectives and continuous improvement achieved and high levels of automation (including an installer

that loaded and configured the approximately 20 tools developers used onto a freshly installed operating system) [9].

On top of this, the team added their own enhancements including keeping all storycards to at most 3 days in size, ensuring one developer is the owner of a story while the other pair rotates daily, quality metrics which forced each class to contain only a small number of simple methods and almost no duplication, many IDE enhancements which automated many repetitive tasks and a system of working in pairs that allowed productivity to remain high.

**Everyone has a voice:** To reach a consistent level of productivity, the practices needed to evolve to allow for different work styles. In the early stages of the project, considerable time was spent setting up metrics that involved many group discussions. Similarly, whenever large architectural decisions needed to be made, typically more than just a single pair made the decisions.

A scheme for adding FIX and TODO comments into the code was established, whereby if one team member wanted to rigorously check every change made to the code (and potentially its knock-on effects throughout potentially large parts of the codebase) they could place leave a trail of breadcrumb comments in the code allowing them to satisfy their rigorous nature while not interrupting the flow of the pair/team.

**Design up-front when appropriate:** Sometimes applying the simplest possible solution to a storycard does not result in the simplest overall solution. This project provided customers with an XML schema that could be used for validation of their output, which greatly reduced the amount of integration testing required and meant getting the XML nailed down ahead of time by not producing it iteratively.

## 5. The Wolf that Huffed and Puffed

The story of the three little pigs would not be complete without the role of the wolf. In our analogy, the wolf represents any influences that threatened to blow away project success. This includes internal wolves dressed as sheep, packs of non-believer wolves looking for any slightly strange-looking (hopefully weaker) projects to devour and even well-meaning management wolves not familiar with where we keep our pots of boiling water!

We certainly observed many kinds of wolves at one stage or another. Some wolves were being threatened because the way they were rewarded traditionally ended with project delivery and not with how it went once the project was in production, whilst some were just skeptical about any kind of change.

Some wolves just didn't understand because we had no shared understanding of what quality meant, or what we needed to do to achieve quality, or even what agile meant (a new CIO wolf is fixing this organization wide now). Some well-meaning wolves tried to help by applying quick fixes to problems that the agile teams themselves should have worked through. Other wolfish behavior just surfaced due to the lack of maturity of agile within the organization.

While many of these wolves were very capable of large scale huffing and puffing, in the case of our three projects, they were resilient enough to not be blown down.

## 6. The Moral of The Story

We don't always have the ability to construct our projects of the finest materials, and even when we do, ongoing maintenance is required to ensure that the construction is resilient enough to survive the huff and puff of the wolves on the outside.

There is no defined template that can be rolled out verbatim to agile practices. Different approaches must be put in place for different team sizes and across different projects such as the ones we described above.

Storycard estimation remains a difficult activity, however having an appropriate set of tools in place for managing iterations helps considerably and allows agile responsiveness if estimations aren't perfect.

Obtaining and keeping team focus is a key goal of an agile team but the ways to achieve this are different for every project.

The organization has grown since the story of these three little pigs. An agile change program is being lead by senior management and rolled out across the organization and the current projects that we are involved in are using the lessons learnt from these projects to further improve the success of projects that are being conducted using the agile methodologies.

## 7. Acknowledgments

The authors wish to thank Suncorp management, members of the EasyDoc, EasySuite and eBusiness

Solutions teams as well as other Suncorp staff who assisted the authors or were involved in the described projects or related activities within Suncorp. Special thanks to Dalton Cranston for his assistance on a draft of this paper.

Suncorp is one of Australia and New Zealand's largest diversified financial services providers, supplying banking, insurance and wealth management products to around 7 million customers through well-established and recognized brands such as AAMI, Australian Pensioners Insurance Agency, Shannons, Vero, Asteron and Tyndall, as well as Suncorp and GIO. Today, Suncorp is Australia's sixth largest bank and second largest domestic general insurance group, with over 16,000 staff. Suncorp has representation in 450 offices, branches and agencies throughout Australia and New Zealand.

## 8. References

- [1] Wikipedia: *Three Little Pigs*, [http://en.wikipedia.org/wiki/Three\\_Little\\_Pigs](http://en.wikipedia.org/wiki/Three_Little_Pigs)
- [2] Wikipedia: *The Chicken and the Pig*, [http://en.wikipedia.org/wiki/The\\_Chicken\\_and\\_the\\_Pig](http://en.wikipedia.org/wiki/The_Chicken_and_the_Pig)
- [3] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*, Prentice Hall, 2001.
- [4] M. Cohn, *Agile Estimating and Planning*, Prentice Hall, 2006.
- [5] K. Beck and C. Andres, *Extreme Programming Explained*, 2<sup>nd</sup> Ed, Addison-Wesley, 2005
- [6] *XPlanner*, <http://www.xplanner.org/>
- [7] *Atlassian Jira*, <http://www.atlassian.com/software/jira/>
- [8] *Atlassian Confluence*, <http://www.atlassian.com/software/confluence/>
- [9] P. King and C. Smith, *Technical lessons learned turning the agile dials to eleven!*, Conference Proceedings Agile 2008