

Going Agile – A Case Study

Dwayne Read
Software Process Consultant
Strategic Systems
dwayne@ss.com.au

Grey Properjohn
Systems Analyst
Snowden Technologies
gproperjohn@snowdengroup.com

Abstract

This case study examines the approach undertaken by Snowden Technologies in adopting a range of agile techniques into their product and custom software solutions for the mining industry. The catalyst to adopt various agile techniques stemmed from the growth of the development team (30+) and the need to further integrate the development activities across multiple offices. This required a clearer and more consistent approach to the software development process.

Snowden Technologies involved the users (Developers, Project Managers, Team Leaders, Geologists, etc) upfront and prioritised the project's scope resulting in the selection of specific techniques from a range of agile methodologies (XP, FDD, Scrum, Crystal, DSDM, RAD/JAD). Implementation of the techniques paralleled various agile principles through the use of scoped releases, development iterations and feedback through regular reflections. The selected agile techniques (e.g. iterative development, Domain Object Models, code inspection, automated testing) were incorporated into a consistent software process with various "value add" techniques (e.g. CUT complete, code coverage, code analysis) that integrated with the development tools.

The key lessons learnt from this approach were that to incrementally introduce agile techniques was very effective; embedding the process elements into the development tools helped to reinforce the techniques; and it proved that the agile techniques and PRINCE 2 for project management could be customised to collaborate into an effective solution. The latter required some compromises from the "pure" agile view such as to allow a project schedule to define all Work Packages (Iterations) at the start of each Stage (Releases).

1. Keywords

Agile, methodology, software process, change management, project management, brainstorming, reflection, iterative development, JAD, Domain Object Model, architecture, code inspection, automated testing, Team Foundation Server, TFS, PRINCE 2.

2. Background

Snowden Technologies is a division of the Snowden Group that provides global consultancy services and software solutions to the mining industry, and has a development capacity of over thirty developers located in Perth (head office), Brisbane and Johannesburg. The reliance on delivering quality software solutions to its clients has continued to increase over recent years to the point that being able to rapidly respond to the needs and unique/specific requirements of clients is a critical success factor to the business.

Reflection workshops, on a monthly basis, with global feedback from all developers sought via the weekly Tec meeting with the Agile Project as an agenda item, and also through direct communications with the Agile Project team members. These feedback mechanisms were the start of establishing a continuous improvement philosophy.

5. Process Flow

A "road-map" of the development process showing the relationship of the selected techniques was produced in the first Work Package for the project. This is seen as the equivalent of the "logical architecture" model and also captures the high level requirements of the proposed implemented methodology. This agile view, Figure 4, also served as a planning and progress "information radiator" (as per the Crystal methodology [4]), by highlighting the techniques that were being introduced for a given Stage of the Agile Project. Figure 4 shows the process road-map at the time of writing this paper. This includes a handful of techniques from the Project Management / PRINCE 2 process (e.g. High Level Requirements, Project Brief, Authorise Project/Stage, Stage Test Plan and Stage Acceptance Testing) which serves to give further context in the developers' view.

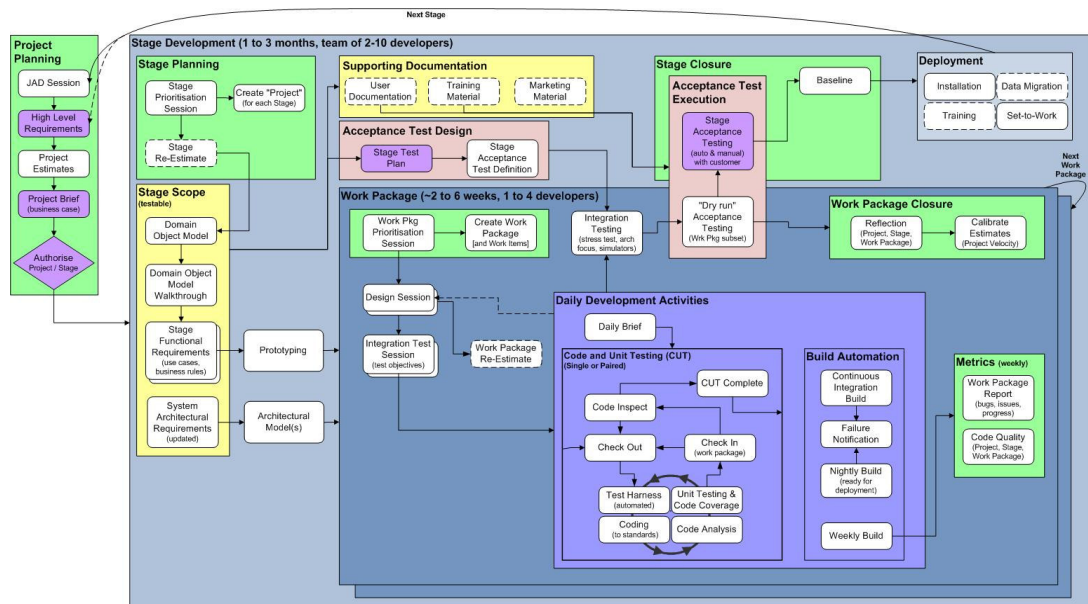


Figure 4: Snowden Technologies' Agile (STAg) Process Road-map

The STAg methodology process road-map captures the complete suite of activities, which were prioritised and implemented, mostly via pilot projects incrementally. The implementation of each activity included the creation of "cheat sheet(s)" which are one page summaries of each technique(s) employed with examples of the output, and templates where applicable. The effectiveness of the activities introduced to date are listed below in the order of implementation.

Technique	Priority (Stage-Wrk Pkg)	Number of:		Effectiveness		Notes
		Cheat Sheets	Templates	Quality	Mgmt	
Daily Brief	1-2	1	-	Medium	High	Ideal communication
Reflection	1-2	1	1	Medium	High	Good for feedback and morale
Code Inspection	1-3	1	2	V. High	Medium	Immediate roll-out, progress indicator
Code Analysis	1-4	1	-	High	Low	A high number of issues initially
Check In	1-4	1	-	Medium	Low	More of a "given"
Coding Standards	1-4	2	2	Medium	Low	To ensure consistency
Estimation	1-5	2	1	Medium	High	Project and Stage estimates using Wideband Delphi
CUT Complete	1-6	1	1	High	High	Ensures Code Insp, auto-test, code analysis, etc
Test Harness	2-1	1	1	V. High	Medium	Automated testing is seen as a must

6. The People Side

The highest priority cause of development issues identified in the initial brainstorming session was "communication". Snowden's main driver for establishing clear communication channels derived from the agile principle that the focus should be on "individuals and interactions over processes and tools" from the Agile Manifesto [7]. As such, two key aspects were focused on to improve the communication:

1. All developers, project managers and domain experts were involved throughout the Agile Project. This includes working groups, feedback sessions, training sessions, weekly Tec meetings and "Seeds" roles within project teams)
2. Agile techniques that actually facilitate (or at least prompt for) focused communications (e.g. JAD sessions, Work Package prioritisation, design sessions, code inspections, daily briefs and Domain Object Model walkthrough) were selected to form part of the STAg.

The implementation of this communication approach was conducted in the Stage One Work Package named "Communication", and its deliverables included rollout of the Daily Brief, Reflection meetings and Role Identification (including the "Seed" role). Outlining the project roles and their lines of communication through the Role Identification deliverable helped to reinforce each of the related activities.

The established project roles and their relationships are shown in Figure 5, with the grayed/italic roles proposed for later adoption.

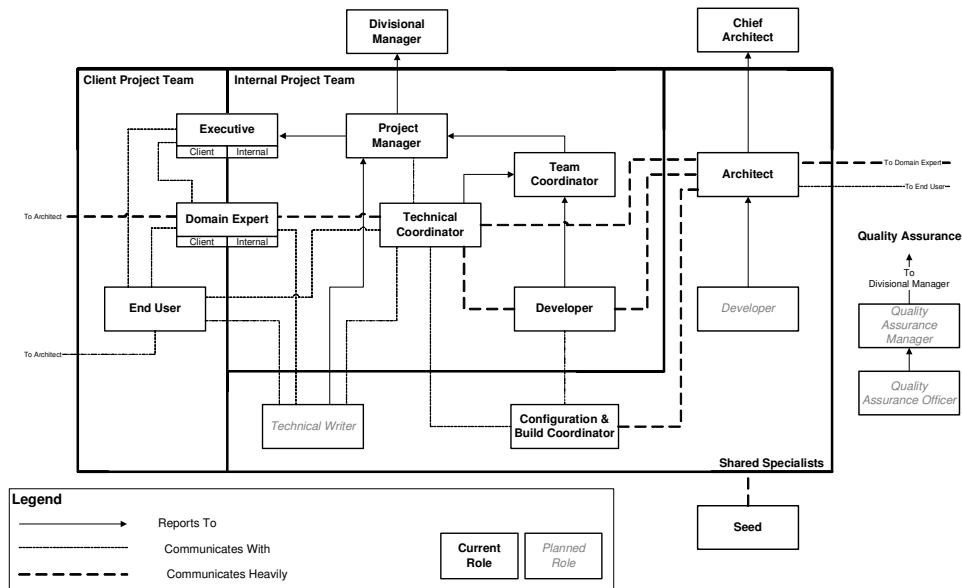


Figure 5: Team Structure - Focused Roles/Responsibilities Helped to Reinforce the Activities

After the initial roll-out of the techniques, a focus group was established to help facilitate the introduction and mentoring of subsequent techniques across the three development sites. This group was named the Technical Operations Management Group or “TOM Group”

7. Tools

Initially, there was a separate improvement project focused on the primary development tool set - Microsoft's Team Foundation Server (TFS). It soon became obvious that the process needed to be the primary focus and that the tools needed to support, automate and reinforce elements of this. As such, the two improvement projects were merged with the emphasis to integrate the applicable process elements into the tools. The following activities (from Figure 4) relate to the use of the tools in the context of the STAg methodology:

- Check In and Check Out
- Test Harness (automated) - TFS has an NUnit style automated test framework
- Code Analysis
- Code Coverage
- CUT Complete
- Continuous Integration Build
- Email Notification of Failure
- Nightly Build
- Weekly Build

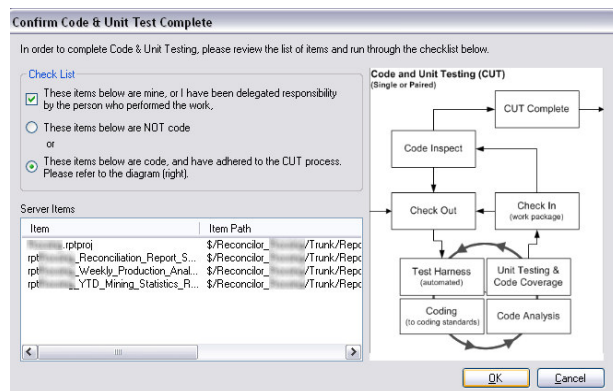


Figure 6: CUT Complete Implemented as an Extension to TFS

- Automated build and test - specs the code and provides the best form of regression testing.
- Daily briefs – a vehicle for good communication!
- Work Packages - frequent closure on subset of scope - good for team morale.
- Static Code Analysis – an integrated tool and process providing easy identification of code problems.
- Reflection sessions - positive reinforcement and adjustments where required.
- CUT complete - tool integration and definitive statement of quality and progress.

8.2. Problems (with some explanation)

- **Communication and Scope Definition** – The implementation of the first Work Package was delayed due to a misunderstanding of the business' expectations of the deliverables by the project team.

A couple of elements caused this situation:

- a. There was no formal briefing or introduction to the projects' expectations and objectives within the team (there was no real "ownership" or "big picture" view of the project);
- b. The project team did not have a clear understanding of the scope of their deliverables to be able to communicate effectively amongst each other and with other developers in the business.

This was resolved initially in the first Reflection session by showing concrete examples of what was required to be produced (templates, cheat sheets, etc) and by providing positive feedback on the direction. Getting closure on the first work packages from each project (which delivered the "road-map") was also an effective reinforcement of the intended approach. It still took another 2-3 weeks before the idea that the group was to actually define the scope of their projects (drawing from industry techniques and methodologies) - but when they did then the true "ownership" began.

- **Information radiators** – The use of Microsoft SharePoint and pin-up boards as information radiators were not as effective as we hoped. They are still useful, but the regular Tec meetings are by far the most effective forum to share the information and seek feedback. In addition, reflection sessions complimented this well. Ultimately, "face-to-face", albeit involving video-conferencing, is the best.
- **Other offices** – The understanding and up-take of techniques in Snowden's other development offices were initially limited. Having only two developers involved in teleconferencing sessions and video recordings of in-house training is now seen as inadequate. One senior developer was involved in pilots, yet due to location was isolated when the sharing of knowledge was paramount. If the "seeds" had been identified and available for mentoring "face-to-face" the techniques would have rolled across offices easier.
- **Alignment with management practices and tools** - The evolutionary development style of iterative development (Work Packages being scoped and planned every 2-3 weeks) was a difficult one for the Project Management planning view to handle. A compromise was developed that required the Work Packages to be identified at the start of the Stage for planning purposes (although this is not considered agile). The (re)prioritization of the later Work Packages would occur to still give the opportunity to ensure the focus is on the highest risk and most important set of features for the next Work Package.

There is concern that the initial Work Package allocation will bias the subsequent prioritization session and therefore lose some of the benefit of this approach (but at least the highest priority elements will still be in the initial Work Packages at the start of the Stage). There was also the alignment with the in-house project management system (SPI) and accounting systems through the integration of reporting requirements and mapping of Stages and Work Packages to a relatively restrictive (and fixed) structure.

8.3. Try

- Create a defined set of objectives for any improvement program (at a Stage level)
- Establish more working groups – although a trade off, and we see it still needs to be selective, but we should (and will) have more working groups to define and refine particular techniques. This was done for the Coding Standards, Code Inspection and Estimation activities and worked well by getting a variety of inputs and ownership.
- Ensure the "Seeds" are more involved in the working groups as they need to "own" and represent the techniques moving forward.
- Increase the communication and visibility to all developers on the progress and direction of the techniques being implemented.
- Establish a continuous improvement environment - this is more an ethos and openness to improve the way we work, but so long as we maintain the Reflection session (and follow through on the agreed actions) then this should be sustainable.
- An exchange program between the team members. There were a few visits by staff from other offices which facilitated a much deeper exchange of knowledge and techniques as well as building up a stronger relationship.

9. Summary

Overall, the approach used so far in tailoring and introducing a selection of agile techniques has worked well. The most fundamental lesson learnt has been to involve as many of the “users” (developers, team leaders, etc) as possible throughout the entire improvement project. It's also been critical that it is done in small, digestible slices (i.e. Iterations / Work Packages) and to seek and give feedback all the way through.

Having the target areas for improvement provided a clearer view for the adoption of particular agile and value add techniques. We selected the techniques from a range of industry agile methodologies, some common, some fairly unique, but all of them to improve our target areas. What has been selected and tailored to date is working and working well. Testimony to this is the very successful delivery of the single and multi-team projects that piloted the initial techniques:

“... it was a very successful trip... big smile on his face and feeling very proud... [the projects] were within 5% of the budget – both delivered on time!”

Rayleen Riske, Project Manager and Geologist

The STAg diagram is a key information radiator providing an ongoing reference point for the context of our development activities on a given project as well as identification of further improvements we wish to incorporate.

The realization that the integration and alignment of our agile software development methodology with the project management methodology and development tools was important as the three perspectives need to complement each other.

Not only have we established a "continuous improvement" mindset, but we have reinforced this through explicitly embedding feedback mechanisms throughout the STAg methodology itself.

10. Acknowledgements

The author's would like to acknowledge the contributions of the following colleagues that have been and continue to be involved in the Agile Project (in alphabetical order):

Jeffrey Alexander

Chris Gilbert

Paul Fox

Mark Holst

Peter Gibbes

Rayleen Riske

and all the developers, team leaders and project managers that have been involved.

11. References

- [1] K. Beck, eXtreme Programming Explained – Embrace Change, 2nd Ed., Addison Wesley, 2004
- [2] Advanced Development Methods Inc., "Controlled Chaos: Living on the Edge", 1996
- [3] S. Palmer, and J. Felsing, A Practical Guide to Feature-Driven Development, Prentice Hall, 2002
- [4] A. Cockburn, Crystal Clear: A Human-Powered Methodology for Small Teams, Addison Wesley, 2004
- [5] DSDM Consortium and J. Stapleton, DSDM: Business Focused Development, 2nd Ed., Addison Wesley, 2003
- [6] J. Wood and D. Silver, Joint Application Development, Wiley, 1995
- [7] K. Beck, et al., "Manifesto for Agile Software Development", www.agilemanifesto.org , 2001