

# AgileTODAY

PERSPECTIVES FOR THE ENTERPRISE INNOVATOR

Volume #1 | MAY 2011



## Why Agile?

Agile finances  
at home

Bankwest goes Agile:  
Project time slashed

Can a project  
manager ask for  
help?

Agile just works

thoughtworks.com

Technical  
Excellence

Better  
Business

Agility

ELIGA

We know it  
backwards.

ThoughtWorks®

Australia | US | UK | Canada | India | China | Brazil

Contact us at [info\\_au@thoughtworks.com](mailto:info_au@thoughtworks.com) or on 1300 723 823

# Contents

- 01 Letter from the editor
- 02 Can a Project Manager ask for help? **Rob Thomsett**
- 03 Celebrating 10 years of the Agile Manifesto
- 03 Spotlight on top topics at Agile Australia 2011
- 04 Why Agile? **Beverley Head**
- 08 Do you wanna be an Architect when you grow up? **Martin Fowler**
- 09 Agile gurus host workshops in Sydney
- 10 It just works **Dave Thomas**
- 15 Bankwest goes Agile: project time slashed
- 16 Win an Apple ipad
- 17 "Even more fun and cruelty" says Agile workshop leader
- 18 Agile Finances @ home
- 20 60 seconds with Craig Smith
- 21 Events on the horizon



## Letter from the editor

Welcome to the first edition of Agile Today!

The Agile movement celebrates its 10<sup>th</sup> birthday this year, and has helped many organisations deliver greater business value faster.

To celebrate, we've put together these articles and ideas from Agile enthusiasts to inspire you on your Agile journey!

We've got Rob Thomsett's views on how to get buy in from your sponsors (page 2), using Agile to sort out your finances at home (page 18), plus an exclusive interview with the technology heads of some of Australia's leading businesses (page 4), and loads more.

If you've got an innovative, thought provoking write-up you'd like to share with us, we'd love to hear from you. Drop us an email at [agile@slatteryit.com.au](mailto:agile@slatteryit.com.au) and you could be featured in the next edition.

For our first edition, we are very excited to have a special give-away for one creative reader. There's an Apple iPad up for grabs, so be sure to check out page 16 for details.

Of course, we've been busily steaming ahead with preparations for the Agile Australia 2011 Conference which is coming up very soon! Make sure you block the dates (15-16 June) in your diary and jump online to register. All the info can be found at [www.agileaustralia.com.au](http://www.agileaustralia.com.au)

I'd also like to say a big thank you to our title sponsor ThoughtWorks; gold sponsors Agile Academy, DiUS, MIP, Rally Software, SMS MT, and IBM; and Silver Sponsor Centrum Systems. We really appreciate your support!

Big thanks to you for opening this magazine and giving it a go. We'd love to hear what you think of it. If you have a comment, or suggestion on how Agile Today could be improved, get in touch with us at [agile@slatteryit.com.au](mailto:agile@slatteryit.com.au). I'm looking forward to your feedback.

I hope this magazine encourages you and helps to refresh your ideas.

Best wishes,

**Zhien-U Teoh**



# Can a project manager ask for help?

By Rob Thomsett

One of the most common topics I get asked about is sponsors and their behaviour.

The typical comments include:  
“My sponsor is too busy. She keeps cancelling our meetings.”

“My sponsor just says JFDI” (for those readers who do not know this acronym, it has something to do with Just F\*\*\*\*\*g Do It).

“If I tell my sponsor the truth, he just yells at me.”

A couple of years ago, my partner Camille gave me the book *Why men can't ask for help and why women can't read maps* by Alan Pease. I clearly have some 'improvement opportunities'.

I read the book with a totally defensive attitude. “I did this. Once. In 1982!” I would exclaim pointing at some accusation in the book such as men not changing the toilet rolls.

Reluctantly, I had to admit that asking for help was something that I wasn't very good at. It has also become clear to me that many project managers suffer from the same psychological flaw.

I was talking with some very senior managers of a client recently. I asked them how the new Agile model was working for them. One of them replied it was going well, and then asked me “What can we (sponsors) do to help?”

This offer of help really raised an interesting problem for me.

I had just had a mentoring session with one of the project managers who had this generous executive as a sponsor on a difficult project.

The project manager had been complaining about the difficulty he was having getting access and attention from the executive.

I went back to talk with the project manager who was very confused. “Look!” he said. “Here is the meeting request in Outlook that he didn't accept! See!”

I suggested the following and it has worked on a number of occasions.

You see a Meeting Request in Outlook is a meeting request. Plain and simple.

*A Request for Help* is a request for help. Plain and simple.

So we changed the Meeting Request in Outlook to a Request for Help and almost immediately the project manager's mobile rang. It was his sponsor asking what he could do to help!

Perhaps all project managers should read Alan Pease's book.

*This article first appeared at*  
<http://thomsett.digeratisolutions.com.au/robblog.aspx>

*Rob Thomsett will be keynoting at Agile Australia 2011*



# Celebrating 10 years of the Agile manifesto



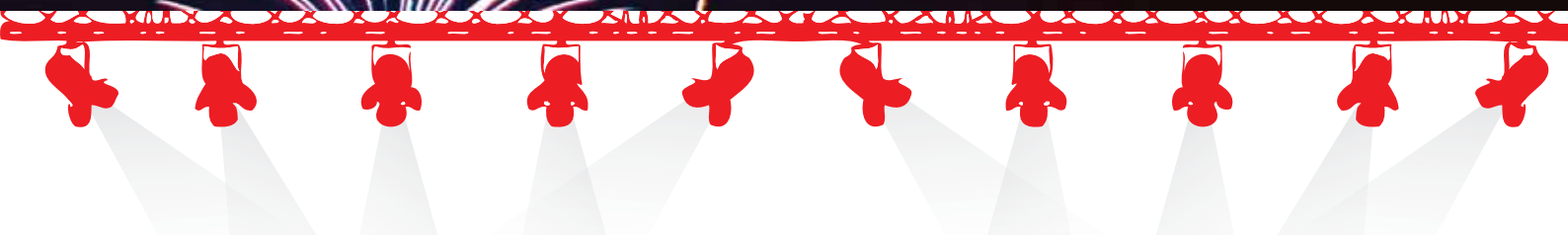
## Agile software development values:

**Individuals and interactions** over processes and tools

**Working software** over comprehensive documentation

**Customer collaboration** over contract negotiation

**Responding to change** over following a plan



## Spotlight on top topics at Agile Australia 2011:



### ALISTAIR COCKBURN

– Co author of the Agile manifesto and ‘Declaration for Interdependence’ Alistair Cockburn will be delivering a keynote address on **‘Beyond Agile: The new face**

**of software engineering’**. Come and hear from one of the founding fathers of Agile development on going beyond ‘just’ Agile, and creating a basis for educating our next generation of developers.



### JEAN TABAKA

– An Agile Coach with Rally Software, Jean has over 25 years of experience in the software development industry and authored

*Collaboration Explained: Facilitation Skills for Software Project Leaders*. Jean will be speaking on **‘Elevating the Agile community of thinkers’**, exploring the power of collective visioning and the slow death of complacency.



### MARTIN FOWLER

– Chief Scientist at ThoughtWorks and co-author of the Agile Manifesto, Martin will be delivering a suite of short talks on patterns, refactoring,

and extreme programming in his session, **‘Software development in the 21st century’**.

# Why Agile?

By Beverley Head



Michael Bromley

**For NBN Co the Agile approach has a particular resonance. It's a company with no legacy and a future still being forged by political fires.**

It needs to move fast, but carefully given that there are plenty of naysayers eager to see it trip. It needs information systems to support fast growth, but the ability to change those on the fly.

For Michael Bromley, the general manager of portals and online services at NBN Co, it's a company crying out for an Agile approach.

Bromley is in charge of development of NBN Co's websites, intranet, ordering and customer portals. He's also responsible for the corporate CRM, enterprise content management, call centre and voice response network.

An investment banker for a decade before moving into the IT&T field, Bromley is quick to point out that he's not a technical guy by trade, and only started to get into development around five years ago. While working at Telstra Bromley found that a waterfall approach just wasn't working given the pace of change in the sector, but like many others found shifting a corporate mindset away from traditional waterfall systems development to Agile isn't always easy as it involves a significant cultural shift.

"To me it's about collaboration. The reason I left Telstra was that I was tired of fighting it. At Telstra I was in the business and I had to fight to be involved in tech delivery."

At NBN Co Bromley has jumped the fence to the technical side, and is now in a position to; "Ask business to join me. This allows IT to move aside from being the whipping boy."

But the business has taken a while to convince he acknowledges. "Most business people expect to fight IT. We said 'we not only expect you to change things, we welcome it'."

Agile's ability to break down legacy thinking, use storyboarding, continuous integration and testing is what makes it such a useful approach for a fast moving start up where new recruits are constantly being added to the business who may want to tweak the operating model.

Bromley admits, "It's challenging because it's a start up moving at break neck speed and very much in the public eye." He feels that he's "Clearly the poster boy for Agile – but by the time it's done I hope to be one of many." He's quick to point though that; "I don't want to make this a religious war. It's just how we deliver solutions to the business that work, faster.

"We bite off little pieces. You don't have to spend \$1 million and wait six months for the business to say 'that's exactly what I asked for, but it's not what I need'." With the iterative Agile approach there is less to throw away when change is needed.

Bromley joined the organisation in April 2010 and in six weeks managed to deliver the first iteration of the website and intranet, working with ThoughtWorks which was at that stage "my delivery arm" according to Bromley. Having started with no staff at all Bromley has grown a team of 62 now, which has added an extra challenge in terms of teaching and coaching those staff, in addition to delivering systems.

# “The beauty of having no culture is that you can start one – we are not dealing with a legacy culture.”

But he acknowledges that; “The beauty of having no culture is that you can start one – we are not dealing with a legacy culture.” While Bromley has managed to find a number of people with Agile experience, he believes the best Agile practitioners are those who have an Agile mindset rather than a fixed set of practices.

NBN is now rolling out the ordering and customer portals. “We have to deliver quickly and iteratively – you can’t have a network if you can’t order and provision it.

“You just couldn’t do this in a truly waterfall fashion. We can’t wait six months to be up and running.”

Speed is also a huge driver for Nigel Dalton, deputy director of digital for Lonely Planet. The company is constantly facing new waves of competition from start-ups, and a travel app developed today can be quickly usurped by an upstart app tomorrow.

While Lonely Planet books may have a two year lifecycle, an app can have a two day lifespan before being knocked down the app list by newer systems according to Dalton.

“We release, try, upgrade and release again,” explains Dalton. “It needs to be OK to fail as long as you learn. Agile is more about the structured stand up, and the retro conversations and the work practices it encourages. We couldn’t have got this through waterfall.”

And proving that the Agile methodology is more than just a technical approach Dalton has even introduced Agile work practices to Lonely Planet’s team of four in-house lawyers who now have an Agile board to plan their workload. It’s an approach which has reformed the way the team works and ensures that the priority work is done by 6pm each day. “Someone puts a new card on their wall,

but also takes one off, rather than the bullshit of saying ‘no you’ll just have to work harder’.”

That’s music to the ears of Rob Thomsett, principal of the Thomsett Company, which specialises in Agile project management. Agile he says should extend beyond the IT department and into general management.

He sees Agile as less of a revolution, and more of an evolution about the way in which projects should be tackled. “The Agile Manifesto articulated what lots of other people had been saying for years – that you had to shift the locus back to the business.

“I’m excited that Agile is past its hippy stage and where people are now taking it seriously.” But Thomsett warns that Agile IT development is only a part of a bigger challenge “It needs to come out from just scrums to the bigger business paradigm,” which is where he is getting traction he says from the likes of NAB, parts of ANZ, Bankwest and the Bank of New Zealand.

Nigel Dalton has been using the approach for around a decade, seizing on it when he ran a San Francisco dot com start up. “I learned the hard way in dot com, where the business model was changing every week. I realised that I was in hell and couldn’t get a business plan written, so I’m going to try it (Agile).”

He acknowledges that there are naysayers to the approach who often argue that Agile has a rubbery governance model. “They say it’s ‘loosey goosey’ governance. I think the exact opposite. The daily stand up explains what you did yesterday, what you’ll do today and what’s blocking you.” That he says is much more active project governance than a typical waterfall approach might offer.



Nigel Dalton

The challenge for the next ten years is talent and the speed of product lifecycle. There's also the question of managing large numbers of Agile teams. Dalton is candid; "I've got 20 developers – but I wouldn't know how to do it with 450."

He also questions the training that there is available. "Everyone's getting Scrum qualified – I term some of them the Scrum zombies," he says, warning that: "Scrum's fine but if you follow it to the letter you won't succeed."

For a company like real estate company REA Group, it was managing success that was the challenge.

Daniel Oertli is the chief information officer at the company which has been using an Agile approach for the last seven years, although Oertli himself only joined two years ago. Initially used for the in-house travel team, the adoption of Agile techniques has gradually expanded more widely.

When Oertli joined the company, although it was clearly a successful business typically growing at rates of 20 per cent plus; "For a dot com business there was a lot of inertia on innovation to introduce more modern compelling products. We had to unlock innovation.

"The gap between our capabilities and our aspirations was so great they (senior management and the IT team) knew something dramatic had to be done."

The bulk of projects had been developed using traditional waterfall techniques and Oertli felt there was a disconnect between IT and the business. Following a consultant's report, Oertli started to introduce Agile approach throughout the organisation and perform an IT group restructure; "To get rid of the fatty management-only layers" replacing them with teams that executed , and customer advocates.

Today the company is "100 per cent agile." That he says has allowed the company to re-platform in just under a year, re-engage with its IT staff, and attract fresh IT talent.



It's ten years since 17 people met in a Utah ski-lodge to find a better way of developing software applications.

That meeting led to the development of the Agile Manifesto and spawned a movement in IT which has gathered pace over the last decade. Agile, with its iterative approach to systems development, is now seen by many organisations as the only sensible way to approach systems development in a fast moving economy.

Traditional waterfall style development, where the software lifecycle steps through concept, analysis, design, construction, testing and maintenance is seen as just too ponderous. By the time the waterfall reaches its destination, the business landscape has changed many argue.

Instead by having IT and business people work together on quick systems iterations, Agile proponents argue they can develop information systems that can be quickly changed in line with a changing business landscape.

In Australia, Suncorp, led by CIO Jeff Smith has been the poster child for the approach since it began rolling out the Agile approach in 2007. Telstra too has been using Agile in some areas of development and increasingly the big banks are exploring the approach.

The most forward looking companies are also taking the Agile approach outside of their IT shops and into general management.

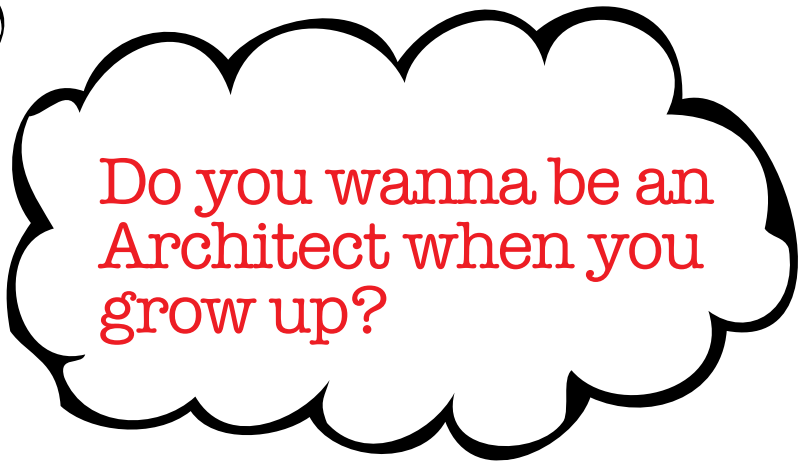
A decade on from the manifesto, there are signals that Agile is increasingly being adopted by organisations of all shapes and sizes, and for a wide range of applications.

-Beverley Head

Daniel Oertli



"As a leader, the role is not to prevent risk but to build the capabilities so we can recover when failures occur."  
- Jeff Smith, CEO Suncorp Business Services, Agile Australia 2010



**MARTIN FOWLER – Chief Scientist, ThoughtWorks**

(excerpt from article: Is Design Dead? At martinowler.com)

**For much of the last decade, the term “software architect” has become popular. It’s a term that is difficult personally for me to use. My wife is a structural engineer. The relationship between engineers and architects is ... interesting. My favourite was “architects are good for the three B’s: bulbs, bushes, birds”. The notion is that architects come up with all these pretty drawings, but it’s the engineers who have to ensure that they actually can stand up. As a result I’ve avoided the term software architect, after all if my own wife can’t treat me with professional respect what chance do I stand with anyone else?**

In software, the term architect means many things. (In software any term means many things.) In general, however it conveys a certain gravitas, as in “I’m not just a mere programmer - I’m an architect”. This may translate into “I’m an architect now - I’m too important to do any programming”. The question then becomes one of whether separating yourself from the mundane programming effort is something you should do when you want to exercise technical leadership.

This question generates an enormous amount of emotion. I’ve seen people get very angry at the thought that they don’t have a role any more as architects. “There is no place in XP for experienced architects” is often the cry I hear.

Much as in the role of design itself, I don’t think it’s the case that XP does not value experience or good design skills. Indeed many of the proponents of XP - Kent Beck, Bob Martin, and of course Ward Cunningham - are those from whom I have learned much about what design is about. However it does mean that their role changes from what a lot of people see as a role of technical leadership.

As an example, I’ll cite one of our technical leaders at ThoughtWorks: Dave Rice. Dave has been through a few life-cycles and has assumed the unofficial mantle of technical lead on a fifty person project. His role as leader means spending a lot of time with all the programmers. He’ll work with a programmer when they need help, he looks around to see who needs help. A significant sign is where he sits. As a long term ThoughtWorker, he could pretty well have any office he liked. He shared one for a while with Cara, the release manager. However in the last few months he moved out into the open bays where the programmers work (using the open

“war room” style that XP favours.) This is important to him because this way he sees what’s going on, and is available to lend a hand wherever it’s needed.

Those who know XP will realise that I’m describing the explicit XP role of Coach. Indeed one of the several games with words that XP makes is that it calls the leading technical figure the “Coach”. The meaning is clear: in XP technical leadership is shown by teaching the programmers and helping them make decisions. It’s one that requires good people skills as well as good technical skills. Jack Bolles at XP 2000 commented that there is little room now for the lone master. Collaboration and teaching are keys to success.

At a conference dinner, Dave and I talked with a vocal opponent of XP. As we discussed what we did, the similarities in our approach were quite marked. We all liked adaptive, iterative development. Testing was important. So we were puzzled at the vehemence of his opposition. Then came his statement, along the lines of “the last thing I want is my programmers refactoring and monkeying around with the design”. Now all was clear. The conceptual gulf was further explicated by Dave saying to me afterwards “if he doesn’t trust his programmers why does he hire them?”. In XP the most important thing the experienced developer can do is pass on as many skills as he can to the more junior developers. Instead of an architect who makes all the important decisions, you have a coach that teaches developers to make important decisions. As Ward Cunningham pointed out, by that he amplifies his skills, and adds more to a project than any lone hero can.

*Martin Fowler will be keynoting at Agile Australia 2011*



# Agile gurus host workshops in Sydney

**International leaders in Agile thinking, Jean Tabaka and Alistair Cockburn will be hosting half-day workshops for the Australian Agile community.**

The workshops are an opportunity to gain in-depth learning in a small group setting from renowned experts.

An Agile Fellow with Rally Software, Jean Tabaka will be using a combination of presentation, games, simulation, and discussion in her workshop, *Lean, Kanban, and the Art of Flow*.

“It doesn’t matter what you’re producing,” says Jean, “when efficiency and workers’ time utilisation become more important than the value of the product, the production system is optimised for the wrong thing.”

A workshop designed to answer the fundamental questions on the why, how, and what of Kanban, Jean helps workshop attendees understand the affects of flow, impact of sub-optimisation, and what she calls the “deadly cost of delay.”

Jean’s workshop is a must for those looking to improve their team’s processes.

Mature Agile practitioners will enjoy the *Introduction to Advanced Agile Development* workshop, lead by co-author of the Agile manifesto Alistair Cockburn.

Alistair goes beyond collecting tips and tricks in this tutorial,

and challenges participants to examine why particular strategies work and how to analyse situations to derive solutions.

“The Agile community is mature enough by now to take this next step,” says Alistair.

“Computers must support the way in which people naturally and comfortably work,” he says.

“I am committed to speeding this development by improving human communities, human-to-computer communication, and efficient, humane working methods.”

Alistair will review five sections of theory about designing in teams, then help small groups co-analyse problems and solutions.

Workshops catered for those just starting the Agile journey will also be available, providing a practical, hands-on learning experience for those new to the Agile approach.

With highly regarded workshop leaders Craig Smith and Marina Chiovetti, participants will learn what is meant by the term ‘Agile’, the key practices that go along with it, and what it’s like to be part of an Agile team.

Workshops for beginners, *Lean, Kanban and the Art of Flow* and *Introduction to Advanced Agile Development* take place on Tuesday 14 June 2011 at the Hilton, Sydney. For more info or to register visit [www.agileaustralia.com](http://www.agileaustralia.com)



# It just works

By Dave Thomas

The majority of improvements in software development tools and techniques focus on new applications or components for greenfield projects. Unfortunately, this means that organisations with assets developed using older, once popular technologies and methods cannot easily migrate to new applications or components. Much has been written about using Agile/XP methods and greenfield development as an alternative to big M methods. However, software experts increasingly see that Agile methods are also well suited to software evolution beyond only greenfield development. This presents a much more important opportunity for applying Agile methods. This isn't completely surprising since Agile methods stress the importance of skilled people, risk reduction, and continuous testing – factors which all contribute to effective software evolution.



Dave Thomas

## **Why is maintenance so challenging?**

In practice, the quality of the software really depends on the quality of the developers, rather than on the specific software technology. Unfortunately, studies that illustrate this seldom get published, however experienced software executives know the value of high quality developers. It is yet another example of the reality of Brooks [1] observation, “You ship your organisation”.

In our experience, the entropy of a software asset increases substantially after its third to fifth version. The entropy is a function of increased KLOCs, and making changes upon changes. The loss of key people over time also contributes greatly to the complexity of code maintenance. Despite the best documentation practices, typically only a few key people understand the inner workings of most systems.

For many years, some of the best software managers observed the following rule: “Try to develop your way out of maintenance after the third version”. Experience shows that defects tend to cluster into specific modules and that these modules often remain problematic despite efforts to repair and enhance them. Managers believed it was more effective to replace the defective module with a redesigned one than to repeatedly attempt to patch it.

In many companies, there is still an attitude that new development is difficult and needs the top developers, perhaps even outside developers. The less experienced or less skillful developers are then assigned to perform the system evolution. There is often also little training or mentoring allocated to the evolution effort. Most companies assume that on the job training will suffice. The standard practice is to promise that a developer assigned to an evolution project can only move to a new project once they have trained a usually less experienced replacement.

## **OO doesn't ensure easy maintenance**

While much has been written about OO reuse and the ability to maintain and enhance OO applications, there is little documented evidence to prove this is true. OO enables reduced maintenance effort, but doesn't ensure it.

Many legacy OO applications often contain complex frameworks, which are very difficult to maintain and extend. OO documentation, just like previous structured methods, is often left at design or version 1 level and is never maintained due to the pressures of evolution. OO libraries can be difficult to understand [2]. OO programs rely on collaborations between objects, which are derived from use cases, scenarios or CRC cards. Unfortunately, many of these collaborations are not well documented, particularly those involving third party libraries and/or programmer defined implementation objects.

Some languages, like COBOL and Smalltalk, have been designed to be very readable and self-documenting. Others, such as Java and C#, have been extended via inline

tags and tools such as JavaDoc to provide minimal literate programming support. However neither of these approaches is sufficient. Unlike simple TP monitors, J2EE applications can be complicated and difficult to read. Even though these complications are clearly reduced if a standard J2EE application framework is used, these applications still leave complex legacies.

**Though you walk through the source code Tree of Death**

The older the code base, and the more mission critical the application, the more difficult it is to maintain or enhance it. The software takes on the qualities of a dangerous unexplored cave, complete with cryptic symbols in design notation and scrolls in older dialects of programming languages. The rumor mill warns of dangerous “rooms” in the software where few developers have succeeded in making changes that work. In order to reduce the risk of failure, the bug fixes and change requests for such modules are slowed to a crawl and only the bravest developers dare to make major changes. While some of the original development team and their expertise remain, their knowledge may also be out of date. Unfortunately, sometimes the expert that remains is either unable to communicate the essentials of the system or chooses to become a roadblock to system evolution.

Given these challenges, it natural for managers to consider a fresh approach.

**Maintenance as Agile Evolution**

Recently several organisations have been considering Agile techniques for their maintenance process. They quickly identified many similarities between Agile concepts and their own software maintenance experiences as shown in the table below. The Agile development process includes the tools and techniques required to effectively deal with the common issues organisations face when maintaining software.

Software Maintenance	Agile Development
Understanding the essence of the system	Metaphor and Stories
Customer defect and feature requests	Customer and Stories
Test suites	Test first, Unit test
Regression testing	Continuous integration
Fixes and “Dot” Release	Small Increments
Change Management	Planning Game, Stand Up Meetings
Reverse Engineering, Debugging, Critical Patches	Pair Programming
Code Reorganisation, Code Simplification	Refactoring

**Legacy Tales**

How can Agile development address the challenge of passing on an understanding of a complex software system? Many systems have lots and lots of use cases, massive requirements and design documents and diagrams. These documents often fail to communicate the essence [1] of the system to developers responsible for system evolution. A second problem is that existing documentation is not up to date and often has gaps in important areas. One of the best ways to develop a shared essence is through real storytellers who have lived part of their lives ‘in’ the system. Recently there has been a great resurgence in the use of story telling as means for conserving corporate memory [3].

For many years, young recruits that joined companies like Nortel and IBM were taken through intensive boot camps during which they were exposed to massive amounts of source code along with the wisdom of the key architects or developers. These

“The benefits... include the ability to provide feedback, increased confidence with estimates, reduced risk and improved quality.”

developers communicated the essence of the systems to new developers. While the stories they imparted were imprecise and sometimes inaccurate, they frequently provided sufficient context to enable new developers to work on the company's products. They learned quickly where the minefields lay, why the system didn't work the way they thought it should, or even work the way it was documented.

This essential understanding of how the system works is still best communicated by knowledgeable storytellers to inquisitive learners who are climbing difficult code mountains. The 'big story' is what XP calls 'Metaphor' and the 'little stories' are story cards or use cases. These practical descriptions slice through the system exposing its essence. Passing on an understanding of the essence of the system is an important part of software maintenance that can be facilitated through using Agile development.

#### **Customer Change Requests**

One of the challenges in new development is finding the right customer and building the right thing. This is a critical success factor for the organisation and is stressed in XP. It can be a particular challenge when building a shrink-wrapped product for a general external market where customer requirements can often only be derived from surrogate customers, focus groups and beta customers.

In addition to defining the requirements for new development, Agile development can address the challenge of requirements analysis for established products. An established product has lots of customers who have identified defects and new features, which they feel are missing from the product. The customer feedback is systematically collected through problem tracking systems in which customers prioritise the important problems and features. These change requests are used to drive the evolution of the system. Unlike initial requirements, which are often only very high-level problem reports, feature requests are usually very specific. In many organizations, customer focus groups are used to represent the customer by organising and prioritising development activities in collaboration with developers. Agile evolution therefore has the key ingredients for development stories and engaging in the planning game with the customer.

#### **Pair Programming – Share the risk, and the return**

Effective software maintenance requires a way for developers to reverse engineer, debug and apply patches in order to make critical software updates. Agile developers can effectively use Pair Programming in these situations. Note that Pair Programming in general refers to more than just coding, and includes design reviews, test cases and all development and deployment activities.

Software developers working on large existing systems are often required to make changes to software they don't understand. They need to reverse engineer legacy code to identify the defective modules, determine which modules need to be changed and determine the order in which to make the changes. Given the uncertainty, these reverse engineering and debugging activities are often done by pairs of developers. For example, mission critical applications changes often need to be made to a system running live in a customer's location.

A common practice among seasoned maintenance developers is to approach such critical changes in pairs, relying on the human redundancy to reduce error. Similarly, design and code reviews are well known ways to eliminate bugs by 'staring them out'. The benefits of an additional pair of hands and eyes include the ability to provide feedback, increased confidence with estimates, reduced risk and improved quality.

#### **Regression testing and Continuous Integration**

Regression testing is a key factor in effective software maintenance. The use of Agile development allows for more frequent regression testing through continuous integration. For many years, strict regression testing has been the mantra of a successful maintenance organisation. Recent efforts in the Agile community to develop efficient continuous integration approaches can easily be embraced by regression testers to allow them to increase the frequency of regression testing through the use of dedicated test servers, mock data bases, and so on.

### **Maintenance as a development activity – Incremental reengineering and development**

*“The most amazing thing was that I learned as much in a maintenance month three years into the project as I did in a programming month before release. In fact, I came to think of the practices we’d put in place as a safety net that made maintenance a safe place to learn. The very best objects in the product weren’t discovered, couldn’t even be discovered, until we had the full richness of the (Agile) maintenance environment in place.” - Ward Cunningham [4].*

How can Agile development be used to deal with large systems that have well known modules, which are known to contain a substantial number of defects? Historically these modules are the least understood by developers hence each new fix or feature is approached with great trepidation. In some cases it may be difficult to make timely releases due to concerns about touching the core components of the system. The time-honored solution to these problems is to incrementally replace the faulty components – one component at a time. This approach is called developing your way out of maintenance.

The XP approach to this problem is to develop stories and then test cases to ascertain the correct operation of this component. Typically these test cases will include existing regression test cases as well as new test cases identified by developers or customers. Once one has sufficient test cases, changes can be safely made to the defective module or it can be replaced without fear of unknown side effects. It is important that both management and developers gain confidence with this approach and move slowly, first making changes in single functions or data structures rather than making wholesale change to whole classes or hierarchies.

### **Refactoring to reduce defects and ease changes**

Code reorganisation and simplification is an important software maintenance activity that can be addressed through refactoring in Agile development. In general, any activity that substantially reduces the number of lines of code and/or improves the readability will reduce defects and facilitate future changes. This is one of the primary goals of refactoring which seeks to reduce duplication of code, simplify complex code and introduce improved names as well as class and method organization. While these changes can be implemented with an editor, I strongly recommend industrial strength refactoring tools if they can be obtained. These tools support the developers in making and unmaking small changes and reduce the risk associated with a refactoring effort. One, of course, should not attempt major refactorings unless the test cases provide full coverage of the component to be refactored.

### **Agile Evolution - A positive alternative approach to software maintenance**

The essence of Agile evolution is to gradually transform a typically conservative, risky and unattractive activity into a positive and proactive Agile development activity. The impact of Agile style practices on the evolution of large software systems may be even more important than its impact on new application development. By adopting similar practices throughout the life cycle, the schism between sustained development and maintenance is reduced. The human customer-centered values of the Agile movement can change “blue collar” maintenance to “white collar” development, just as the JIT revolution in manufacturing changed assemblies into quality circles.

*Dave Thomas will be delivering a keynote address at the Agile Australia 2011 conference.*

[1] Frederick P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, Anniversary Edition (2nd Edition)

[2] Dave Thomas, *The Deplorable State of Class Libraries*, *Journal Of Object Technology*, [http://www.jot.fm/issues/issue\\_2002\\_05/column2](http://www.jot.fm/issues/issue_2002_05/column2)

[3] Adele Goldberg, *Story Telling: Collaborative Software Engineering*, *Journal of Object Technology*,

[http://www.jot.fm/issues/issue\\_2002\\_05/column1](http://www.jot.fm/issues/issue_2002_05/column1)

[4] Ward Cunningham Personal Communication, <http://fit.c2.com/wiki.cgi?WhatsWhat>



# Tech23

*Celebrating innovation in 2011*

**Tuesday 23 August 2011**

**The Auditorium, 37 Reservoir Street, Surry Hills**

Tech23 2011 is *THE* event for those wanting to get a handle on the most innovative and fast-growing tech companies in Australia.

Join the industry's brightest as 23 young Australian companies show off their company to an audience of potential investors, entrepreneurs, and customers. If you want to be amazed by the hot new innovation in areas like mobile, green, digital, health and education, Tech23 is the event for you.

**Understand the bleeding edge of Australia's future, network with the biggest names in the industry, and be part of making great things happen.**

platinum sponsors



silver sponsors



TECH23 2011  
INDUSTRY LEADERS



**Dr Michelle Deaker**  
OneVentures



**Todd Forest**  
ninemsn



**Martin Hosking**  
RedBubble



**Lars Rasmussen**  
Facebook



**Deepak Natarajan**  
Intel Capital



**Matt Rockman**  
Co-founder of SEEK

[www.tech23.com.au](http://www.tech23.com.au)

On September 17th, 2010 in ZDNet.com.au, David Braue wrote 'Bankwest goes agile: project time slashed' after hearing Andy Weir's presentation at Agile Australia 2010.

# Bankwest goes Agile: Project time slashed

**Bankwest began its shift to Agile development in 2008, when a \$2.1 billion buyout by the Commonwealth Bank of Australia forced it to quickly improve its governance processes and execution capabilities.**

Bankwest's answer was to step away from its traditional "waterfall"-styled project methodology, and instead put together teams of developers and business managers to deliver projects through a series of weeks-long "sprints".

Adoption of this methodology allowed Bankwest's developers to significantly speed the delivery of a debt management project that was built earlier in the year and

eventually included more than 59,000 lines of code. The project went through three iterations, with a heavy emphasis on problem identification that saw more than 900 unit and functional tests run over 2100 times. This approach allowed the team to identify and remedy 106 defects during development, with a further seven identified in a subsequent sprint and remedied well before the final product was delivered.

By following the lean, accelerated schedule, Bankwest was able to deliver the project in just 15 weeks, compared with the 13 months that it was estimated the project would have taken using a conventional approach. This has been so significant that the company has recently enshrined Agile development as one of its six core business principles: within the next three years the company is aiming to deliver fully half of its change portfolio using Agile methodologies.

To read David Braue's full article and to learn more about some of the surprises and challenges the agile approach presented to Bankwest, please visit

<http://www.zdnet.com.au/bankwest-goes-agile-project-time-slashed-339306091.htm>



“We delivered the project in just 15 weeks compared with the estimated 13 months”  
– Andy Weir, CIO, Bankwest

# Win an Apple iPad

Taking notes at the Agile Australia 2011 conference will be a breeze with a funky Apple iPad. Flex your creative muscles because this sexy tablet could be yours.

For your chance to win, send us your most creative caption for this photo. The winner will be announced at the Agile Australia 2011 conference.



To enter:

1. Come up with a creative, quirky, colourful, or comic caption for the photo
2. Email it to [agile@slatteryit.com.au](mailto:agile@slatteryit.com.au) with **CAPTION** in the subject line
3. Be sure to include your name, job title, company, email address and phone number
4. Be at the Agile Australia 2011 conference to receive your prize!  
(register at [www.agileaustralia.com.au](http://www.agileaustralia.com.au))

Competition closes 15 June 2011.





# “Even more fun and cruelty” says Agile workshop leader

The hugely popular Red Bead experiment workshop will be reprised at Agile Australia 2011.

Back by popular demand, this famous learning tool explores the impact of an organisation’s ‘system’ on individuals in a software team.

The experiment asks volunteers to simulate an organisation producing white beads; ensuring defective red beads are removed.

The exercise sees the volunteers participate under different management strategies including slogans and posters, financial rewards, and annual appraisals.

Agile Coach David Joyce leads the workshop and takes the role of Project Manager, while colleague Jane Nguyen plays Product Owner.

“It will be my job to get results,” jokes David, “and Jane’s job to ensure I deliver!”

Originally performed by Dr W Edward Deming, the workshop introduces some of his 14 Points for Management and

Deadly management Diseases and Obstacles.

The final part of the workshop shows how to identify and remove the red beads found in an organisation.

“The organisational system is constraining you and is your largest impediment!” David says.

“[The workshop] will contain advice you can take into your organisations and make your teams even more productive.”

This year, the workshop will have a full 90 minutes of airtime. Those who participated last year reported valuable learnings.

“Very entertaining and very informative,” commented one delegate.

“For those of you who attended the shortened session last year,” says David, “this will be a longer session with even more fun and cruelty.”

“If you work as part of, or manage, software development teams, you will not want to miss this.”



# Agile@Home – Finances!

By Jonathan Coleman

**Let me begin by telling a story. It's got little to do with software, builds, or business value. However it does have a lot to do with motivation, inspiration and gratefulness!**



“We had astonished ourselves.”

In our household, there are two adults and three kids. We live off one income, and have done for the last eight-and-a-half years. We've tried all sorts of things to keep our day-to-day budget under control – everything from pencil and paper adding up, to complicated spreadsheets that only a third generation statistician can understand, to nothing at all (short lived), back to paper and pencil, to little envelopes, to credit cards, to waving our hands in the air, etc etc.

We get the job done with the weekly budgeting, the bills get paid, family gets fed most times, etc.

But what, I hear you cry, about all the 'extras' – the nice to haves, the mini projects, the anchovy stuffed olives that make life worth living?

#### **Motivation:**

Well we were fed up and grumbling to each other that neither of us was getting what we wanted – although we deep down knew we had what we needed. I really wanted a motorbike, and my wife was insistent that a dishwasher was more important!

What the? I thought one evening after doing the budget together. I know how we could do this!

What is it that helps us stay inspired at work? What is it that allows us to see progress? What allows us to deliver on our promises in a short timeframe?

And in a fit of card cutting and column drawing – our Agile finance board system was born!

#### **Executing:**

We taped up the back of a door into four columns: Must-haves, Should-haves,

Nice-to-Haves and “Split to regular payments”. Next to this, we had a ‘Kanban wall’ made of two columns: ‘In progress’ and ‘Done’. We used the three principles of Kanban: Make work visible; limit work-in-progress; and help work to flow.

Everything from cosmetic dentistry, extra text books, motorbike restoration project, fits of clothes shopping, and the new dishwasher went onto the wall.

We prioritised the Must-have column first, and then plonked the rest of the cards into the remaining columns.

At this point we stopped. We still didn't know quite how it would work, but we left it at that point.

#### **Delivering on the promise!**

The next week, at budgeting time, we made sure there was cash in the right accounts, bills were paid – staying one step ahead of the shoe-shine as the song goes.

And then we looked at the top priority ‘Must-have’ card. It just happened to be a new dishwasher to replace the one that had died.

We put in place a savings plan for this, and stuck the dishwasher on our ‘Kanban’ wall. The ground rules? Only one project at a time – and it had to be the highest priority thing that we'd both agreed on.

We then threw our available spare money at this project, measuring velocity using an old fashioned colour thermometer graph (remember those??) on the fridge. When some bonus money came in we put that towards our highest priority project.

MUST HAVE



SHOULD HAVE



COULD HAVE



# Agile AUSTRALIA'11

Building tomorrow's leading enterprises today!

**WHAT** Agile Australia 2011: 3rd annual Australian conference on Agile software development  
**DATE** Wednesday 15 - Thursday 16 June 2011  
**VENUE** Hilton Sydney  
**REGISTER** [www.agileaustralia.com.au](http://www.agileaustralia.com.au)

We had astonished ourselves. Before long a new dishwasher turned up at our door, was installed, and we weren't fighting our way through stacks of plates anymore!

We then had a celebration ceremony – moving the 'dishwasher' card to DONE, and floating the next one to the top of the stack.

### Inspiration!

Months later, we look at our 'Done' pile and the list of done items is slowly growing. We have something to be grateful for. Our needs and wants are getting met – albeit slowly – but with fun and gratefulness mixed in.

### What's in it for me??

I am ecstatic to report I got that motorbike, with the appropriate training, gear, and new parts, and plenty of learning out in the shed, and now I am happily riding this beast into work. Hugely satisfying!



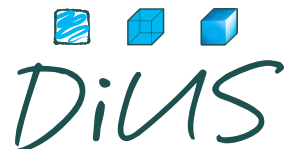
View the original post at [www.theagiletribe.net](http://www.theagiletribe.net) or [www.scrumology.com](http://www.scrumology.com)

Jonathan Coleman will be delivering a lightning talk on this topic at Agile Australia 2011.

## TITLE SPONSOR

# ThoughtWorks®

## GOLD SPONSORS



## SILVER SPONSOR



SPLIT TO

# 60 seconds

with Craig Smith



**With 15 years of software development and eight years of Agile practice under his belt, Craig Smith is an experienced and vocal advocate of the Agile methodology. He has regularly spoken at both the global and Australian Agile conferences, and currently spends his days as an Agile Coach at Suncorp's Agile Academy.**

Craig is a Certified Scrum Master, a member of the Scrum Alliance and Agile Alliance, an advisor to Agile Australia, and will be speaking at Agile Australia 2011.

**Everybody starts their Agile journey somewhere. What was your 'a-ha!' moment?**

My a-ha moment was in the days before many folks were even calling it Agile in 2001 – 2002. I worked on a project to write a lending application written in Java where we overtook a small meeting room, started writing tasks and designs on a whiteboard, split designing screens down via CRUD and core functionality and we paired and worked as a team to get things done. I could never go back after that.

**What has been your greatest challenge when introducing Agile to an organisation? How did you overcome it?**

In the early days it was trying to get people to take you seriously, as not delivering reams of documentation at the start of a project was seen like being a cowboy yet we were delivering faster than the teams around us. It felt much like working in a bubble because when we went outside our team environment we had to fall back to the waterfall processes used by the rest of the organisation. When Jeff Smith joined Suncorp, it was refreshing that someone in higher management had similar views, and since that point it has been a challenge to find approaches to make our IT teams (and now the entire organisation) to work more effectively.

**What is your favourite Agile-related quote?**

I am always having to remind people that “our job is not to do quality Agile, it is to deliver quality software or solutions”. We just use Agile values, principles and practices to help us do that. I am quite concerned how much the term Agile is overloaded or used as an excuse by many people now, so have started a movement to come up with a new label, and joked we should call it “raccoon”. (in hindsight I should have come up with a better name!)

**What is the strangest situation you've applied an Agile principle to?**

It's amazing how many situations the core practices of stand-ups, retrospectives and Big Visual Charts are applicable to. I find it more amazing that when getting together to plan work with other Agile coaches or working on different Agile conferences, how often I have to remind people to visualise their flow or do a reflection on progress.

**If you could have a total career change, what would you be?**

I never set out to work directly in IT, as I did a dual IT – librarianship degree at university. Part of me still wants to tick that box at some stage. But if I could find a job that I had the skills for that related to my love of motorsport, that would be awesome.

**What is your favourite thing on your desk right now?**

I don't have a desk, so I live out of a backpack (one of my colleagues calls me “the turtle” because I carry my desk around). So when I do find a real desk, a power pack is usually pretty good. As for the cool stuff, I have a Spongebob Squarepants and a bunch of Simpsons characters on my desk at home!

“We were delivering faster than the teams around us.”

# Events on the horizon

**Agile Australia 2011:  
Sydney – 15-16 June, 2011.**

Attend Agile Australia 2011 and discover how leading businesses are adopting an Agile approach to stay ahead! Over two days, Australian and international thought leaders will explain how this innovative methodology has helped their organisation to deliver exceptional business value. [www.agileaustralia.com.au](http://www.agileaustralia.com.au)

---

**Agile 2011:  
Utah, USA – 8-12 August, 2011.**

Back to the Future — 10 years of Agility and Beyond. It has been 10 years since the signing of the Agile Manifesto in Snowbird, just outside Salt Lake City, Utah. Much has happened in the world of software development and agility in those 10 years since. Agile2011 returns to Utah Salt Lake City to rediscover some of our roots and to try to discover new ways of improving software development. <http://agile2011.agilealliance.org/>

---

**iAwards Gala Dinner and  
National Awards Presentations:  
Melbourne, 4 August, 2011.**

iAwards Gala Dinner and National Awards Presentations – Melbourne: Thursday 4 August, 2011. Recognising excellence in the IT&T industry. Network with the Australian ICT community and celebrate with the award winners. [www.iawards.com.au](http://www.iawards.com.au)

---

**Tech23 2011:  
Sydney - 23 August, 2011.**

Tech23 2011 is THE competition for young, innovative and fast-growing Australian companies. 23 young Australian companies will be selected to show off their company to the 400+ potential investors, mentors, entrepreneurs, and customers. Come along and be part of making great things happen! [www.tech23.com.au](http://www.tech23.com.au)

---

**Web Directions:  
Sydney – 11-14 October, 2011.**

Web Directions conferences cover the full range of interests for web professionals — web design, front-end and back-end development, information architecture, interaction design, accessibility, data visualization and more. <http://www.webdirections.org/about/>

---

**YOW! 2011 Developers Conference:  
Melbourne – 1-2 December, 2011  
and Brisbane – 5-6 December, 2011.**

The YOW conference is a unique opportunity for you to listen to and talk with international software experts in a relaxed setting. <http://www.yowconference.com.au>

ALISTAIR COCKBURN

MARTIN FOWLER

JEAN TABAKA

ROB THOMSETT

# Agile

AUSTRALIA '11

*Building tomorrow's leading enterprises today*

Only in Sydney 15-16 JUNE

TITLE SPONSOR

**ThoughtWorks®**

GOLD SPONSORS



SILVER SPONSOR



[www.agileaustralia.com.au](http://www.agileaustralia.com.au)