

# Critical Decisions in Agile Adoption

Steve Hayes  
Cogent Consulting Pty Ltd

<http://www.cogentconsulting.com.au>

This is Brad



<http://www.flickr.com/photos/oter/3316795815/>

Brad is a Agile coach and consultant

Brad is offered a gig at Ponderous  
Software Development

Ponderous want to become agile

Brad gives Ponderous his “Agile 101” presentation, and they love it

They ask Brad to coach their  
adoption

However, Ponderous can see that  
agile as Brad described it clearly  
won't work for them...

Because they are different!

They have some complexities that  
other people don't have, so...

Brad can do whatever he wants,  
except...

He can't change anything about  
operations or the production  
environment

(different department)

He can't have access to the business  
people

(they're too busy)

Every project needs a business case  
accurate to +/- 10% before analysis  
(CFO requirement)

Projects must have fixed costs,  
fixed scope and fixed delivery date  
before development starts  
  
(business requirement)

All the requirements need to be  
documented to ISO-666 before  
development starts  
  
(audit requirement)

The process needs to be identical  
across all teams  
(QA requirement)

Developers can't access the  
internet

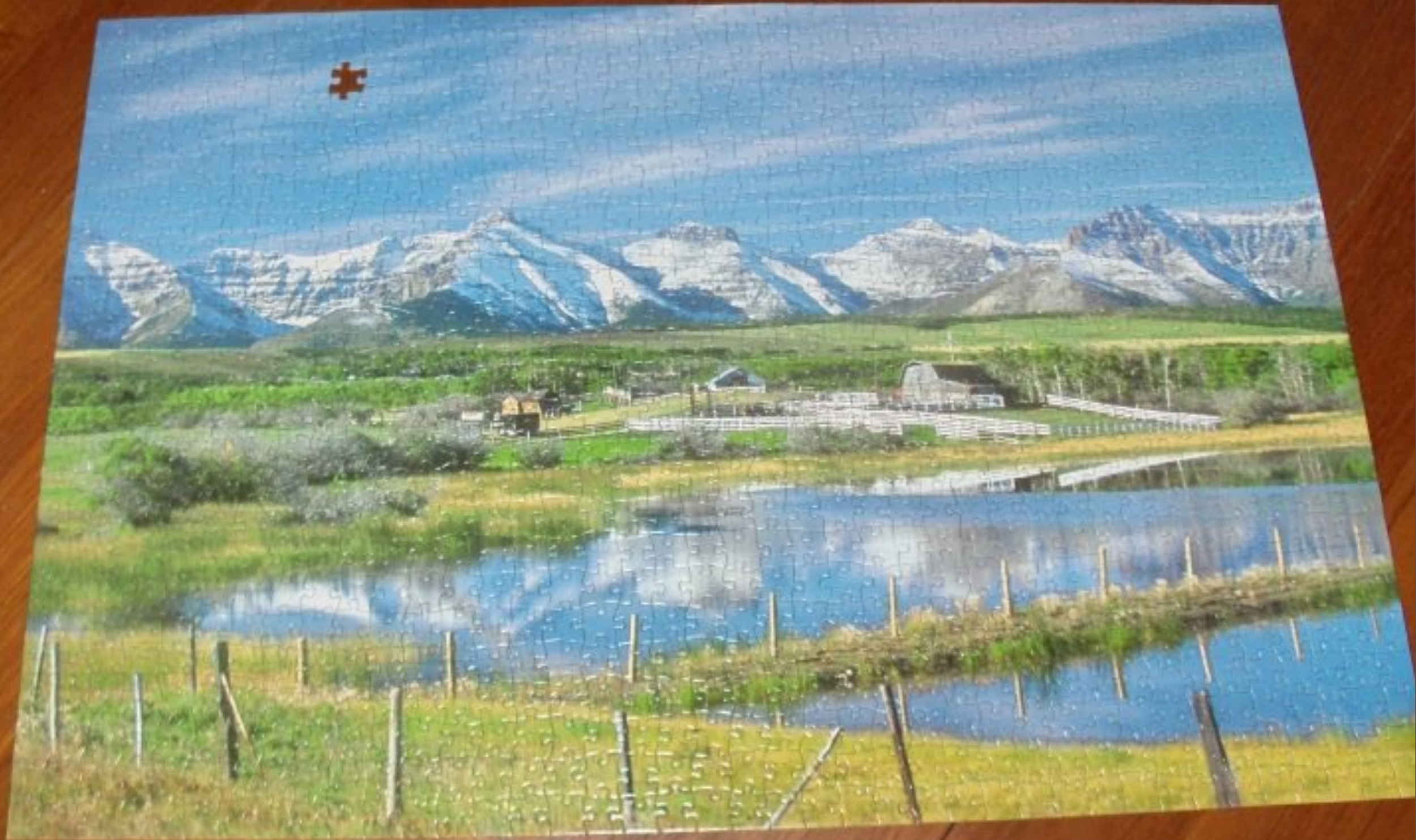
(security requirement)

He can't post project information  
on the walls

(facilities requirement)

He can't spend any money on  
hardware or software  
(budget constraint)

Development must be in a new  
language, with no developers  
experienced in that language  
  
(architectural requirement)



08/27/2006

<http://www.flickr.com/photos/barbanddean/227027259/>

# Hayes' First Law of Agile Consulting

Everyone thinks they're special

# Hayes' Second Law of Agile Consulting

**They're not**

(but treat them as if they are)

# Josh Kerievsky - Agile Readiness Assessment

# Summary

You actually need to be  
willing to change

**Brad takes a break from  
Ponderous, and talks to Inspired  
Business Solutions**

Brad gives Inspired his “Agile 101”  
presentation, and they love it

They ask Brad to coach their  
adoption

Except...

They say IT is resistant, so they  
don't want to change development  
practices

(different department)

They've heard of this thing called  
Scrum...

...and they want to start by getting  
the business people Scrum certified

This can work, but...

“There's a mess I've heard about with quite a few projects recently. It works out like this:

- They want to use an agile process, and pick Scrum
- They adopt the Scrum practices, and maybe even the principles
- After a while progress is slow because the code base is a mess”

Martin Fowler - Flaccid Scrum

Jeff Sutherland ... says he has never seen a Scrum team become hyper-productive without adopting the XP practices.

quoted by Ron Jeffries

Traditional methods are “shrink to fit”

Agile methods are “stretch to fit”

Scrum is the lightest agile method,  
and requires more stretching than  
most

**Retrospectives are critical in  
Scrum, and need to include the  
“whole team”**

# Summary

Improvement is  
constrained by technical  
practices

Eventually, Brad manages to get  
everyone in Inspired Business  
Solutions into a common  
retrospective



The developers think the customer  
is doing a bad job

The testers think the developers  
are leaving in too many defects

The customer thinks everyone else  
lacks focus

Everyone thinks operations sucks  
(operations aren't in the retro)

This is not a single team

XP and Scrum and Agile are not guaranteed to succeed, and they aren't the only way to succeed. The only way to succeed — other perhaps than catching a really lucky break — is to build a team who work well together and who get things done.

Ron Jeffries

Mantra: fix testing - double velocity.  
Fix Product Owner - quadruple  
velocity. Fix team ...

Jeff Sutherland, on Twitter

# Summary

You probably need a social  
transformation

Brad meets another coach, Max, for a drink, and they talk about testing

This is Max



<http://www.flickr.com/photos/aknacer/2820654956>

Brad's team is using a lot of  
automated testing

automated unit tests (with 98%  
coverage)

automated functional tests (driving  
the browser)

the testers specify the functional tests they need, before a story is developed

the developers automate the test  
specifications

testers do exploratory testing  
story by story

regression testing is done at the  
touch of a button

testers often need to make changes  
to test specs, and repeat  
exploratory testing

Max's environment is a little bit  
different

the developers still have some  
automated unit tests

but their ETL tool doesn't support  
versioning or unit testing

the developers automate functional  
tests

the testers have their own scripts,  
that they execute manually

they don't start testing until the  
end of the iteration

and regression testing is manual - it  
takes days, or weeks

testers never waste any effort -  
they don't start until other people  
are finished

# Summary

Testing needs to be  
integrated into the entire  
development process

Testability needs to be  
part of all technical  
decisions

The process shouldn't be  
optimised for a single role

Let's look at one large company for  
our last example

Could be a bank, could be an oil  
company, could be something else

The have 300 developers, organised  
into about 50 teams

“The Process” needs to be the  
same for every team

A team discovers something that works better for them - they want to move their retrospectives to  
Tuesdays

They need to get this approved by  
the Process Review Committee

And every team needs to make the  
same change

What do you think happens?

On a more serious (technical) note  
- what if everyone needs to use the  
same dependency injection  
mechanism?

# Summary

The team needs to own  
the process

# Extreme Programming

Has some planning practices

Has some social practices

Has significant technical practices

Ignore them (any of them) at your  
peril

Agile : Extreme Programming

Nite Owl : Rorschach

Alan Francis, on Twitter

# Final Summary

# Summary

- You actually need to be willing to change
- Improvement is constrained by technical practices
- You probably need a social transformation
- Testing needs to be integrated into the entire development process
- Testability needs to be part of all technical decisions
- The process shouldn't be optimised for a single role
- The team needs to own the process