

How your choices affect your agility

Steve Hayes

Cogent Consulting Pty. Ltd.

steve.hayes@cogentconsulting.com.au

Where did we start?

“Leave your campsite cleaner than
you found it”

“I will prescribe regimens for the good of my patients according to my ability and my judgment and never do harm to anyone”

“From each according to his ability,
to each according to his need”

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas



This was a marketing document

it was a deal with the business, with obligations and benefits on either side



permission

The time for permission is past

The original manifesto doesn't
provide guidance on execution

“Values are what keep us on the straight and narrow path in the face of temptation. Teams that have strong internalized values will stick to good Agile practices — and get good Agile results — while teams without guiding values will drift into the ditch.”

Brian Marick

Beck

- Transparency
- Responsibility
- Accountability



Marick

- Courage
- Working Software
- Ease-of-work
- Reactive
- Fast Feedback
- Visibility
- Naiveté
- Joy

Ease of work (ready-to-hand vs
present-to-hand)

“A characteristic of high-performing agile teams seems to be how much they value ease of work”

What do these tell us?

Pick tools and practices that are
compatible with these values!

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...

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)

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)

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)

What do you think of Brad's
chances?

Brad's agile implementation will
help things

but not as much as it could

Context

Context is your set of constraints

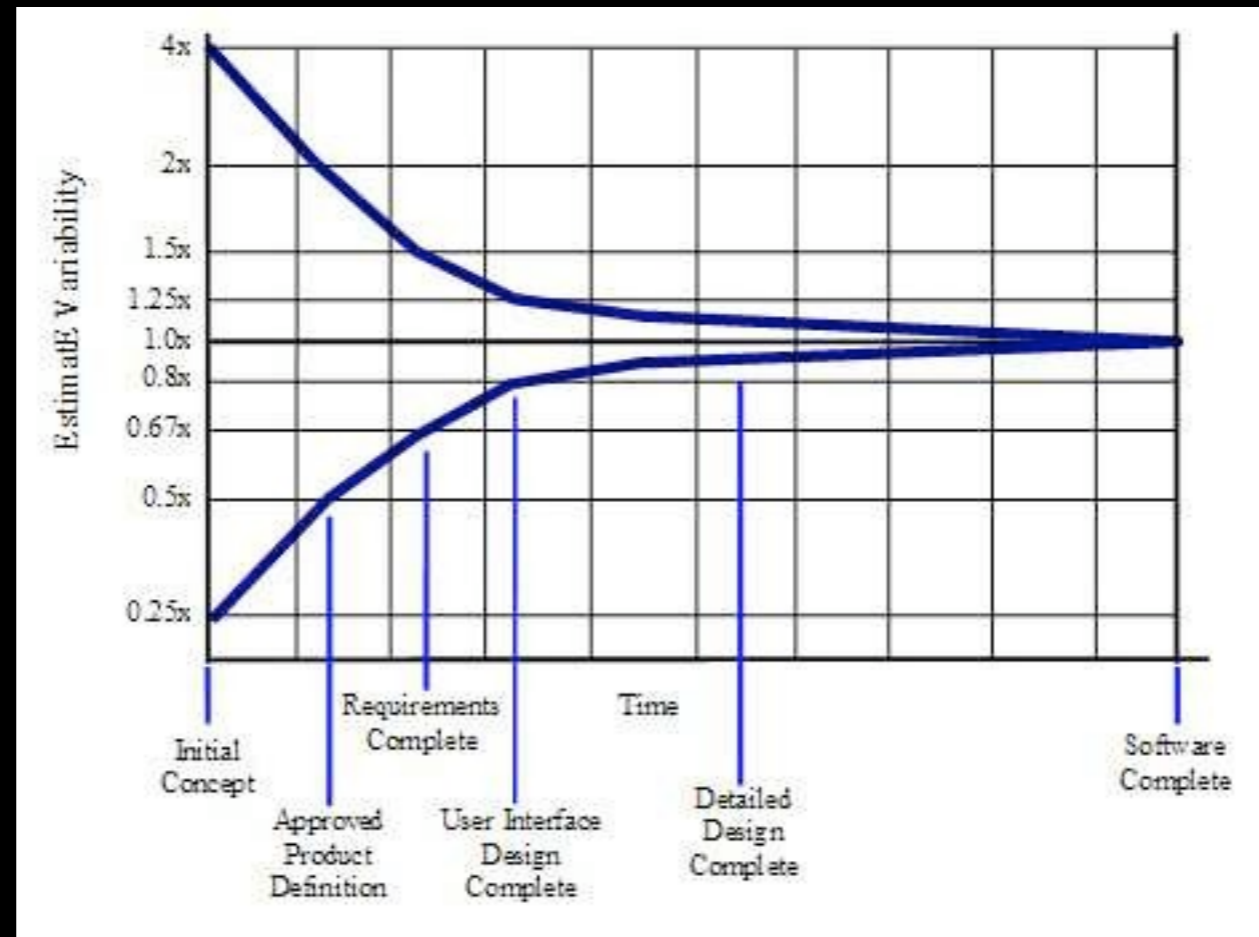
the more constraints you have, the
fewer chances for optimisation



Why will Brad struggle?

**Axiom - software development is a
creative process**

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



Steve McConnell's Cone of Uncertainty

Portfolio approach to product initiatives

Michael Hammer ('Reengineering the Corporation')

Minimal initial investment,
get some feedback

Getting enough detail to cost to
“10%” is going to cost money

And it's money you can't spend on
software later

Emotional investment makes it hard
to be reactive

All the requirements need to be
documented to ISO-666 before
development starts

(audit requirement)

Projects must have fixed costs,
fixed scope and fixed delivery date
before development starts

(business requirement)

Do you know what you want in 6 months? In detail?

Are there no opportunities to learn
anything along the way?

Makes feedback irrelevant

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

(different department)

Typically this means scheduled,
manual deployments

Difficult to maintain visibility and
receive feedback

Interrupts ease of work

Hard to be reactive

He can't have access to the
business people
(they're too busy)

Can't get fast feedback

Can't be naive

The process needs to be identical
across all teams

(QA requirement)

Indicates the team isn't in control -
generally reduces joy

Hard for the team to be reactive

Team can't have the courage to
implement process changes based
on fast feedback

Developers can't access the
internet

(security requirement)

Ease?

Doesn't encourage technical
courage

Not much point getting fast
feedback if you can't act on it

He can't post project information
on the walls

(facilities requirement)

Well, visibility sucked anyway

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

Sends the message that ease of
work isn't important

Development must be in a new
language, with no developers
experienced in that language

(architectural requirement)

Might increase joy if the team made
the decision

Probably reduces it though

Depends if the team can reject that
choice based on feedback/
experience

Context can kill

WRONG

WAY

GO BACK

Why are all these constraints in place?

**SPEEDED
KILLS**

Manufactured by Life Safety Signco, Inc.

SC000000-10000

But going too fast is just one way
to lose



**VELOCIDADE
MÁXIMA**

Risk management is not the same
as risk elimination

Agile is about going as fast as we
can

And reducing the cost of falling off

Let's look at another example

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

How do Max's problems relate to agile values?

The ETL tool doesn't support
versioning or unit testing

Was there an explicit decision that
having this tool was more
important than having automated
tests?

the developers automate functional
tests

the testers have their own scripts,
that they execute manually

Displays a lack of courage

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

but there's no fast feedback either

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

“The difference between the possible and the impossible lies in a person's determination.”

Tommy Lasorda (baseball coach)

<http://www.runwayapp.com>