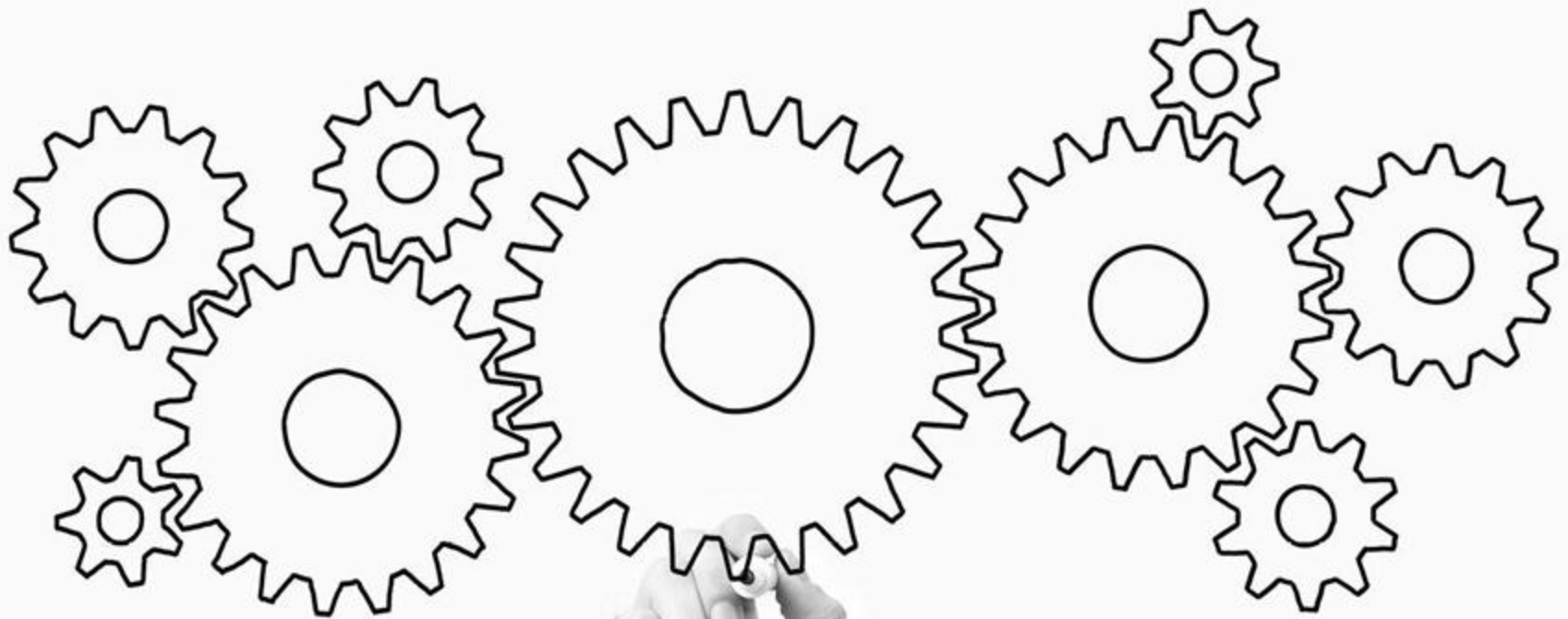


# Agile Contracts: Building Trust



Ewan Milne  
ewan@calenture.org



# Contracts: a necessary evil?

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.

© 2001, see <http://agilemanifesto.org>

# Contracts: a necessary evil?

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.

© 2001, see <http://agilemanifesto.org>

# Contracts: a necessary evil?

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.

© 2001, see <http://agilemanifesto.org>

# Let's blame the lawyers!

Lawyers think differently from us.

Creatures of habit, guided by the tried and tested.

Trained to be distrustful.

- If not of other parties, at least of anticipated outcomes.

Have a duty to think the unthinkable.

Often view software projects as sequential and relatively predictable processes.

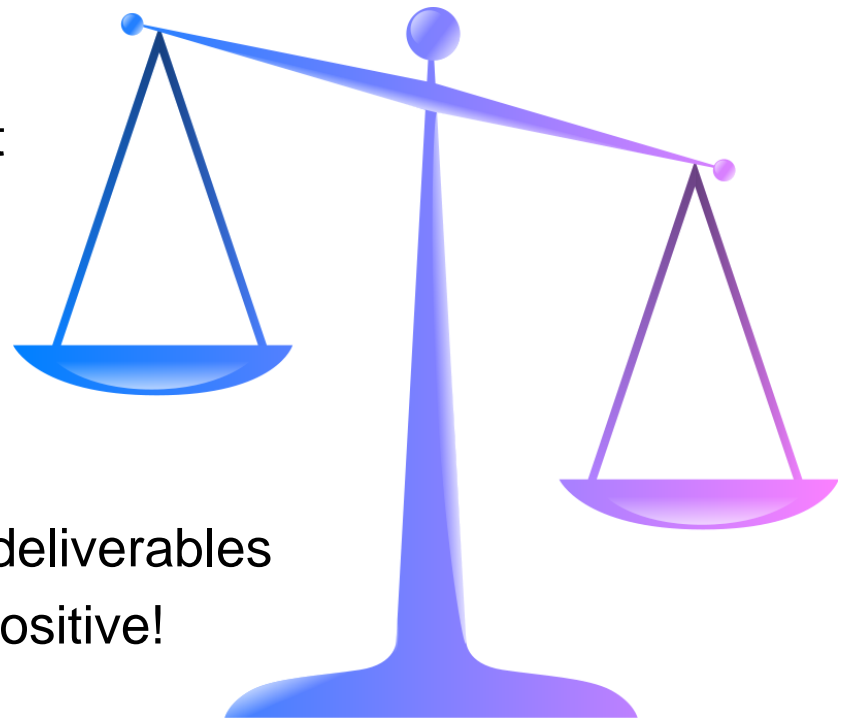


# Lets work with the lawyers

Include the legal staff in your cross-functional team.

Educate lawyers in agile development:

- How agile can reduce risk
- How iterative schedules can affect the delivery and payment cycle
- How adaptive planning can replace traditional change management
- How a goal can drive project, eliminating prescriptive lists of deliverables
- How early termination can be positive!



# Let's work with the customers

Try to engage customers with relational contract models where goals are aligned, and a relationship based on trust and support can develop.

Ensure that you establish a contract which:

- manages risk appropriately,
- provides flexibility to allow for change,
- provides clarity with regard to obligations and expectations for both parties.



# Fixed Price

Very "popular" form of contract.

Designed to protect the customer by transferring risk to the supplier... in theory.

- Fixed Price,
- Fixed Schedule,
- Fixed Scope.

Change Control: RFCs, Off-Specs.

Chance of failure even if the terms of the contract are fulfilled.

Actively erects barriers to collaboration and trust.





# Time and Materials

The *other* commonly used contract type.

Acknowledges uncertainty.

Moves risk from supplier to customer.

Does not incentivize supplier  
to be efficient  
(actually, quite the reverse).

*Can* be used as the contractual  
basis for agile development.

...but still fails to establish  
a collaborative relationship  
based on trust.



# Framework Agreements

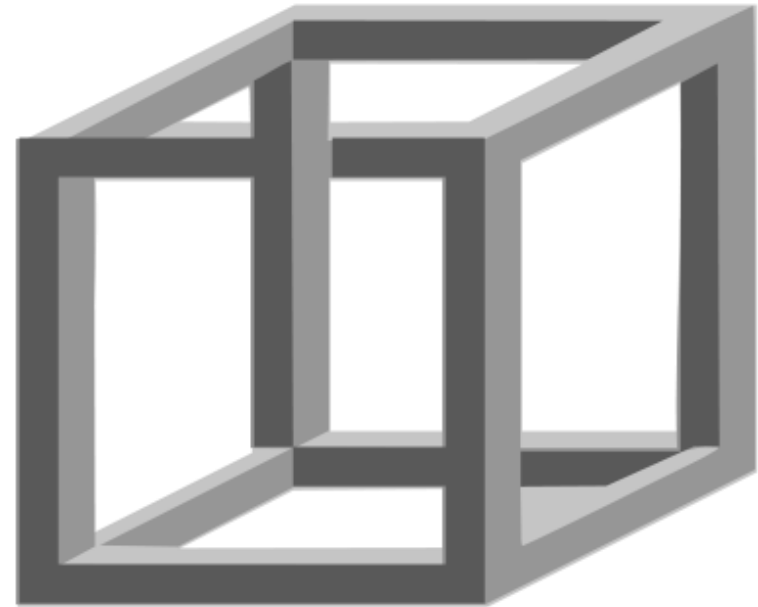
Agreement between customer and supplier, defining general terms for specific contracts over period of agreement.

Common in public sector procurement.

Establishing a relationship between customer and suppliers.

However, multi-supplier frameworks tend to lead to lengthy procurement processes.

EU procurement rules are ***not*** compatible with agile approach.



# Framework Agreements

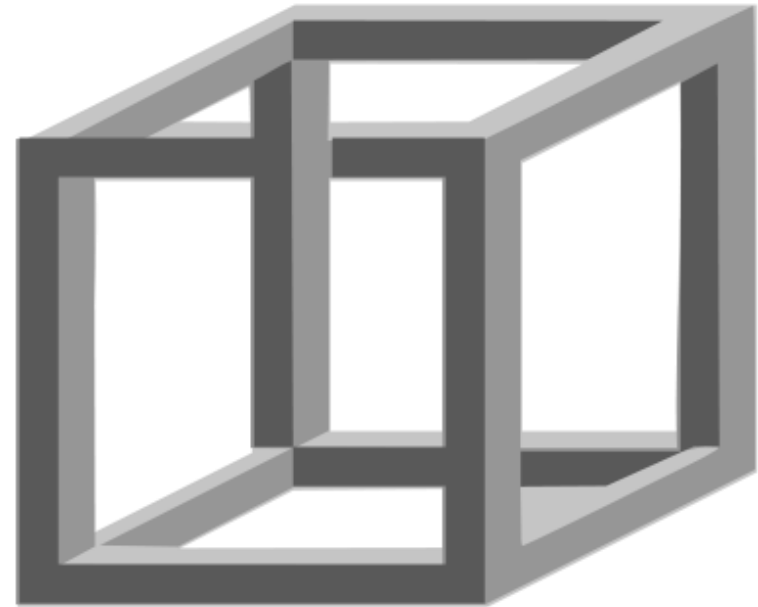
UK government's proposed **GDS Digital Framework**.

Focuses on Agile development.

Available from "middle of this year".

Details still emerging: may be run by a "neutral vendor".

Agile is "flavour of the month" in government procurement at the moment, commitment still to be proven.



# Incremental Delivery

One way to address inherent risks of Fixed Price.

Fixed Price per iteration.

Easy to adapt to agile development.

Incremental delivery, regular payment.

But also, frequent opportunities for either party to abandon.

Bilateral Monopoly?

Latter risk mitigated if highest priority features are delivered first.



# Price per Story Point

Directly bakes Story Points into contract.

Short initial period to set up backlog, calibrate velocity.

Supplier decides the points.

Customer sets priorities, pays on acceptance.

But as Allan Kelly has it, "Just Say No!".

Goodhart's Law; gaming of the system.

Misuse of points-based estimation.

Not a very good way to build trust.



# Target Cost

Options covered so far fail to align the incentives of customer and supplier.

Joint effort implies shared risk, and both parties striving for a common purpose.

We want contract to act as platform to build partnership.

Target cost contract makes total cost a joint responsibility.

Define target cost, and therefore price.

Any costs above target shared fairly.

Any savings under target also shared.

“Shared pain, shared gain”.



# Money For Nothing, Change For Free

Proposed by Jeff Sutherland.

A variation on Incremental Delivery, with addition of two clauses to specifically support agile development.

- Money for Nothing
- Change for Free

Start with “a standard fixed price contract, which includes time and materials for changes”.

Customer and supplier required to follow Scrum rules, otherwise contract reverts to time and materials.

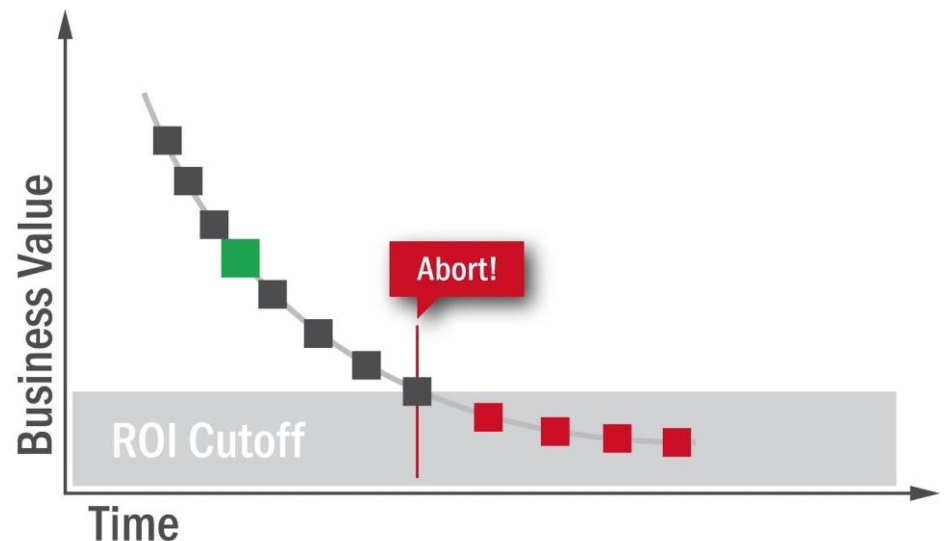


# Money For Nothing

Customer decides on ROI cutoff, when proceeding to next sprint does not offer sufficient business value.

Supplier receives cancellation fee based on remaining profit they would expect from remaining work.

- Supplier is cushioned from effects of early termination.
- Customer does not pay for work that they no longer want.



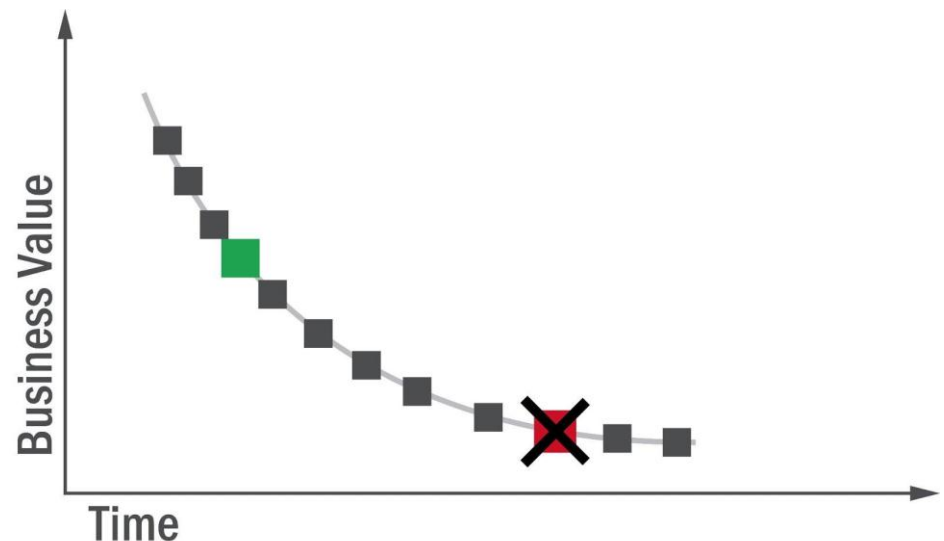


# Change For Free

Product Backlog is reprioritised at end of each iteration.  
Changes can be included at this point.

New features added for free, if low priority items of equal size removed from Product Backlog.

- Allows the ability to respond to change by altering scope within contract terms.



# Optional Scope

Proposed by Kent Beck and Dave Cleal back in 1999.

Return to the Iron Triangle: Fixed Price means

Fixed time, fixed cost, fixed scope ∴ variable quality

Whereas Optional Scope Contract proposes

Fixed time, fixed cost, fixed quality ∴ variable scope

But if requirements are not part of the contract, what is delivered?

A product with guaranteed quality, and genuinely valuable features agreed in collaboration.



# An example from elsewhere: The T5 Agreement

Heathrow Airport Terminal 5: Opened in March 2008.

The opening itself was, frankly, a fiasco.

Multiple problems with operational systems led to chaotic scenes, widely reported in the media.

500 cancelled flights, BA losses of £16m.

A "national humiliation" said MPs at the time.

How very British.

Terminal's construction:

- ✓ cost £4.3bn,
- ✓ came in under budget,
- ✓ completed on time.

[theguardian](#)

Passengers fume in the chaos of  
Terminal 5's first day

Flights cancelled and baggage system collapses at BA's £4.3bn showpiece

---

Dan Milmo, transport correspondent  
The Guardian, Friday 28 March 2008

# An example from elsewhere: The T5 Agreement

BAA realised from the outset that a project of such scale required a break from traditional procurement.

To this end, they developed the T5 Agreement drawing from Lean principles:

- based on relationships and behaviours
- exposing and managing risk rather than transferring it to external sources
- “Success is in everyone’s interest”



# Summing up

Do not just settle for Fixed Price – it's not the only game in town.

Collaborate with your legal staff and your customer or supplier.

Accept that software development is a dynamically complex activity.

Explore the options available to you.

Establish an agreement which aligns interests, and allows the development of a relationship based on trust.

$$2 + 2 = 5$$

# References

Mary Poppendieck, Tom Poppendieck. *Lean Software Development: An Agile Toolkit*. Addison-Wesley, 2003.

Craig Larman, Bas Vodde. *Practices for Scaling Lean & Agile Development*. Addison-Wesley, 2010.

Jens Coldewey: Contracting Agile Projects.

([www.cutter.com/content-and-analysis/resource-centers/agile-project-management/sample-our-research/apmu0617.html](http://www.cutter.com/content-and-analysis/resource-centers/agile-project-management/sample-our-research/apmu0617.html))

Allan Kelly: Agile Contracts

([www.infoq.com/articles/agile-contracts](http://www.infoq.com/articles/agile-contracts))

Allan Kelly: Points based contracts? Just Say No.

([allankelly.blogspot.co.uk/2012/05/points-based-contracts-just-say-no.html](http://allankelly.blogspot.co.uk/2012/05/points-based-contracts-just-say-no.html))

Sourcing For Agile

([www.emergn.com/insights/blogs/insights/thought-papers/sourcing-for-agile](http://www.emergn.com/insights/blogs/insights/thought-papers/sourcing-for-agile))

Jeff Sutherland: Agile 2008 - Money For Nothing

([scrum.jeffsutherland.com/2008/08/agile-2008-money-for-nothing.html](http://scrum.jeffsutherland.com/2008/08/agile-2008-money-for-nothing.html))

Kent Beck, Dave Cleal: Optional Scope Contracts

([xprogramming.com/articles/optional-scope-contracts](http://xprogramming.com/articles/optional-scope-contracts))





**Thank you for listening.  
Any questions?**