

FINAL VERSION



Smarter business
through IT transformation

How much quality can we afford?

Allan Kelly, allan@allankelly.net

<http://www.allankelly.net>

Agile Business Conference 2010



Allan Kelly

» **Training & Coaching for Agile adoption and deepening**

» **Author:**

- *Changing Software Development: Learning to be Agile*, Wiley 2008.



97 Things Every Programmer Should Know,
Henney, 2010

*Context Encapsulation in Pattern Languages of
Program Design* volume 5, 2006





Quality, not Qualities



1970's Leyland Mini

- Rusts quickly
- Doesn't start well
- Engine floods
- etc. etc

1970's Rolls Royce

- Spacious
- Leather upholstery
- Low MPG



2000's BMW Mini

- Starts first time
- Engine just works
- Doesn't rust
- Nice to drive

Images from Wikipedia: Rolls-Royce public domain from Bull-Doser; Minis creative commons licenses, DeFacto (Leyland), BMW (IFCAR)



Quality without Gold-plating



Fit for purpose

- » No rework
- » Free of bugs
- » Features which work
- » Fewer features make for more usability
- » Maintainable
- » Knife through butter testing

No over engineering

- » No unused features
- » No “reusable” code
- » No “that would be cool”
- » No half baked ideas



Can you afford to reusable code?

Single Use



- Break even on third (re)use
- Profit on fourth (re)use
- How much of your code is (re)used four times?

“Reusable” - costs three times more

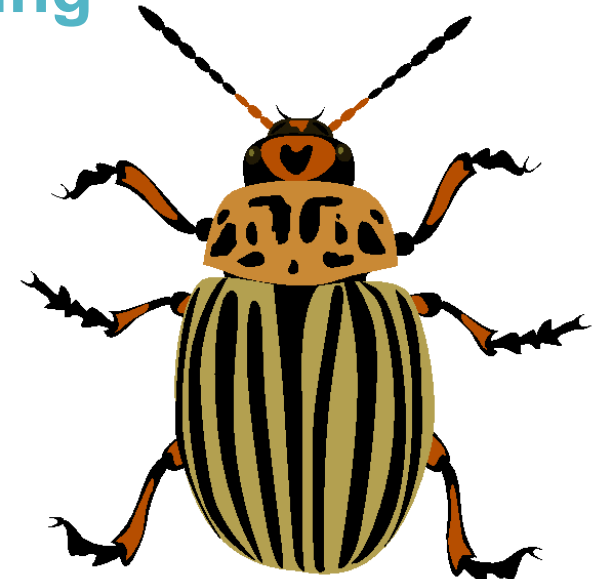


**Reuse
Myth**



Bugs

- » How much time do you spend finding bugs?
- » How many testers do you need?
- » How many bugs do you have logged?
- » How many bugs do you fix before shipping?
- » How much time do you spend in meetings discussing bugs?



How would your life change if there were no bugs?



Quality is Free™

Philip
Crosby 1980

- » Semi-conductors
- » Missiles
- » Cars
- » Etc. etc.

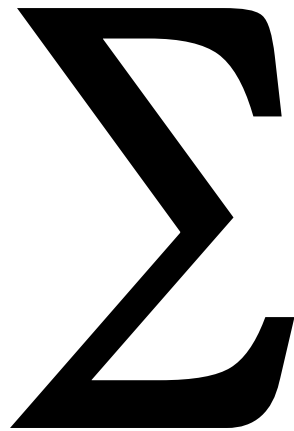
Quality is basis of Lean, and
Lean is the basis of Agile

Are we in danger
of forgetting
quality?

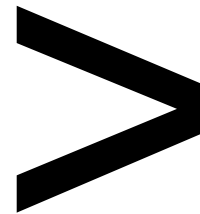




An Old Idea



Original work
Finding defect
Scheduling fix
Fixing
Retesting
Customer
inconvenience
Schedule
disruption



Extra work
to prevent it



Agile without quality?



» How do you know you are done?

» How do you time box?

- How do you eliminate Test-Fix cycle?

Agile without
Quality is like
Starbucks without
Coffee



Starbucks image © Louis Abate, Creative Commons License, c/o



Follow the Logic

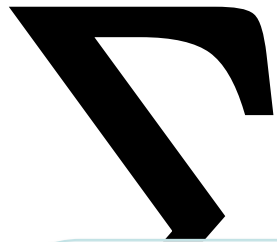
- » **Without quality you need test-fix**
 - With test-fix you can't close an iteration
- » **If you can't close an iteration you can't be done**
 - Thus Iterations (Time-boxes) fall apart
- » **Without time-boxes delivery becomes random**
 - People retreat to plans and demands



How is this different to the old world?



Old idea – why didn't it work?



Original work
Finding defect
Scheduling fix
Fixing

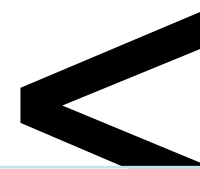
Retesting

Customer

inappropriate

Disruption

disruption



Extra
work to
prevent

defects

Old solutions made defect prevention very expensive

Copious documentation, Heavy weight code reviews, Manual testing

And very very slow

Detracted from ability to respond (Agility)



Old Idea, New Tools

- » Invest in quality
- » Make defect prevention cheap
 - Continuous integration
 - Virgin install
 - Test Driven Development
 - Acceptance Test Driven Development
 - Lightweight code-reviews
 - Pair programming
 - Static analysis tools



**Feedback
based**



Unit Testing on Steroids

Automated TDD is to Traditional Unit Testing
what Amazon is to Great Universal Stores





TDD works

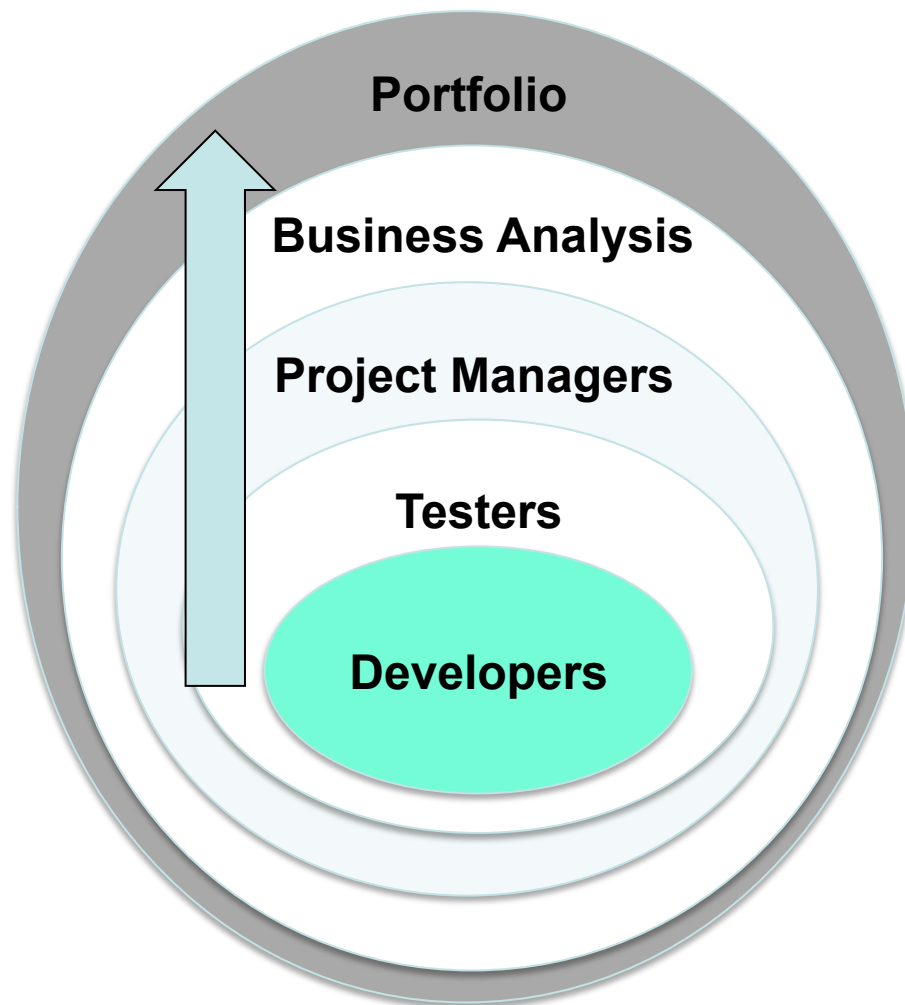
	IBM drivers	Microsoft Windows	Microsoft MSN	Microsoft Visual Studio
Defect density (non-TDD)	W	X	Y	Z
Defect density (with TDD)	61% of W	38% of W	24% of Y	9% of Z
Increased time (with TDD)	15-20%	25-25%	15%	25-20%

Nagappan, Maximilien, Bhat and Williams (Microsoft Research, IBM Research, North Carolina State University). Empirical Software Engineering journal 2008

http://research.microsoft.com/en-us/projects/esm/nagappan_tdd.pdf



Agile change model



Drive quality

- 1 Interest Developers
 - Improve quality
- 2 Enroll testers
- 3 Refocus Project Managers
 - Deliveries over plans
- 4 Change Business Analysis
 - Goals over shopping lists
- 5 Change Portfolio parameters
 - Delivering value over following plan
6. Realign Project Managers

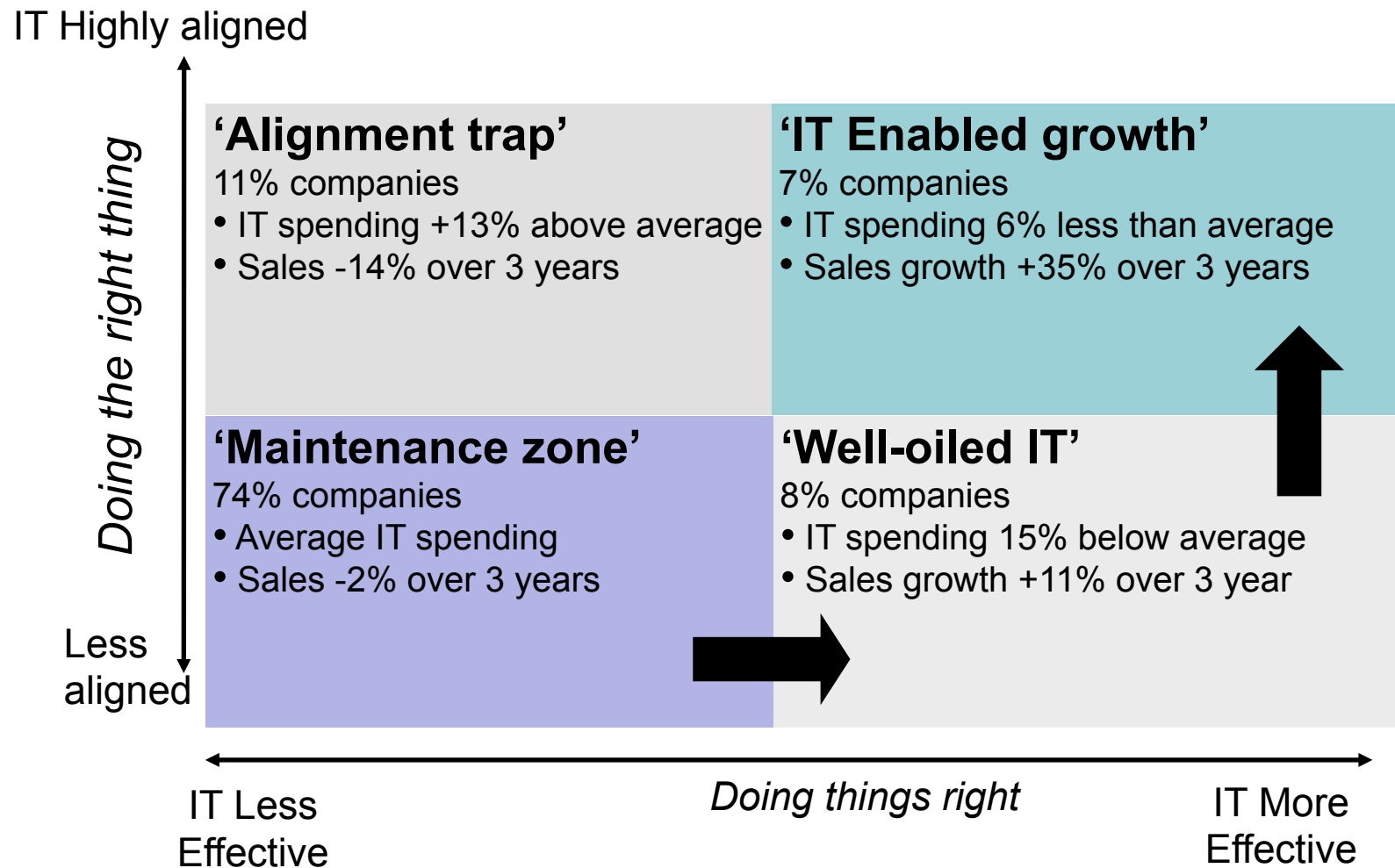


An effective delivery machine

Build effective delivery machine
Then
Move outwards and upwards



The Alignment Trap



Source: Shpilberg, Berez, Puryear, Shah: MIT Sloan Review, Fall 2007



How much quality can we afford?

- » Lots
- » Quality is free
 - If you invest in it

Thank you!

allan@allankelly.net

<http://www.allankelly.net>

Questions?

