

15 September 2014

# Architecture, Agile and DevOps

From Continuous Meetings to Continuous Delivery ☺

Peter Eeles

FSS Industry Lead, IBM Rational Worldwide Tiger Team

Email: [peter.eeles@uk.ibm.com](mailto:peter.eeles@uk.ibm.com)

Website: [www.architecting.co.uk](http://www.architecting.co.uk)

Twitter: [@petereeles](https://twitter.com/petereeles)

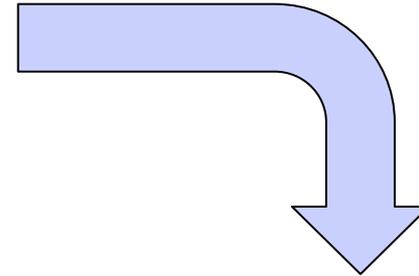


---

If you want to innovate, then you'd better think about ...

- Architecture ... How you build flexible systems
- Agile ... How you develop @ speed
- DevOps ... How you deliver @ speed
- THE END ... and thanks for listening 😊

# The changing face of FSS



31 March 2014 Last updated at 12:34



## Mobile phone banking transactions double in a year, BBA says



Mobile banking app use almost doubled over the past year, BBA figures have suggested

A "revolution" in mobile banking use is under way in the UK, the British Bankers' Association (BBA) has said.

[Related Stories](#)

3 April 2014 Last updated at 10:41



## RBS to close 44 branches across UK



Royal Bank of Scotland is closing a total of 44 branches across the UK

Taxpayer-backed Royal Bank of Scotland (RBS) has announced it is closing 44 branches across the UK, 14 of which are classed as the "last banks in town".

That is despite the fact that the bank pledged in 2010 not to close such branches.

A spokeswoman for RBS said "the world had changed since then", and there had been a 30% fall in branch transactions over the last four years.

[Related Stories](#)

[UK sees mobile banking 'revolution'](#)

[More Ulster Bank branches to close](#)

[Bank branches to close in overhaul](#)



# Mobile payments – funds

## Prepaid debit cards



## Online store cards



## Mobile wallets

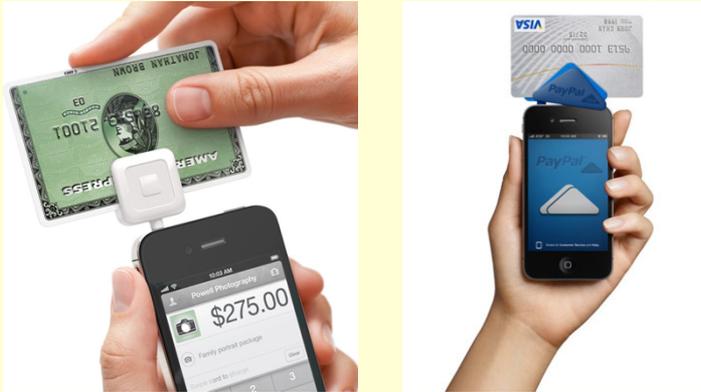


## Virtual currencies



# Mobile payments – mechanisms

## POS Devices



## Peer-to-peer



## Contactless



# Mobile money

M-PESA (M=Mobile, PESA=Money, in Swahili)



## CBA Kaching for Facebook payments

- Make payments to anyone who is in your Facebook friends list and has an Australian bank account

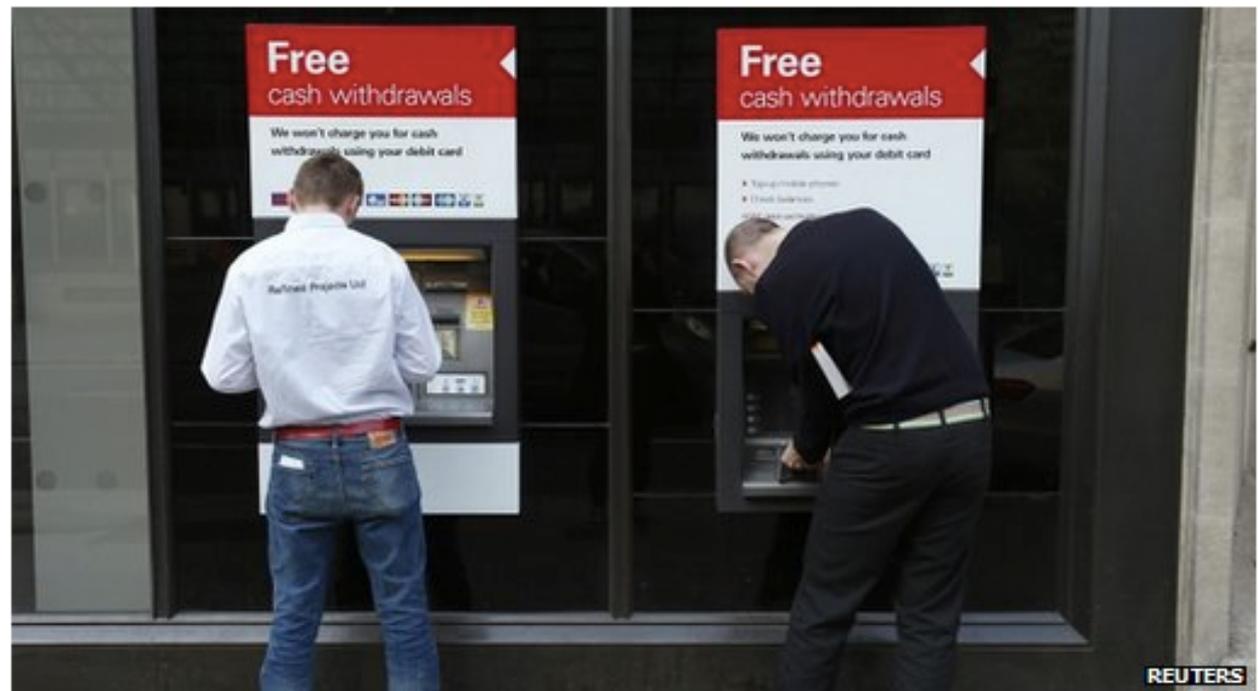
The screenshot shows the Facebook interface for the 'Commonwealth Bank Kaching' page. The top navigation bar includes the Facebook logo, a search bar, and the user's name 'Jim Marous' with 'Home' and a settings icon. Below the navigation bar, the page header features the Commonwealth Bank logo and 'Kaching' with a dropdown arrow, a 'Like' button, and a 'Create Page' button. The main content area has a large yellow 'K' logo, the text 'Pay anyone, anytime, anywhere', a green 'Get started' button, and a link to 'View CommBank Kaching for Facebook's Terms & Conditions'. Below this is a row of four feature tiles: 'Take a tour' (with a play button icon), 'Social payments' (with a Facebook 'f' icon, a person icon, and a dollar sign icon), 'Everyday banking' (with a document icon), and '100% security guarantee' (with a shield icon). At the bottom, there is a 'Take a tour' section with the text 'Watch the video to view what you can do on Kaching for Facebook.'

On April 8,  
Microsoft will  
end support for  
Windows XP.

00  
DAYS

09  
HOURS

55  
MINUTES

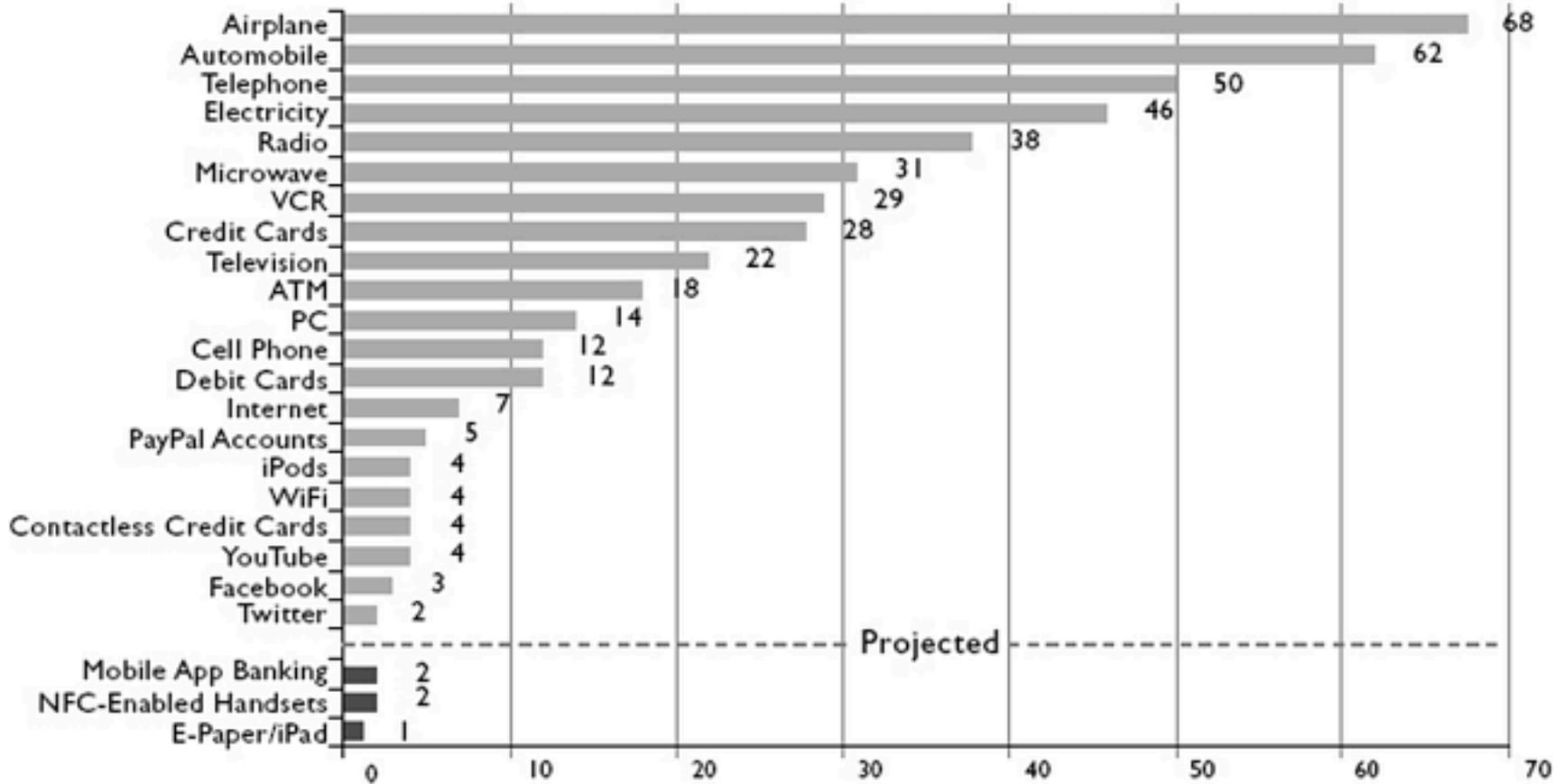


REUTERS

Many of the world's cash machines are still believed to be running Windows XP

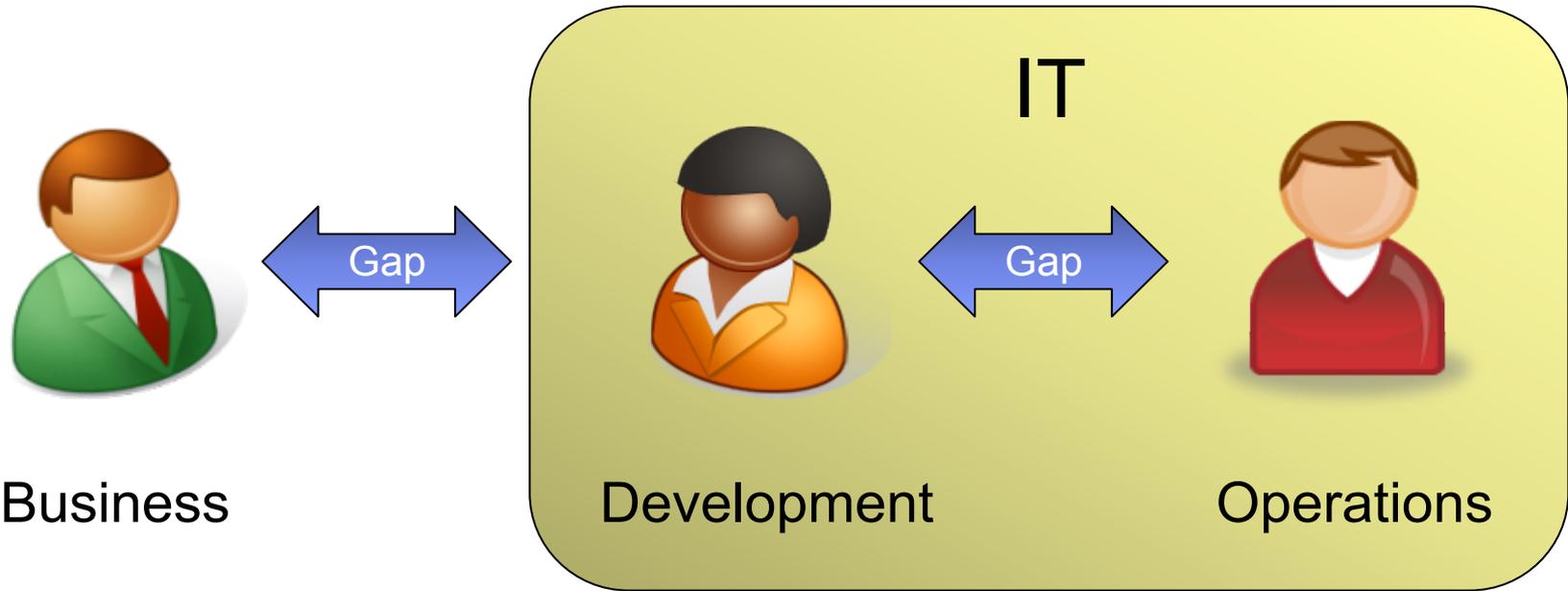
# Years till 50 million users

## Years till 50 million users



From Bank 3.0 – Brett King

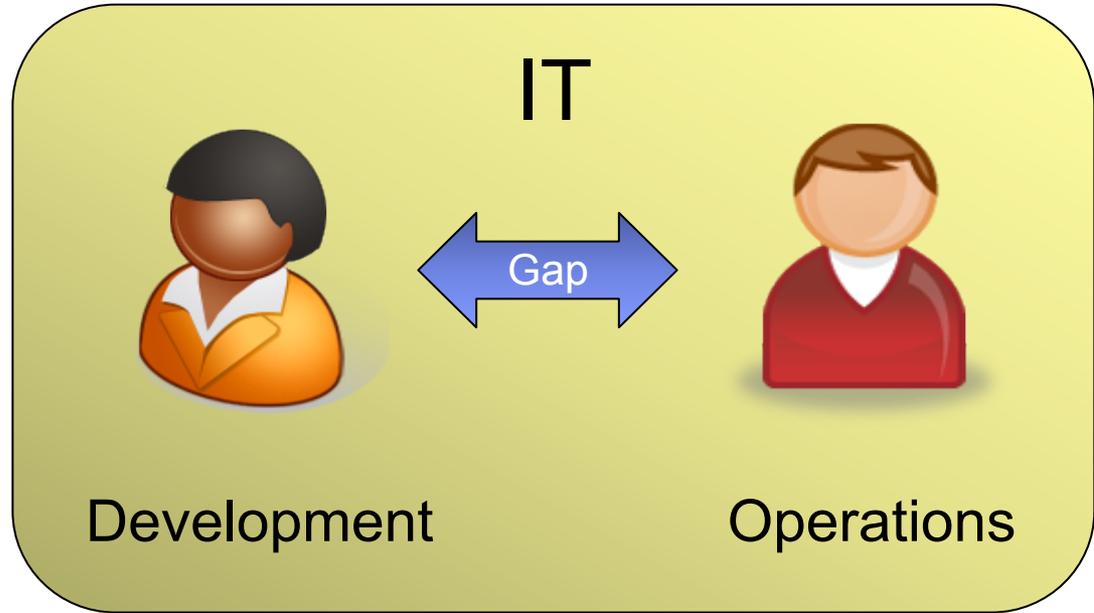
# The “Business-IT” and “IT-IT” gaps



# Viewing the landscape as a supply chain



Business



Development

Operations

Enterprise  
Architecture

Portfolio  
Management

Requirements

Design

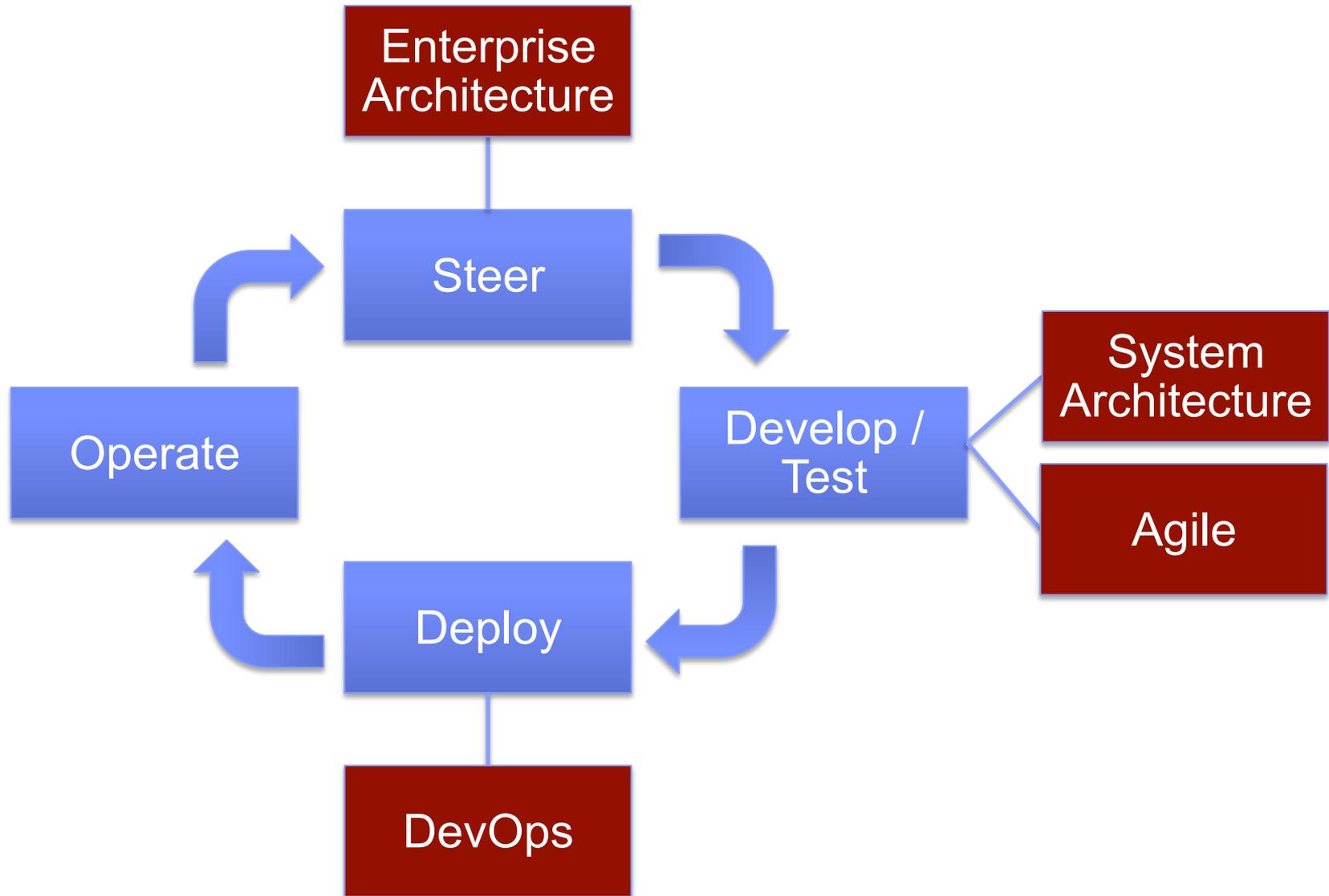
Development

Test

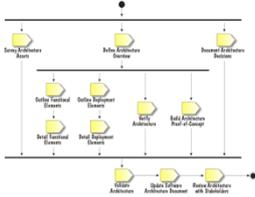
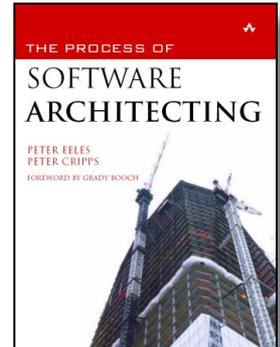
Release

Monitoring

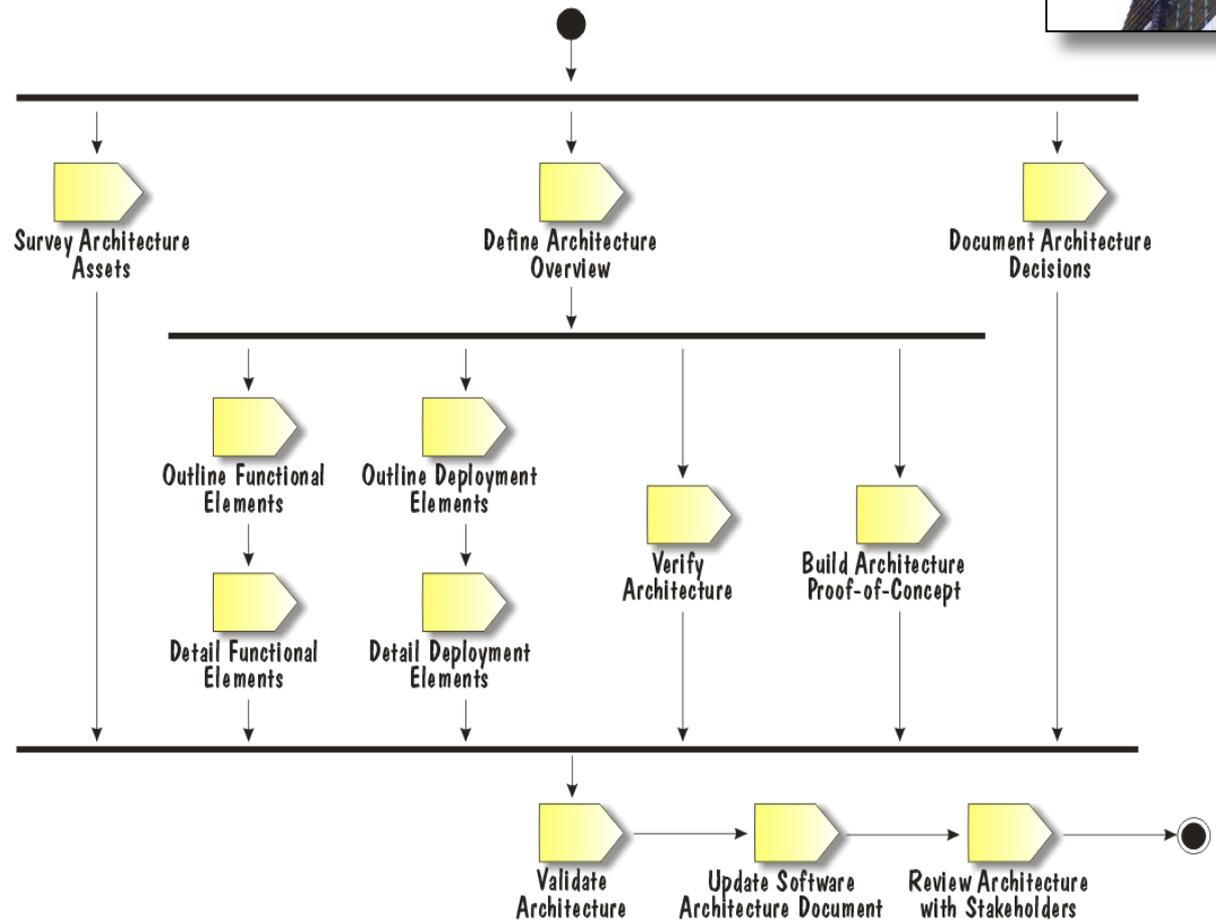
# Closing the gaps with Architecture, Agile and DevOps



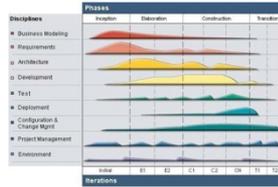
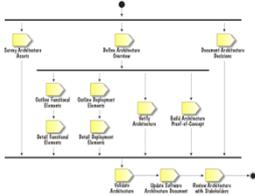
# Practice summary – traditional



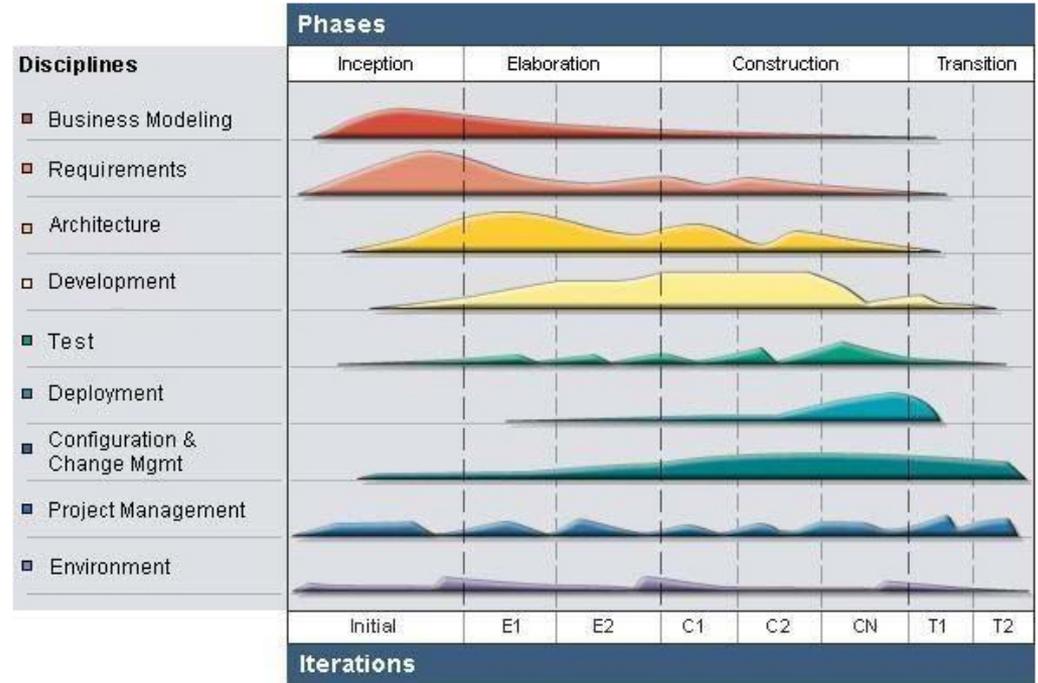
- ## Traditional
- Multiple Views
  - Quality Attribute-Driven Development
  - Component-Based Development
  - Asset Reuse
  - Decision Capture
  - Architecture Proving



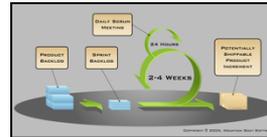
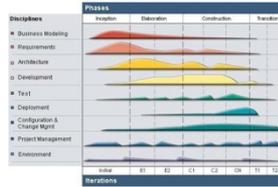
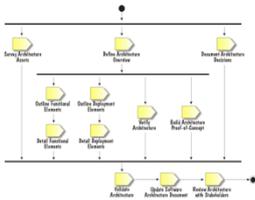
# Practice summary – iterative



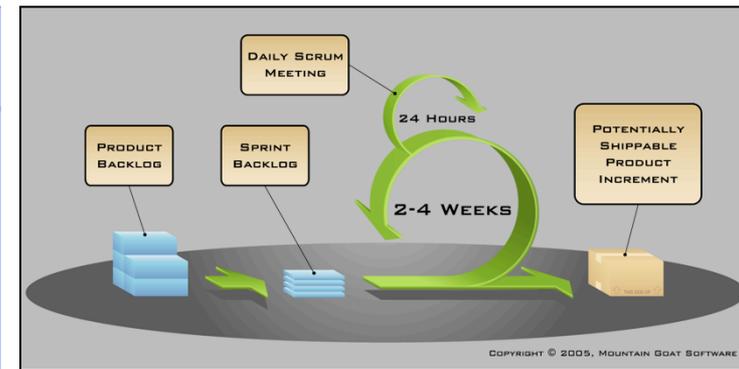
Traditional	Iterative
<ul style="list-style-type: none"> <li>• Multiple Views</li> <li>• Quality Attribute-Driven Development</li> <li>• Component-Based Development</li> <li>• Asset Reuse</li> <li>• Decision Capture</li> <li>• Architecture Proving</li> </ul>	<ul style="list-style-type: none"> <li>• Iterative Development</li> <li>• Risk-Value Lifecycle</li> <li>• Shared Vision</li> <li>• Use Case-Driven Development</li> <li>• Release Planning</li> </ul>



# Practice summary – agile



Traditional	Iterative	Agile
<ul style="list-style-type: none"> <li>Multiple Views</li> <li>Quality Attribute-Driven Development</li> <li>Component-Based Development</li> <li>Asset Reuse</li> <li>Decision Capture</li> <li>Architecture Proving</li> </ul>	<ul style="list-style-type: none"> <li>Iterative Development</li> <li>Risk-Value Lifecycle</li> <li>Shared Vision</li> <li>Use Case-Driven Development</li> <li>Release Planning</li> </ul>	<ul style="list-style-type: none"> <li>Test-Driven Development</li> <li>Continuous Integration</li> <li>Refactoring</li> <li>Whole Team</li> <li>User Story-Driven Development</li> <li>Team Change Management</li> </ul>



## Architecture and agile – a clash of cultures?

Architects' Perception of Agile	Agilists' Perception of Architecture
Leaves key decisions until last possible moment	Locks down key decisions too early
A lack of forethought leads to technical debt	BDUF over YAGNI ☺
A lack of modelling leads to significant rework when software is poorly thought through and does not scale	Comprehensive models slow your development efforts down to a snail's pace

*“I see two parties not really understanding the real issues at hand, stopping at a very shallow, caricatural view of the “other culture”, not understanding enough of the surroundings, beliefs, values of the other one, and stopping very quickly at judging behaviors.” (Kruchten, 2010)*

*“Scrum is a management and control process that cuts through complexity to focus on building software to meet business needs. Scrum is superimposed on top of and wraps existing engineering practices, development methodologies and standards”. (Schwaber & Beedle, 2002)*

*“Look at a large successful software system and beneath it you’ll find an architecture that’s kept its evolution on track”. (Spinellis, 2010)*

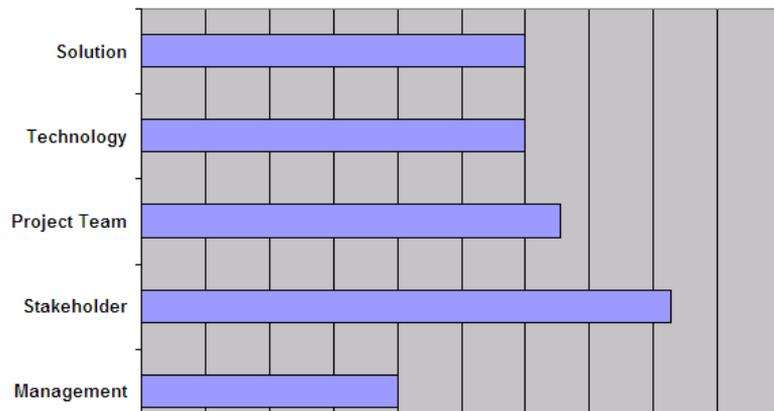
## Lifecycle selection framework (summary)

- **Management Influences**
  - Business Flexibility
  - Empowered Teams
- **Stakeholder Influences**
  - Acceptance of Agile
  - Number of Stakeholders
  - Stakeholder Responsiveness
- **Project Team Influences**
  - Team Skills
  - Embracing Change
  - Co-located Teams
  - Team Stability
  - Team Roles
  - Agile Disciplines
- **Technology Influences**
  - Development Environment
  - Execution Environment
- **Solution Influences**
  - Requirements Churn
  - Solution Complexity
  - Time-To-Market
  - Dependencies
  - Release Frequency
  - Demonstrability

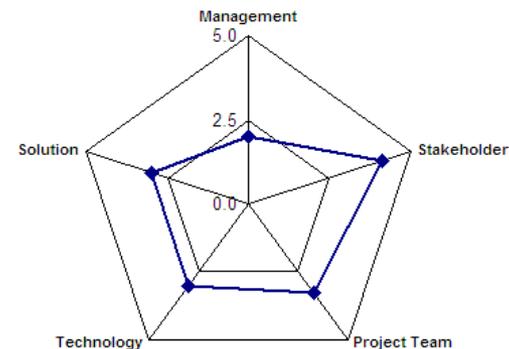
Dimension	Brief Description	Score
Management	Management Influences	2.0
Stakeholder	Stakeholder Influences	4.1
Project Team	Project Team Influences	3.3
Technology	Technology Influences	3.0
Solution	Solution Influences	3.0

15.4 pts out of a total of 25 pts in favour of an agile (rather than a traditional) approach  
61.6 %

Results



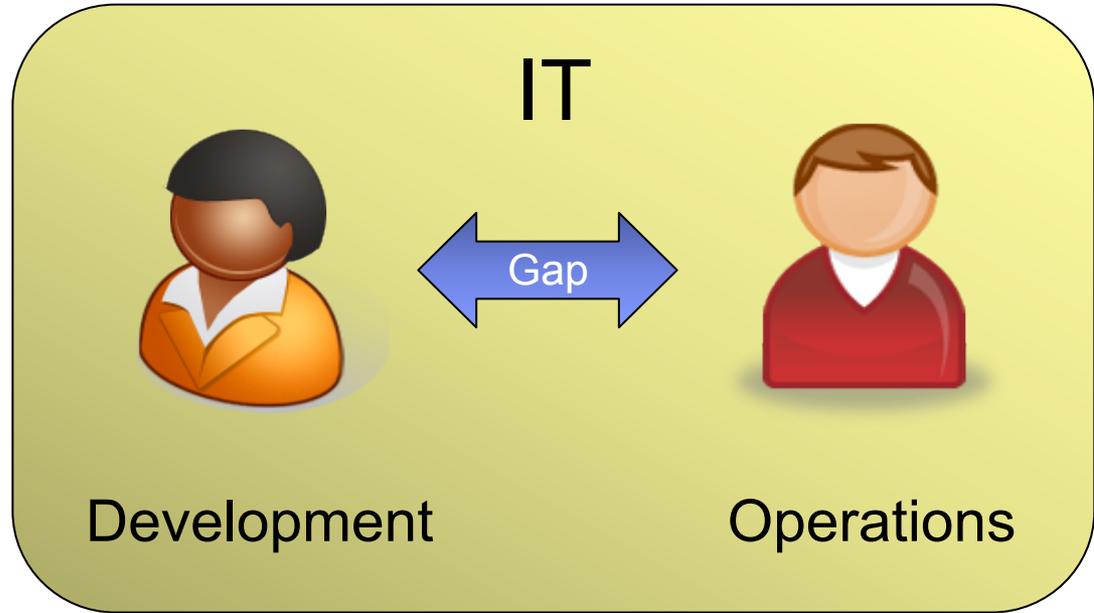
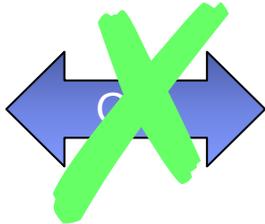
Radar Chart



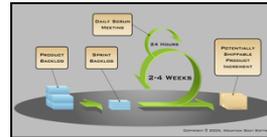
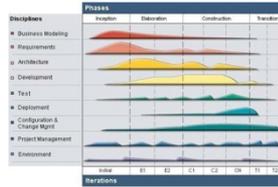
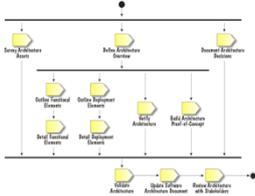
# The impact of agile



Business

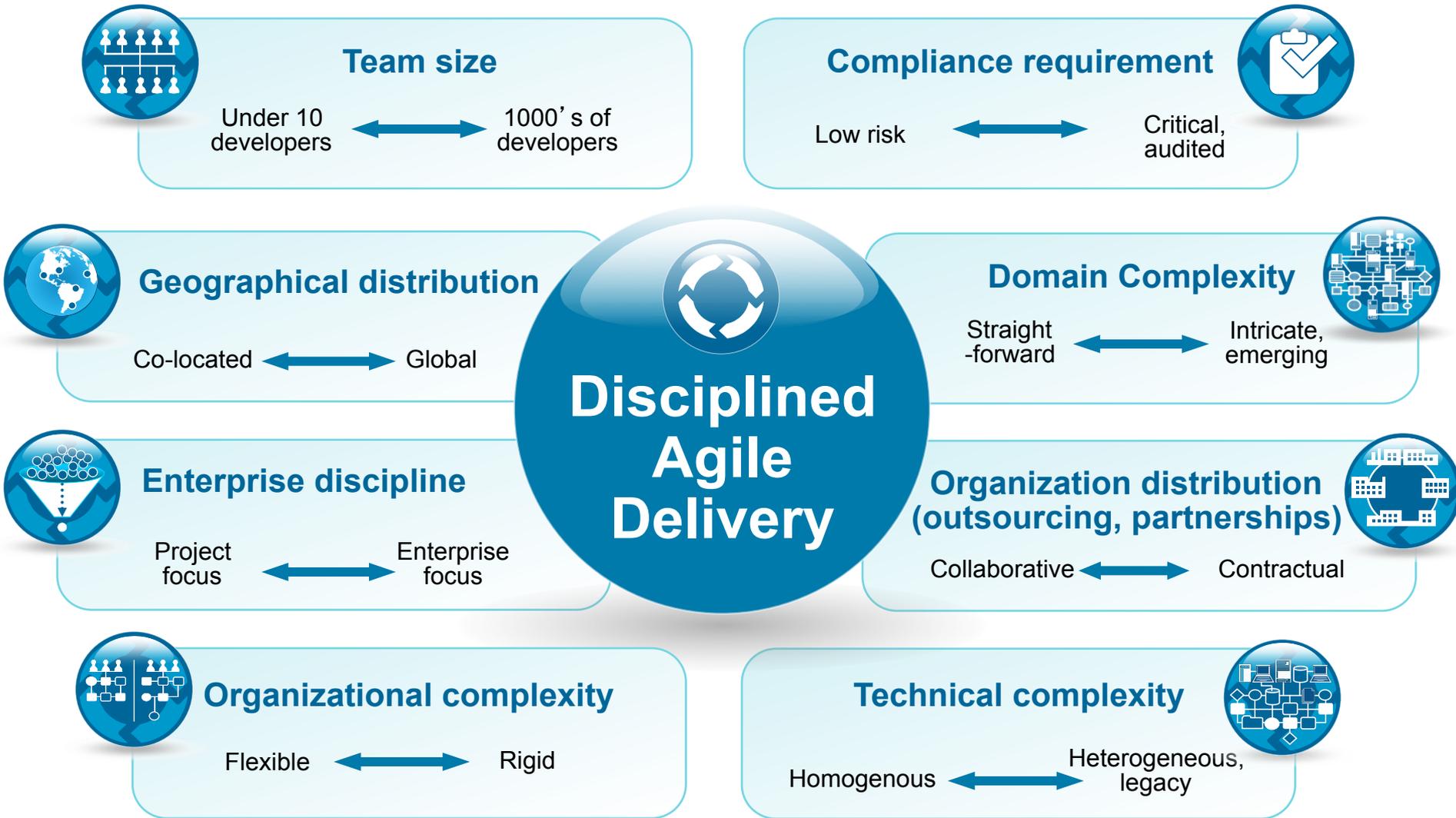


# Practice summary – disciplined / scaled agile

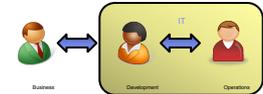
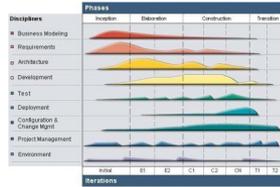
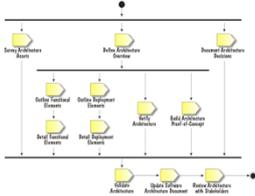


Traditional	Iterative	Agile	Disciplined Agile
<ul style="list-style-type: none"> <li>Multiple Views</li> <li>Quality Attribute-Driven Development</li> <li>Component-Based Development</li> <li>Asset Reuse</li> <li>Decision Capture</li> <li>Architecture Proving</li> </ul>	<ul style="list-style-type: none"> <li>Iterative Development</li> <li>Risk-Value Lifecycle</li> <li>Shared Vision</li> <li>Use Case-Driven Development</li> <li>Release Planning</li> </ul>	<ul style="list-style-type: none"> <li>Test-Driven Development</li> <li>Continuous Integration</li> <li>Refactoring</li> <li>Whole Team</li> <li>User Story-Driven Development</li> <li>Team Change Management</li> </ul>	<ul style="list-style-type: none"> <li>Measured Performance</li> <li>Formal Change Management</li> <li>Concurrent Testing</li> </ul>

# Agile with discipline



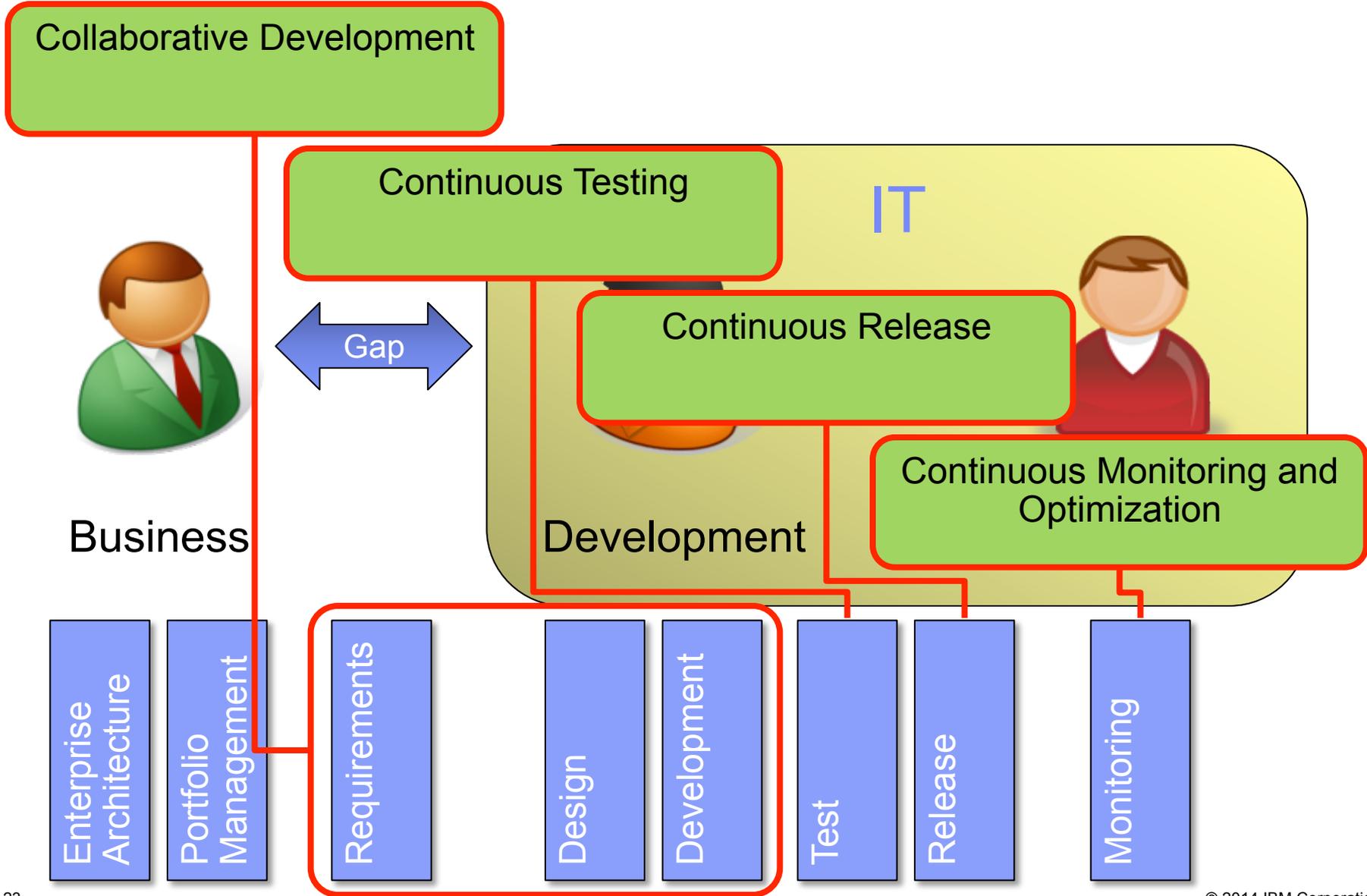
# Practice summary – DevOps



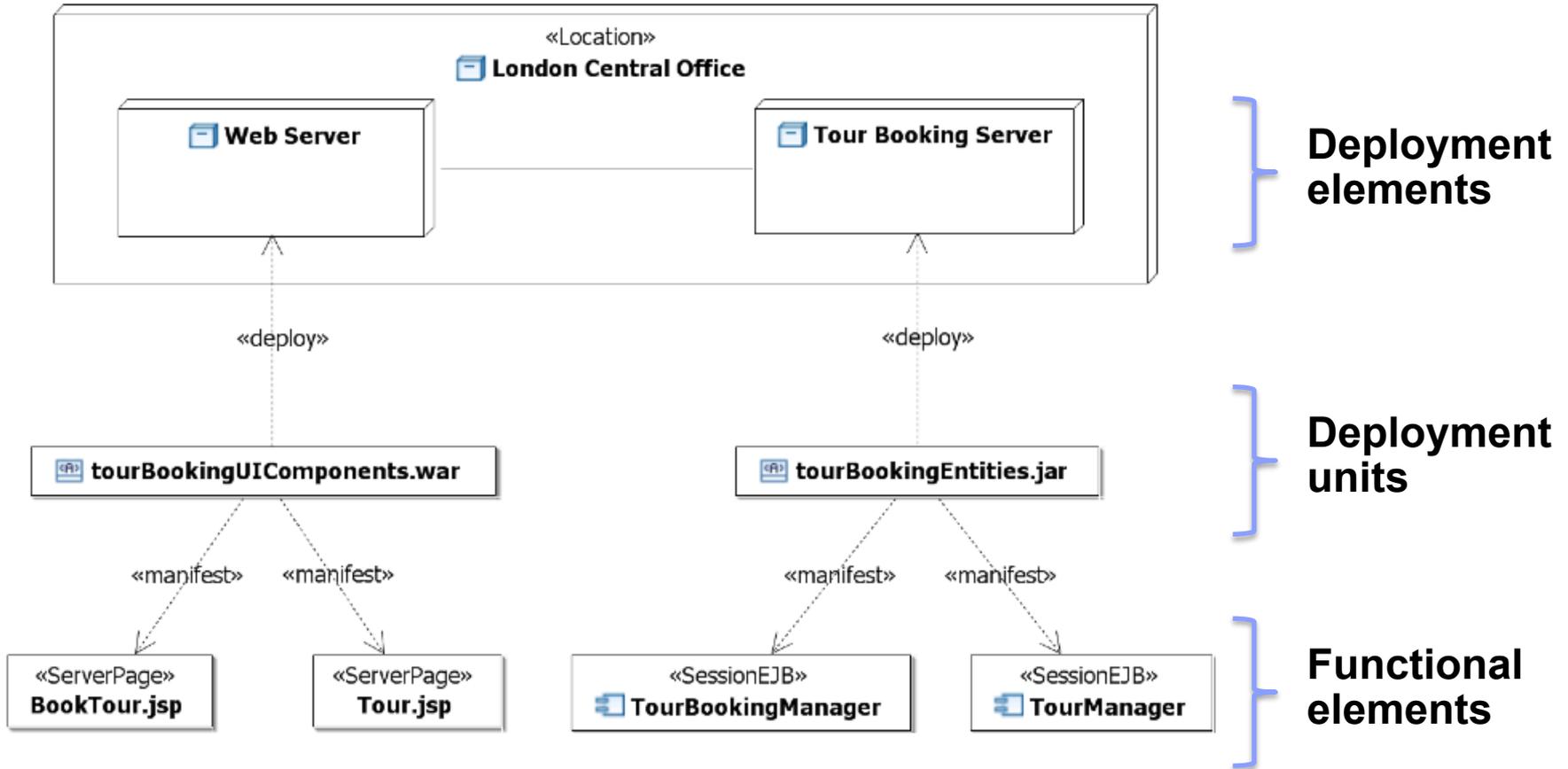
Traditional	Iterative	Agile	Disciplined Agile	DevOps
<ul style="list-style-type: none"> <li>Multiple Views</li> <li>Quality Attribute-Driven Development</li> <li>Component-Based Development</li> <li>Asset Reuse</li> <li>Decision Capture</li> <li>Architecture Proving</li> </ul>	<ul style="list-style-type: none"> <li>Iterative Development</li> <li>Risk-Value Lifecycle</li> <li>Shared Vision</li> <li>Use Case-Driven Development</li> <li>Release Planning</li> </ul>	<ul style="list-style-type: none"> <li>Test-Driven Development</li> <li>Continuous Integration</li> <li>Refactoring</li> <li>Whole Team</li> <li>User Story-Driven Development</li> <li>Team Change Management</li> </ul>	<ul style="list-style-type: none"> <li>Measured Performance</li> <li>Formal Change Management</li> <li>Concurrent Testing</li> </ul>	<ul style="list-style-type: none"> <li>Collaborative Development</li> <li>Continuous Testing</li> <li>Continuous Release</li> <li>Continuous Monitoring and Optimisation</li> </ul>



# DevOps practices



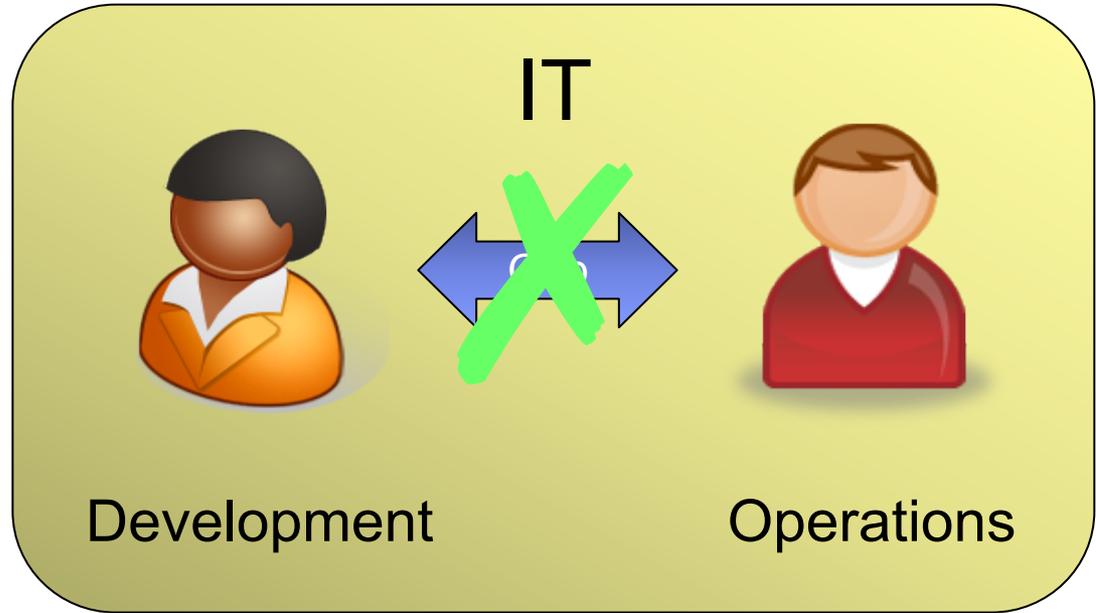
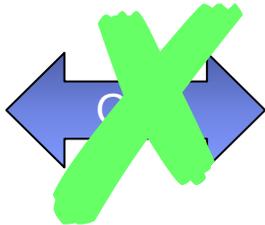
# Architecture and DevOps



# The impact of DevOps



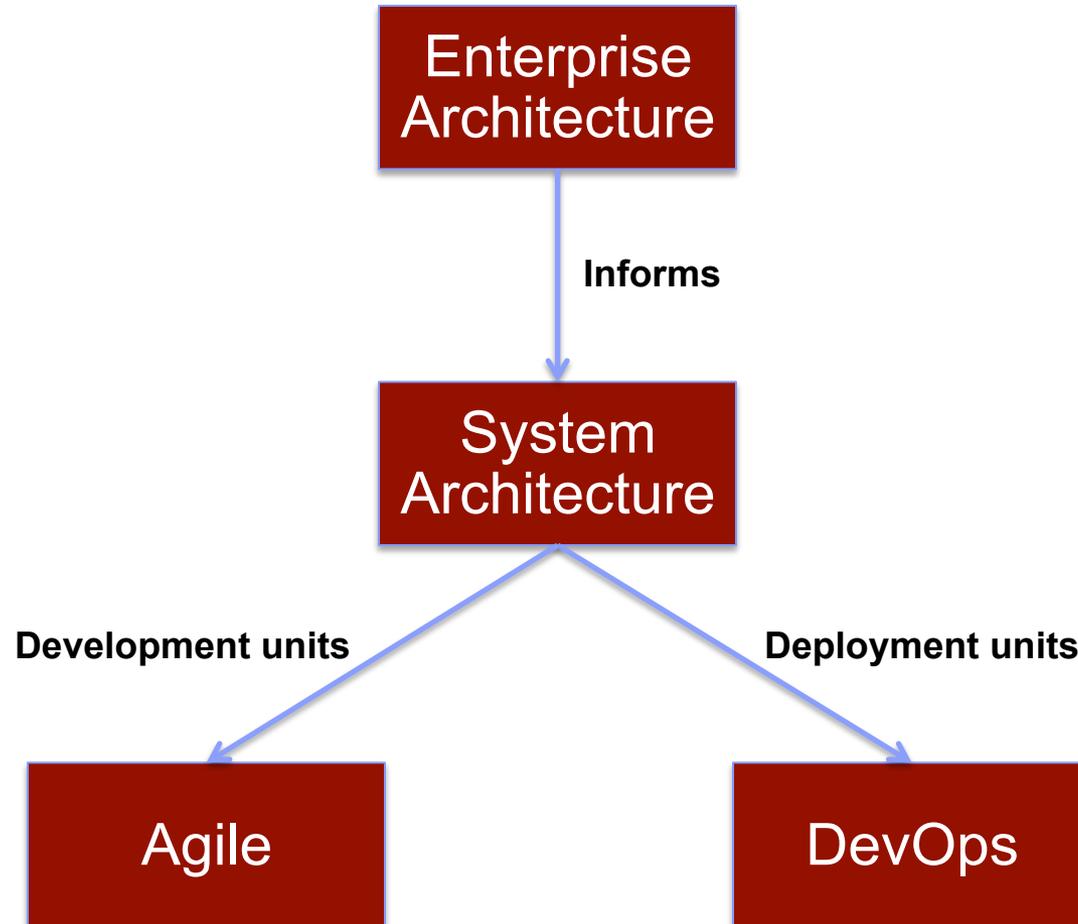
Business



Development

Operations

## Summary



Thank  
YOU