Please note

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
If you want to <Innovate@SPEED>

... then you’d better think about:

1. Architecture … How you build flexible systems
2. Agile … How you develop @ speed
3. DevOps … How you deliver @ speed

THE END … and thanks for listening 😊
Agenda

1. Introduction
2. Enterprise Architecture
3. System Architecture
4. Agile
5. DevOps
6. Summary
Architecture, Agile and DevOps are not independent
Agenda

1. Introduction
2. Enterprise Architecture
3. System Architecture
4. Agile
5. DevOps
6. Summary
What is Enterprise Architecture?

IBM:
“The Enterprise Architecture discipline defines and maintains the architecture models, governance, and transition initiatives needed to effectively co-ordinate semi-autonomous groups towards common business and/or IT goals.” ¹

Gartner:
“Enterprise architecture is the process of translating business vision and strategy into effective enterprise change by creating, communicating and improving the key principles and models that describe the enterprise's future state and enable its evolution.” ²

Similar definitions, consistent intent

¹ Enterprise Architecture in the era of On-Demand, IBM Academy of Technology Study, October 2004
² Short form, Gartner Defines the term ‘Enterprise Architecture’, Anne Lapkin, Gartner, July 12, 2006
There’s more to an EA than “just an architecture”

Effective business transformation and optimization requires deep understanding of the enterprise architecture as a control point for governance, integration and project planning.

- **Transition Planning**
  - Are we still moving in the right direction?

- **Governance**
  - Are our target architectures still right?
  - Are we doing these things the way we said we want them done?

- **Project Prioritization & Planning**
  - These are the things we should do

- **Solution Development & Delivery**
  - This is the way things should be architected

- **Enterprise Architecture Models**
  - Building blocks, rules, patterns, constraints

- **Strategy, purpose, objective, vision, etc.**
EA motivation and progression

Cost Reduction
- What do we have?
- Need all of it?
- Consolidate to reduce costs?
- Desire for impact analysis

Standardization
- Develop standards and recommended best practices (e.g. technology stacks, server platforms)
- Seeking repeatability
- Encourage IT evolution
- Focusing on IT scope only

Broaden Scope
- Meet business needs by linking IT to business
- Managing architectures outside IT
- Increasing focus on business architecture and business processes

Actionable EA
- Develop business strategy
- Value propositions, capabilities, resources?
- Refine into to-be
- Compare to as-is
- Create transition plan
- Execute

Opportunistic
Systematic

There is a range of EA goals & scope
### Enterprise architecture and system architecture

<table>
<thead>
<tr>
<th>Business strategy</th>
<th>Governance</th>
<th>Organization</th>
<th>Key business processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical data models</td>
<td>Physical data models</td>
<td>Data management resources</td>
<td>Application blueprints</td>
</tr>
<tr>
<td>Hardware</td>
<td>Software</td>
<td>Network infrastructure</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Business Architecture</th>
<th>Data Architecture</th>
<th>Application Architecture</th>
<th>Technology Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informs</td>
<td>Informs</td>
<td>Informs</td>
<td>Informs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System Requirements</th>
<th>Data Architecture</th>
<th>Application Architecture</th>
<th>Technology Architecture</th>
</tr>
</thead>
</table>
Can you spot the innovation enablers?

Survey Architecture Assets → Define Architecture Overview → Document Architecture Decisions


Detail Functional Elements → Detail Deployment Elements

Validate Architecture → Update Software Architecture Document → Review Architecture with Stakeholders

#ibminnovate
A requirement

Tour Booker

Customer

Sales Clerk

Book Tour

Payment Engine

Reservation System

#ibminnovate
A logical application model (structure)
A logical application model (behavior)
A logical application model

Application-Specific

- Boundary - Tour Booker Boundary
- Control - Book Tour Controller
- Entity - Tour Manager
- Entity - Tour Booking Manager
- Entity - Tour Rules

Business-Specific

- Boundary - Payment Engine Boundary
- Boundary - CRM System Boundary
- Boundary - Reservation System Boundary

Business-Independent

- Entity - Transaction Log
- Entity - Message Bus
- ACME messaging bus
A logical infrastructure model

- Remote Office
- Branch Office
- Central Office
- MyPay Data Center
- MyReservation Data Center
A logical infrastructure model

«Location»
- Central Office
- Content Management Server
- Security Server
- CRM Server
- Tour Booking Server
- Integration Server
A logical model
A physical infrastructure model

- **Admin Client**
  - CPU: 1x1.2GHz
  - Model: Dimension
  - Memory: 2Gb
  - Mfr: Dell
  - Disk: 160Mb

- **Web Server**
  - CPU: 1x1.2GHz
  - Model: p310
  - Memory: 1Gb
  - Mfr: IBM

- **Tour Booking Server**
  - CPU: 2x1.2GHz
  - Model: p320
  - Memory: 4Gb
  - Mfr: IBM

- **Database Server**
  - CPU: 2x1.2GHz
  - Model: p320
  - Memory: 2Gb
  - Mfr: IBM

- **Content Management Server**
  - CPU: 1x1.2GHz
  - Model: p310
  - Memory: 1Gb
  - Mfr: IBM

- **Security Server**
  - CPU: 1x1.2GHz
  - Model: p310
  - Memory: 1Gb
  - Mfr: IBM

- **Integration Server**
  - CPU: 2x1.2GHz
  - Model: p320
  - Memory: 2Gb
  - Mfr: IBM

**Location**
- **London Central Office**

**Disk Array**
- Mfr: Sun
- Model: Fire X4-140
- Disk: 1Tb
A physical model

Deployment elements

Deployment units

Functional elements
Key dependencies

Enterprise Architecture

Informs

System Architecture

Development units

Agile

Deployment units

DevOps
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.

“Software architecture has a history of excesses that in part spurred the reaction called Agile. Software architecture of the 1980s was famous for producing reams of documentation that no one read”. (Coplien & Bjørnvig, 2010)
## Architecture and agile – a clash of cultures?

<table>
<thead>
<tr>
<th>Architects’ Perception of Agile</th>
<th>Agilists’ Perception of Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaves key decisions until last possible moment</td>
<td>Locks down key decisions too early</td>
</tr>
<tr>
<td>A lack of forethought leads to technical debt</td>
<td>BDUF over YAGNI 😊</td>
</tr>
<tr>
<td>A lack of modelling leads to significant rework when software is poorly thought through and does not scale</td>
<td>Comprehensive models slow your development efforts down to a snail’s pace</td>
</tr>
</tbody>
</table>

“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)
Iterative development practices

- Iterative development
  - Incremental releases provide improvements in capability until the final system is complete

- Risk-value lifecycle
  - Phases represents the strategic plan for the project and drives the goals and objectives of each iteration

- Shared Vision
  - Ensures that all stakeholders share a common view of the problems being solved

- Use Case-Driven Development
  - Use cases make natural units of implementation in an iterative development approach

- Release Planning
  - Focused on the just-in-time project planning needed to scope the release of executable software within an iteration
Agile practices

- **Test-Driven Development (TDD)**
  - Creating tests that are a specification of what the code should do first
- **Continuous Integration**
  - Encourages frequent the integration and testing of programming changes
- **Refactoring**
  - Changing an existing body of code in order to improve its internal structure
- **Whole Team**
  - A focus on the value of highly-collaborative teams as exemplified by Scrum’s daily standup meeting. Instills of sense of collective ownership and responsibility
- **User Story-Driven Development**
  - Capture requirements in a lightweight manner. Encourages collaboration with the relevant stakeholders throughout a project
- **Team Change Management**
  - Supports the logging of defects or new requirements, by any member of the team, that are within the scope of the current iteration
“Disciplined Agile Delivery” practices (examples)

- **Measured Performance**
  - This practice allows project and portfolio-level measurements to inform key business decisions

- **Formal Change Management**
  - Applied when change approval is required from stakeholders outside of the project team, or when a deliverable has been baselined as part of a contract and the deliverable needs to be modified

- **Concurrent Testing**
  - Often an independent test team is present (especially in larger organizations) that typically provide a level of user acceptance testing before the solution is put into production
## Practice summary

<table>
<thead>
<tr>
<th>Traditional</th>
<th>Iterative</th>
<th>Agile</th>
<th>Disciplined Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Multiple Views</td>
<td>- Iterative Development</td>
<td>- Test-Driven Development</td>
<td>- Measured Performance</td>
</tr>
<tr>
<td>- Quality Attribute-Driven Development</td>
<td>- Risk-Value Lifecycle</td>
<td>- Continuous Integration</td>
<td>- Formal Change Management</td>
</tr>
<tr>
<td>- Component-Based Development</td>
<td>- Shared Vision</td>
<td>- Refactoring</td>
<td>- Concurrent Testing</td>
</tr>
<tr>
<td>- Asset Reuse</td>
<td>- Use Case-Driven Development</td>
<td>- Whole Team</td>
<td></td>
</tr>
<tr>
<td>- Decision Capture</td>
<td>- Release Planning</td>
<td>- User Story-Driven Development</td>
<td></td>
</tr>
<tr>
<td>- Architecture Proving</td>
<td></td>
<td>- Team Change Management</td>
<td></td>
</tr>
</tbody>
</table>
# Picking your fights – when to “go agile”

<table>
<thead>
<tr>
<th>Management Influences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business Flexibility</strong></td>
</tr>
<tr>
<td>Management are willing to accept that business parameters, such as cost, schedule and intermediate milestones, are flexible</td>
</tr>
<tr>
<td><strong>Empowered Teams</strong></td>
</tr>
<tr>
<td>Management is willing to allow the team (including the product owner) to make key project decisions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stakeholder Influences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Acceptance of Agile</strong></td>
</tr>
<tr>
<td>Stakeholders understand and accept agile practices and the consequences of following these</td>
</tr>
<tr>
<td><strong>Number of Stakeholders</strong></td>
</tr>
<tr>
<td>The number and diversity of stakeholder relationships to be managed is limited</td>
</tr>
<tr>
<td><strong>Stakeholder Responsiveness</strong></td>
</tr>
<tr>
<td>The business representative, end users and testers are committed to spending a good deal of time working with the team in an iterative fashion</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project Team Influences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Team skills</strong></td>
</tr>
<tr>
<td>Individuals on the team are team players, good communicators and are familiar with agile practices</td>
</tr>
<tr>
<td><strong>Embracing Change</strong></td>
</tr>
<tr>
<td>Team members expect and embrace frequent changes and iterative refinement of the solution</td>
</tr>
<tr>
<td><strong>Colocated Teams</strong></td>
</tr>
<tr>
<td>The project team will be co-located</td>
</tr>
<tr>
<td><strong>Team Stability</strong></td>
</tr>
<tr>
<td>Individuals will be assigned to the team for the duration of the project</td>
</tr>
<tr>
<td><strong>Team Roles</strong></td>
</tr>
<tr>
<td>Team members are able (and willing) to take on multiple roles during the project and to take on new roles if/when needed</td>
</tr>
<tr>
<td><strong>Agile Disciplines</strong></td>
</tr>
<tr>
<td>Team members have proven ability in performing disciplines that are critical for agile development with short iterations (design, testing and configuration management)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technology Influences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Development Environment</strong></td>
</tr>
<tr>
<td>The development environment (method, tools, training) will support an agile way of working (such as automated regression test, continuous integration and real-time dashboards) and is sufficiently mature</td>
</tr>
<tr>
<td><strong>Execution Environment</strong></td>
</tr>
<tr>
<td>The execution environment can support regular releases</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solution influences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements Churn</strong></td>
</tr>
<tr>
<td>There is a strong likelihood that there will be significant changes to requirements (and the solution) during the project</td>
</tr>
<tr>
<td><strong>Solution Complexity</strong></td>
</tr>
<tr>
<td>The required solution is relatively-complex (e.g. requires the use of unfamiliar technologies) and/or there are many different solution options</td>
</tr>
<tr>
<td><strong>Time-to-market</strong></td>
</tr>
<tr>
<td>The deadline (time) is the most important factor for the solution while the scope of the solution is flexible.</td>
</tr>
<tr>
<td><strong>Dependencies</strong></td>
</tr>
<tr>
<td>There are no (or only a few) dependencies on internal or external suppliers</td>
</tr>
<tr>
<td><strong>Release Frequency</strong></td>
</tr>
<tr>
<td>The solution can be subdivided in viable and meaningful business releases that can each be delivered within 3-4 months</td>
</tr>
<tr>
<td><strong>Demonstrability</strong></td>
</tr>
<tr>
<td>The solution can be easily demonstrated on an incremental basis (through a user interface, for example)</td>
</tr>
</tbody>
</table>
Agenda

1. Introduction
2. Enterprise Architecture
3. System Architecture
4. Agile
5. DevOps
6. Summary
Capitalizing on technology trends

Big Data
Insights on client behaviors drive personalization and identify moments of truth

Cloud
On-demand businesses require fast, scalable environments for dev, test, and production

Social Business
P2P and Social transactions are opening the door to a new set of competitors

Instrumented Products
The internet of things (IOT) is rapidly becoming a platform of commerce

Mobile
Anytime, anyplace omni-channel experiences emerging

Intelligent/ Connected Systems
Componentization creating all new value-chains and differentiation

Software delivery

A major Financial Services firm CIO in Europe stated they needed disruptive innovation: “If you can’t get applications out, you cannot reach the customer.”
The “Business-IT” and “IT-IT” gaps
Viewing the landscape as a supply chain

Business

Enterprise Architecture
Portfolio Management
Requirements

Development

Design
Development
Test
Release
Monitoring

Operations

MIND THE GAP

IT

Gap

Gap
DevOps practices

Collaborative Development
- Change management
- Software configuration management
- Continuous integration

Continuous Testing
- Quality management
- Test automation
- Service virtualization

Continuous Release
- Release management
- Environment provisioning
- Deployment automation

Continuous Monitoring and Optimization
- Application performance monitoring

IT

Development

Requirements
Design
Development
Test
Release
Monitoring

Gap

Business

Enterprise Architecture
Portfolio Management

Innovate 2014

#ibminnovate
Agenda

1. Introduction
2. Enterprise Architecture
3. System Architecture
4. Agile
5. DevOps
6. Summary
Summary

- Architecture, agile and DevOps are related and complementary
If you want to Innovate @ SPEED

... then you’d better think about:

1. Architecture ... How you build flexible systems
2. Agile ... How you develop @ speed
3. DevOps ... How you deliver @ speed

THE END ... 😊
Acknowledgements and Disclaimers

**Availability.** References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

© Copyright IBM Corporation 2012. All rights reserved.

- U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
- Please update paragraph below for the particular product or family brand trademarks you mention such as WebSphere, DB2, Maximo, Clearcase, Lotus, etc

IBM, the IBM logo, ibm.com, [IBM Brand, if trademarked], and [IBM Product, if trademarked] are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

If you have mentioned trademarks that are not from IBM, please update and add the following lines:

[Insert any special 3rd party trademark names/attribution here]

Other company, product, or service names may be trademarks or service marks of others.
Thank You!

Your Feedback is Important!

Access the Innovate agenda tool to complete your session surveys from your smartphone, laptop or conference kiosk.