

YOW

Leveraging Internet Technologies to Build a New Breed of Software Development Tools

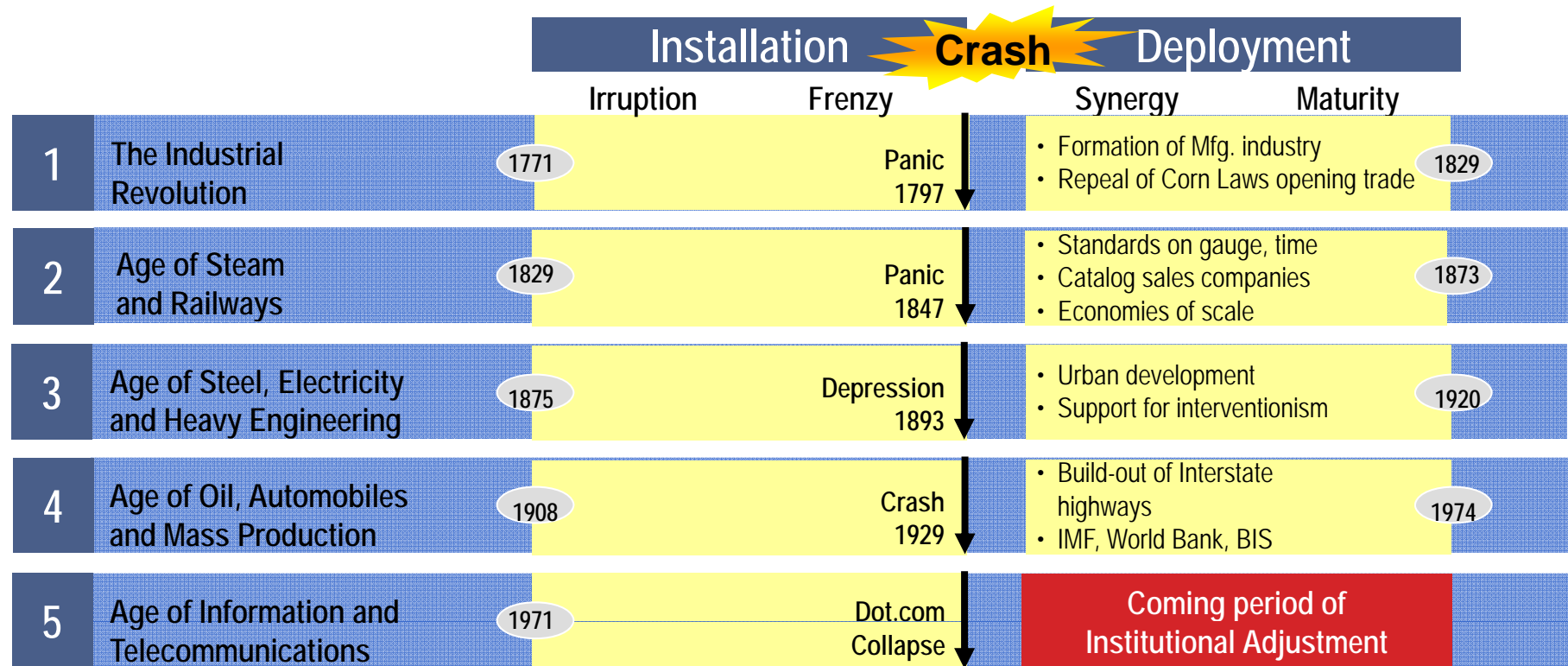
Martin Nally, VP & IBM Fellow
CTO, IBM Rational



Let's **build** a smarter planet

IBM

Five Historical Waves of Economic & Social Transformation



Source: Perez, C., *Technological Revolutions and Financial Capital*, 2002

Let's build a smarter planet



IBM

Millions of units shipped

2,000,000

1,000,000

500,000

10,000

5,000

1,000

1960

S/360

1965

1970

1975

IBM PC

1980

1985

World Wide Web

1990

1995

2000

2005

2010

2015



“Internet of Things”

Source: IDC, SSR and IBM Market Insights

Let's build a smarter planet



IBM - Smarter Planet



Instrumented

Companies, Institutions, Industries



Interconnected

Man-Made Systems



Intelligent

Nature's Systems

Economist Nov 6-12, 2010
– “It’s a smart world”

CISCO
– “Smarter-connected
Communities”

HP
– “Central Nervous System
for the Earth”

Let's **build** a smarter planet



Perspectives

- **Technology**

- Convergence of IT and Systems

- **Vision and Journalism**

- Convergence of physical and virtual
 - E.g. Shopping, travel

- **Business and Government**

- Huge potential fiscal and societal impact
 - Energy & climate change, traffic, environment, healthcare, ...

Best-in-class product & service companies are those that build a strong competency in software

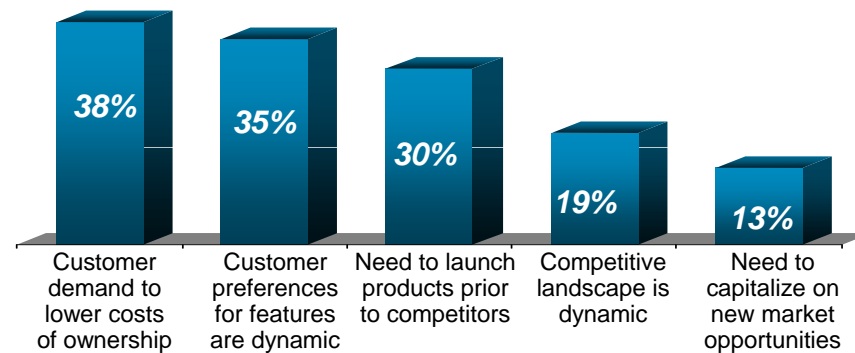
Best-of-class produce results:

- **19%** more likely to meet revenue targets than the industry average
- **4.4x** more embedded software than competitors
- **50%** fewer defects in embedded software
- **25%** decrease in product development time

“Software has evolved to become the keystone of product differentiation and end-user experience.”



Top 5 pressures driving improvements in embedded product / IT software



Source: "Embedded Systems Development", Aberdeen Group, March 2009

Let's build a smarter planet



Smarter Planet Challenges

Much discussed in the press

- Privacy
- Security
- Reliability

The rest of this talk

- Selecting investments that will bring returns
- Building them

Building a Smarter Planet is a high-risk Endeavour

Innovative, first-of-a-kind

Technology

Business models

Building systems of systems is terrifically hard

Behavior emergent, rather than designed

Testing before deploying is hard-to-impossible

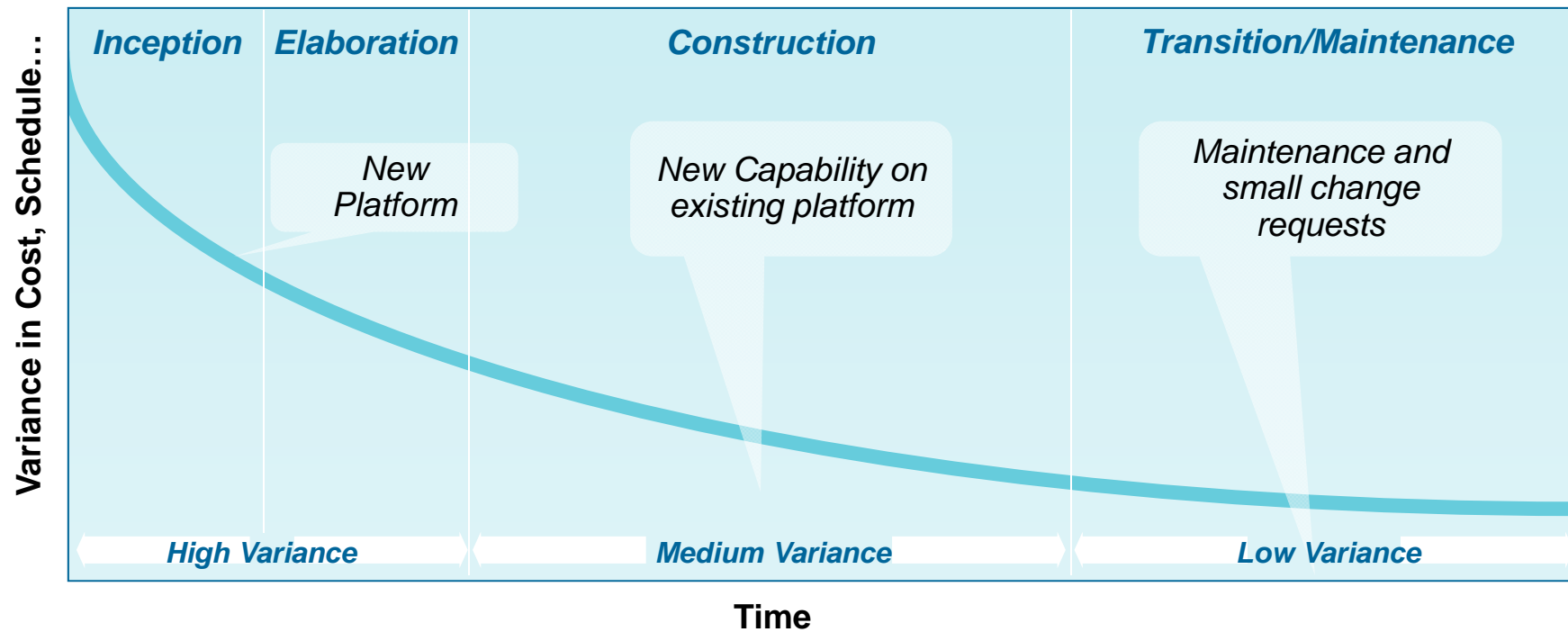


Let's **build** a smarter planet

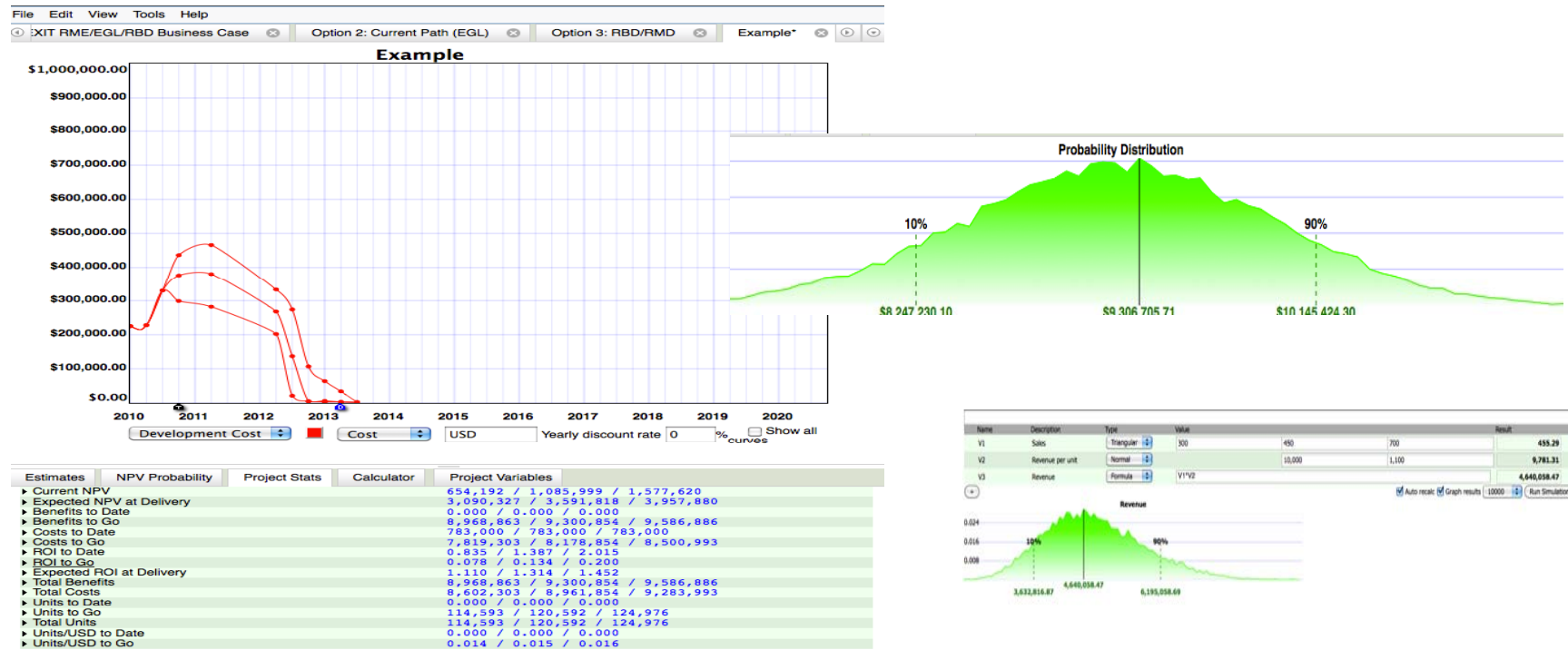
IBM

Different projects need different governance

Risk/uncertainty are the key discriminators



Financier – using probability distributions to evaluate project value



Let's build a smarter planet



Ways to improve software development outcomes

Improve skills

New software development technologies

Don't develop software at all

- Outsource development
- Use packaged applications

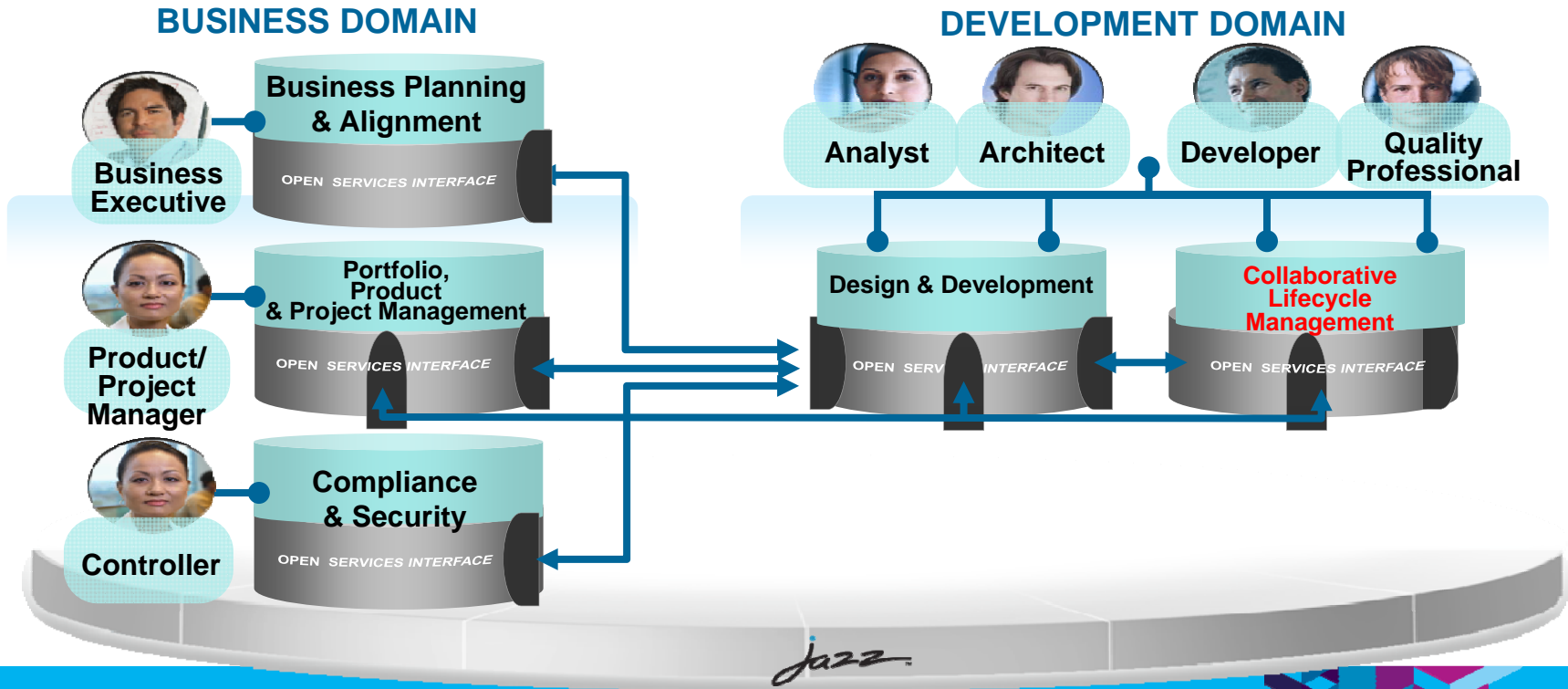
Improve processes



Let's **build** a smarter planet

IBM

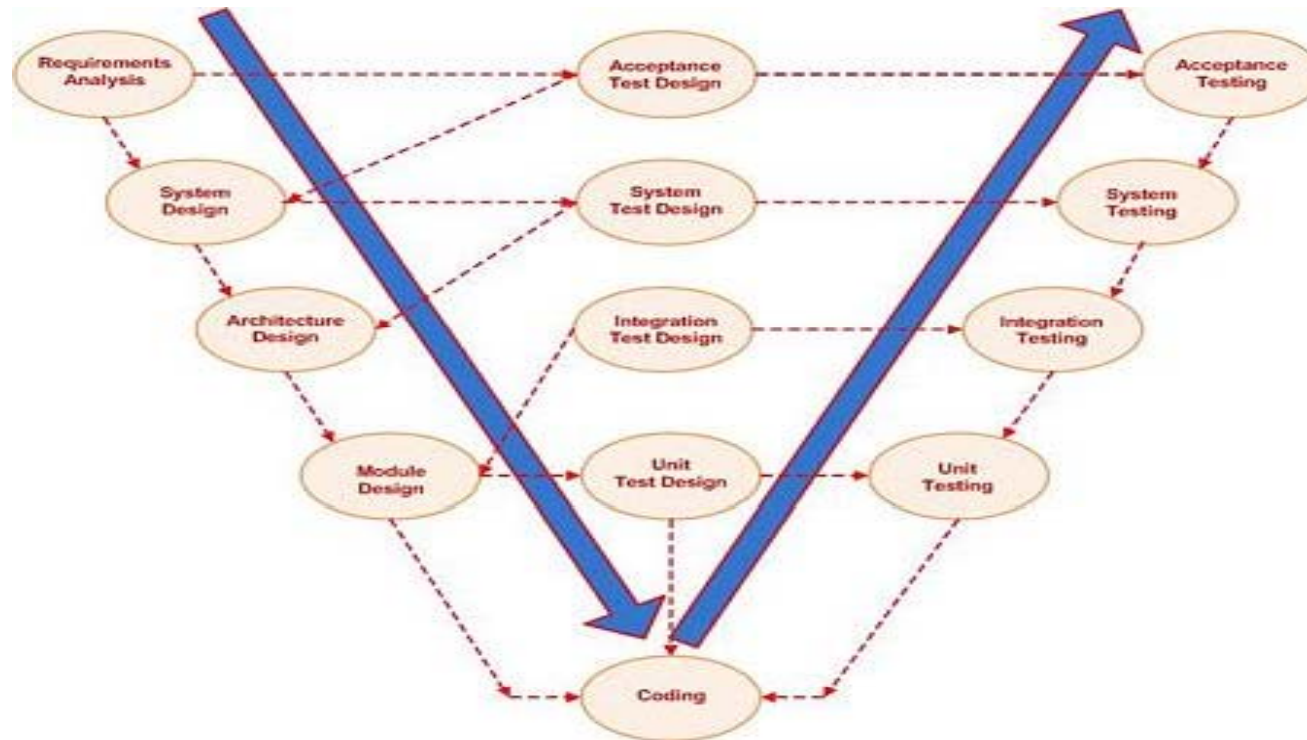
Successful software delivery requires alignment of business and technology domains...



Let's build a smarter planet



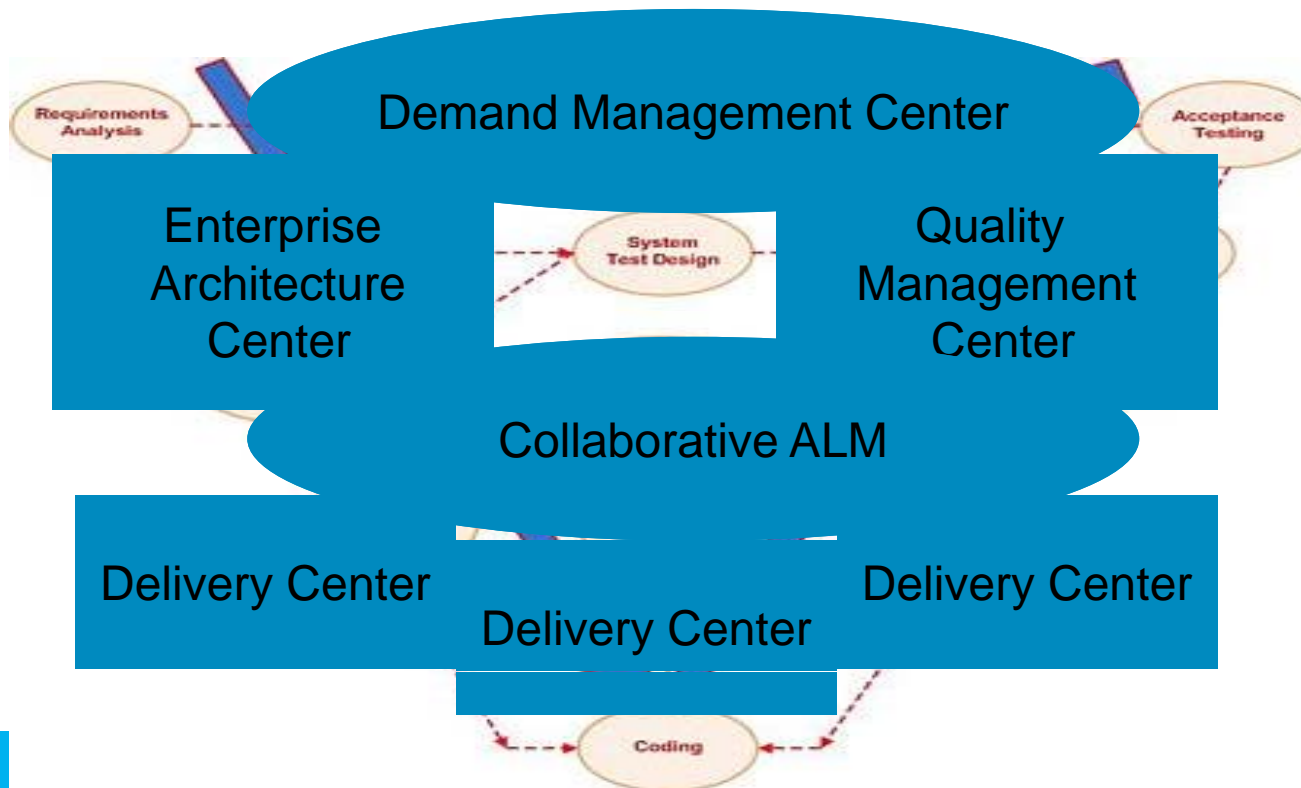
Implications for Organizations



Let's build a smarter planet



Implications for Organizations



Process Improvement depends on the integrations



What can we do in tools and methods to address these?

Top 3 reasons Application Lifecycle Management (ALM) fails to deliver promise

Distracted by day-to-day delivery pressures – 78%

Tools don't integrate properly – 62%

Lack the necessary internal expertise – 56%

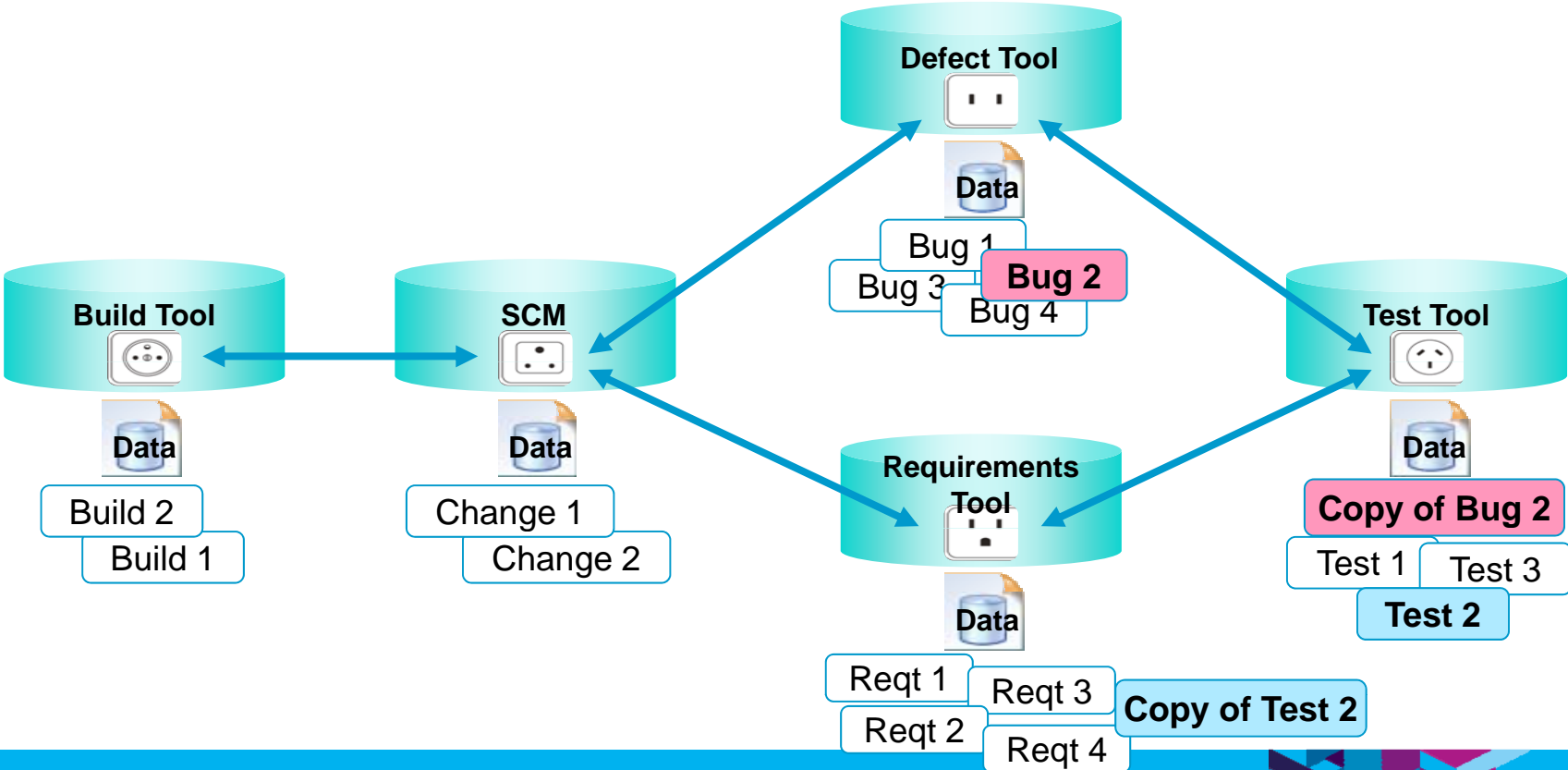


Source: Forrester study commissioned by Wipro, 2008

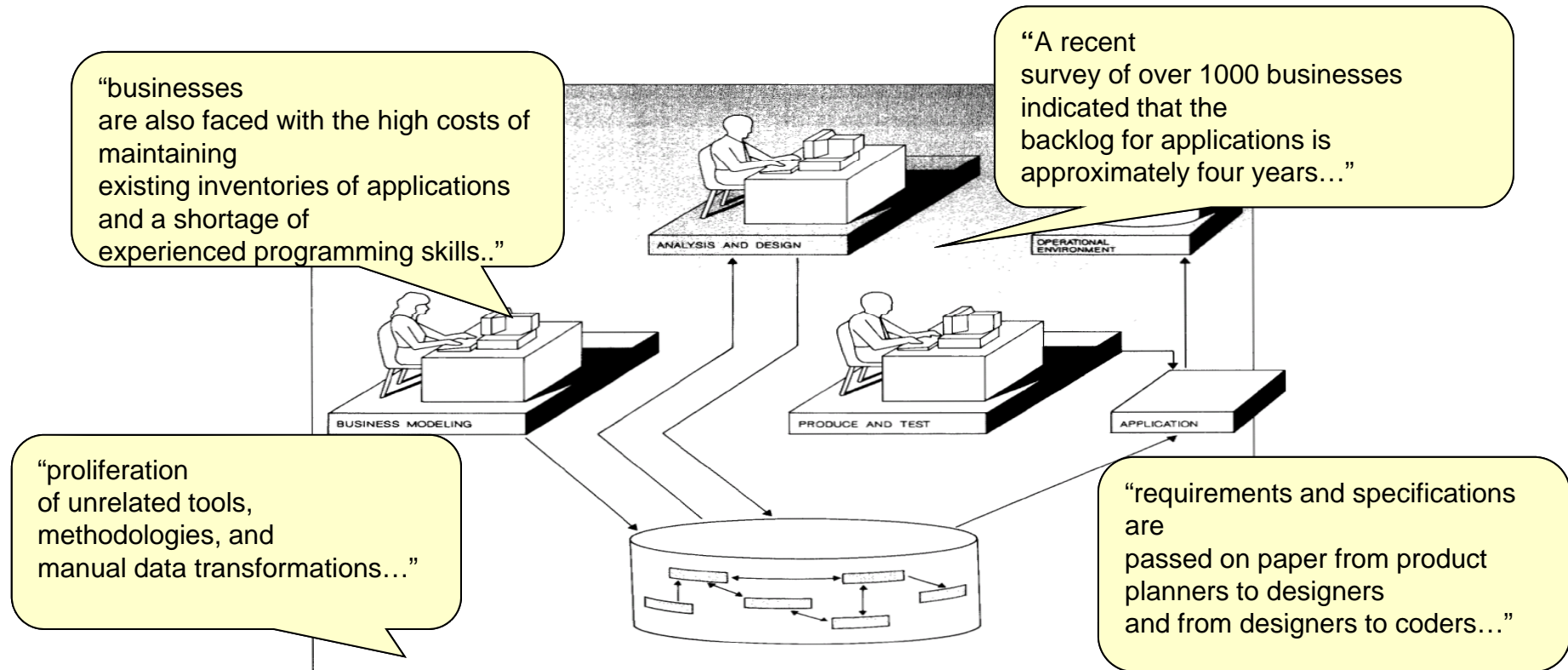
Let's build a smarter planet

IBM

Tool integration today



What did we say about this 20 years ago?



Source: Presentation on IBM's AD/Cycle, circa 1990!

Let's build a smarter planet



What is the state-of-the-art today?

Most other vendors still trying to build AD/Cycle

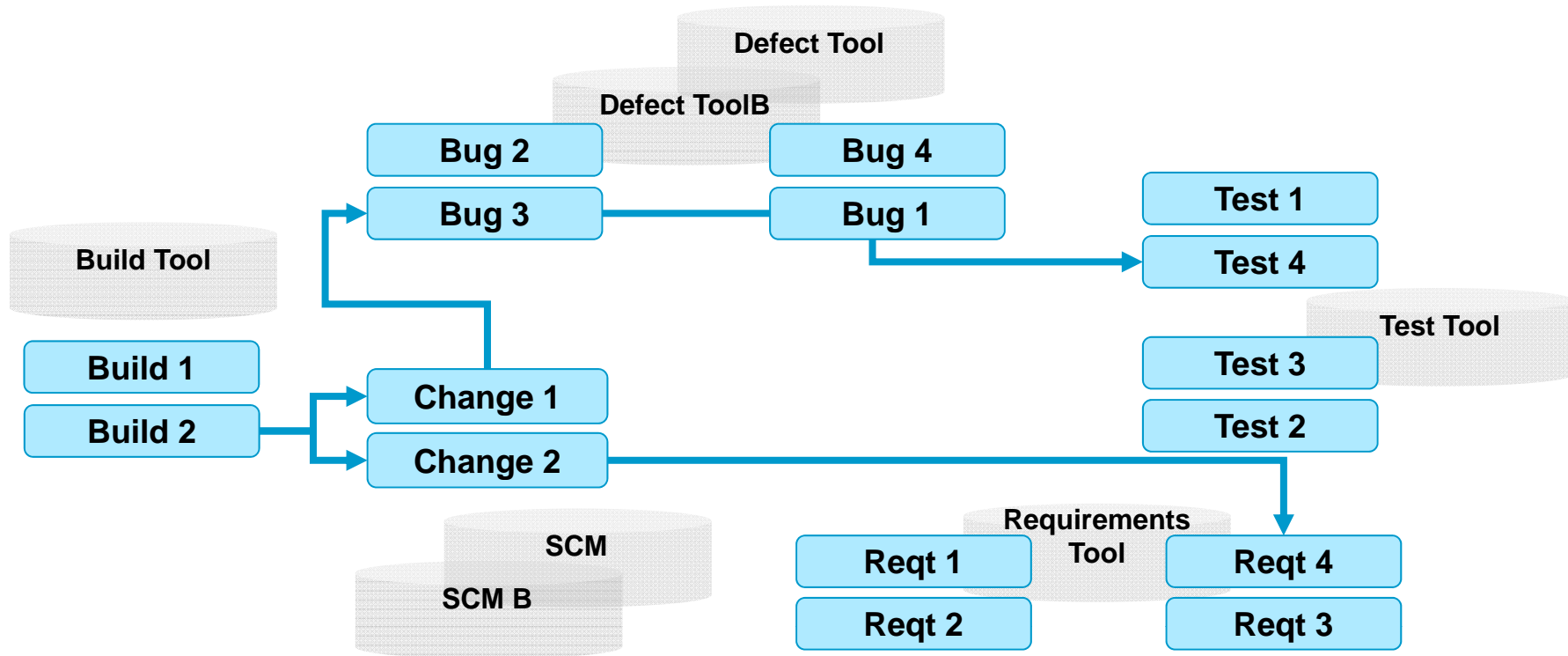
Requires all tools to integrate around centralized repository

- Data import (duplication) for foreign tools

Works as well as other centrally-planned economies have worked

- Do your company's needs match a fixed, pre-planned solution, or is an open, integrated economy a better model?

Another approach: Linked [Lifecycle] Data



What is Linked Data?

- Give everything a URL
- Return something interesting when clients perform a GET on a URL
- Link everything together – embed links to related things in representations
- Use standards like RDF and SPARQL

Linked Data – a major transformation

Adopting this simple conceptual model turns *everything* on its head

- There have only been two major model shifts in my 30+ year career. First was shift to client server from mainframe. This is the second.
- Nothing you thought you knew is valid anymore
 - The HTTP resources are central, your application a minor detail
 - The HTTP URLs are permanent reality, the data in the database a detail
 - Closed, fixed in scope -> open, extensible scope
 - Fixed in time -> everything evolves over time
 - Don't import data – address it where it is

What is RDF?

A “universal” data representation for the web

- Relational, IMS, COBOL, XML, object, ... data can all be expressed in RDF

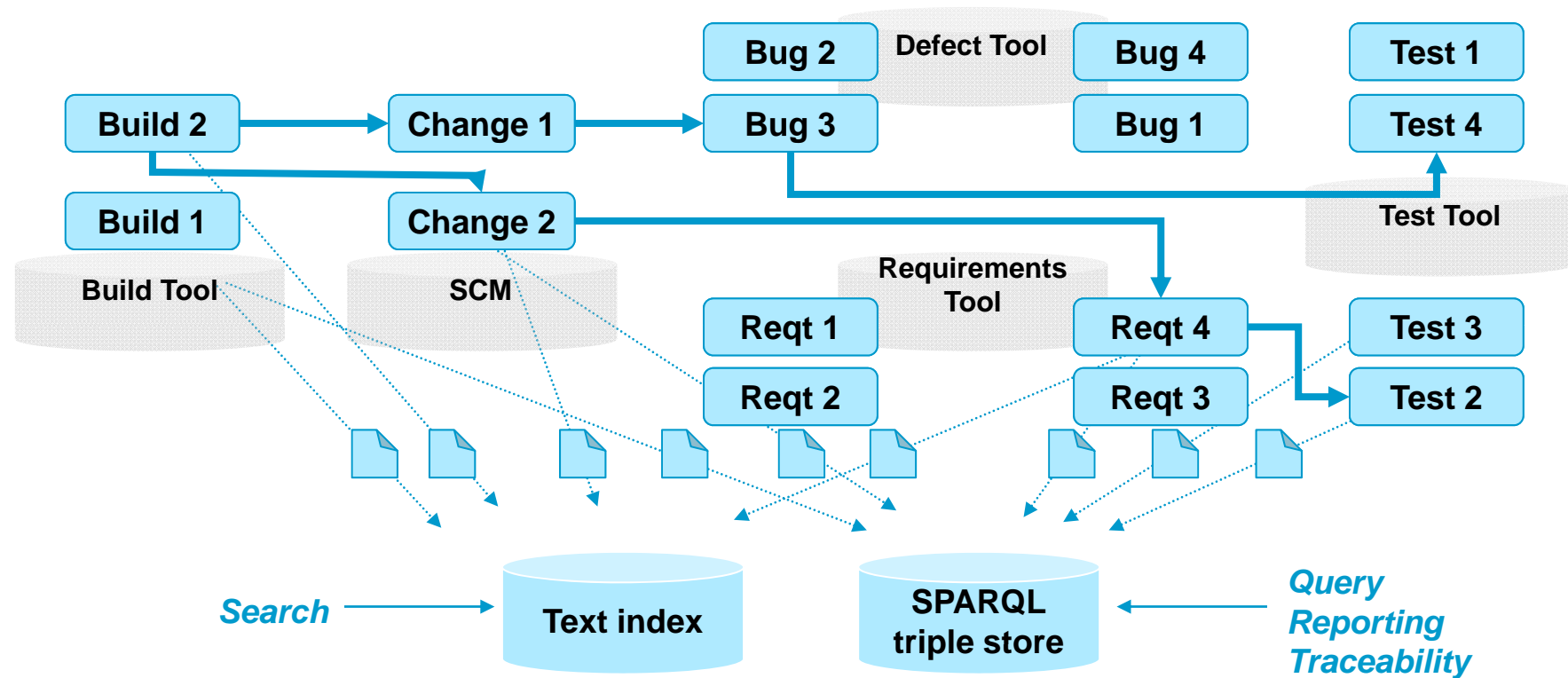
A very simple model and syntax for representing data

- RDF is like property, value pairs
- RDF adds “subject” – what is it the property of – so triples, not pairs
- RDF properties are themselves resources with URLs.

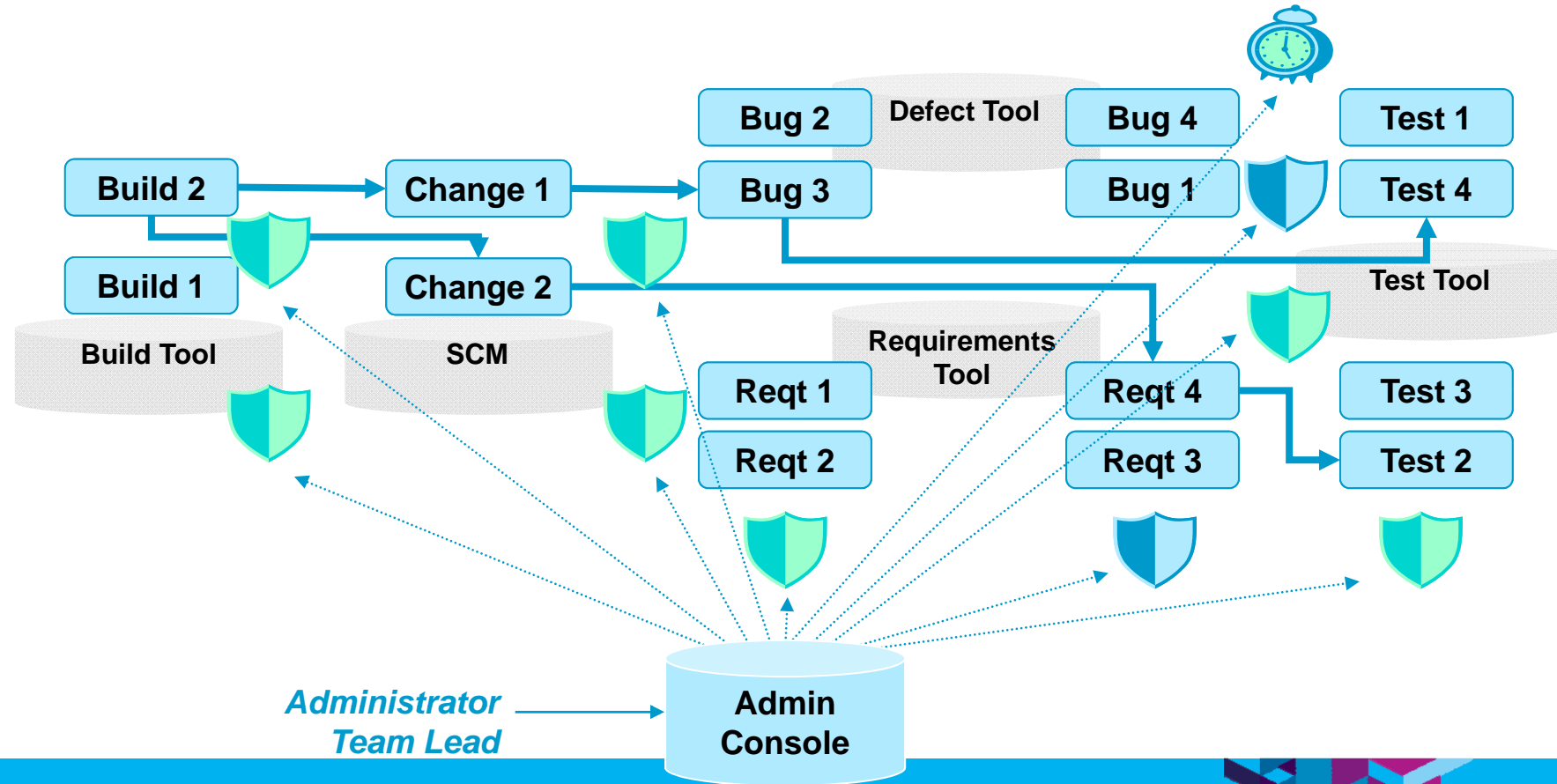
That’s about it! – most of the rest is hype and pretention, or detail

- RDF also can describe containers and collections
- RDF has the notion of type, but it’s not similar to OO type, it’s like type in the natural world.
- There is a language for querying over RDF, called SPARQL. (SPARQL adds graphs, so quadruples, not triples)
- You can write down RDF data in XML, as a twisted experiment of no value, but there are much nicer, more natural formats

Finding and analyzing data




Defining process rules



Open Services for Lifecycle Collaboration

Specifications for linked lifecycle data

[Home](#) [About](#) [Community](#) [Wiki](#) [Learn](#)



Open Services for Lifecycle Collaboration

open community. open interfaces. open possibilities.

Open Services for Lifecycle Collaboration (also known as OSLC or Open Services) is a community effort to help software delivery teams by making it easier to use lifecycle tools in combination. The OSLC community is creating open, public descriptions of resources and interfaces for sharing the things that software delivery teams rely on, like change requests, test cases, defects, requirements and user stories.

By agreeing on common specifications for lifecycle resources and the services to access them, we can eliminate traditional barriers between tools and open the door to new forms of collaboration. OSLC can bring value to software delivery teams and tool providers alike, from the most Agile to the most ceremonial of projects, and for commercially-licensed, open source, and internally developed tools. [More.](#)

With OSLC's open and scenario-based approach, businesses benefit from the ability to tie disparate tools together. This collaborative approach gives our consultants the flexibility to make lifecycle tool choices based on specific client project demands.

Randy Vogel, Accenture

Learn more

- [Presentation: ALM Integration in a Web 2.0 World](#)
- [Presentation: RESTful Work Items: Opening up Collaborative ALM](#)
- [Podcast: Open Services Bears first fruit. A conversation with Steve Abrams, Mik Kersten, and Carl Zetie.](#)
- [Whitepaper: The Case for Open Services](#)
- [Podcast: John Wiegand and Steve Abrams introduce the OSLC initiative](#)

News and events

- [Implementations delivered for Change management 1.0 spec \(press release\)](#)
- [Change management 2.0 spec workgroup expanding participants.](#)
- [Requirements management and Asset management workgroups draft early specs.](#)
- [Primer authored for Software Estimation and Measurement](#)
- [New Reporting workgroup call for participation.](#)

Quick links

- [Wiki: Open Services specifications](#)
- [Mailing list: OSLC community](#)
- [Blog: Let's try something different - Carl Zetie's commentary on OSLC](#)
- [Twitter - follow us: @oslcNews](#)

[Terms of Use](#) [Privacy](#) [Feedback](#)

An open community of individuals from industry, commercial tools vendors, systems integrators, open source projects, and academia.

Focusing on sharing of lifecycle data (requirements, test cases, change requests) between tools and across the lifecycle.

Taking a technology-neutral approach based on Internet standards and protocols.

Operating at open-services.net

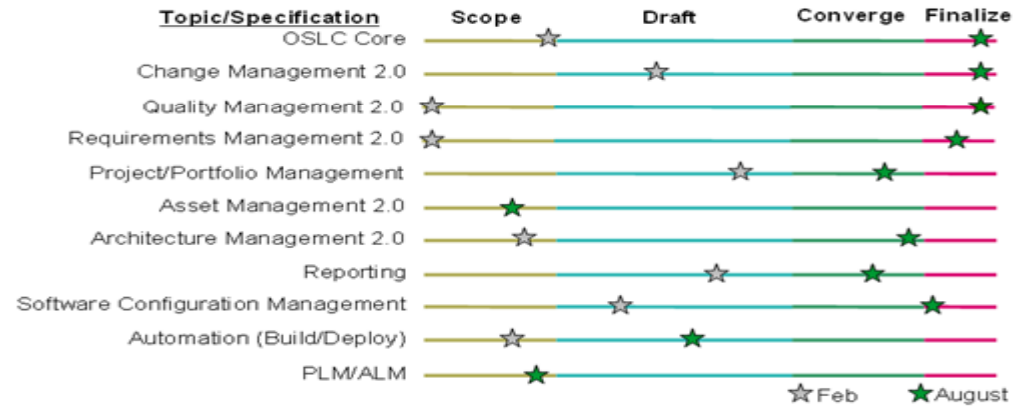
OSLC Community

Eleven workgroups operating at open-services.net

- Intensive focus in 2010 on Core and CLM related specs (CM, RM, QM, Arch Mgmt, SCM)
- PLM/ALM workgroup defining cross-cutting scenarios and driving a systems perspective

Continuing to grow

- 345+ registered community members (up from 70 people at RSC 2009)
- Individuals from 34+ different companies have participated in OSLC workgroups (up from 5 companies at RSC 2009)



- | | |
|---------------------------|----------------------------|
| Accenture | Lender Processing Services |
| APG | Northrop Grumman |
| Black Duck | Oracle |
| Boeing | QSM |
| BSD Group | Rally Software |
| Citigroup | Ravenflow |
| EADS | Shell |
| Emphasys Group | Siemens |
| Empulsys | Sogeti |
| Ericsson | SourceGear/Teamprise |
| Fokus Fraunhofer | State Street |
| Galorath | Tasktop (Eclipse Mylyn) |
| General Motors | Thales |
| Health Care Services Corp | Tieto |
| IBM | TOPIC Embedded Systems |
| Institut TELECOM | UrbanCode |
| Integrate Systems | WebLayers |

OSLC Core Spec

Applies to all resources in an OSLC system.

Tries to answer some simple questions

- What URLs can I POST to create new resources?
 - What properties could/should I set when POSTing to these URLs?
- How do I query the resources already POSTed at an URL?
 - What properties might be available to query on a set of resources?
- How is pagination of large representations handled?
- How can I delegate to the UI of another service, instead of dealing with its data?
- Best practices for expressing hyper-links between resources

A bit like a superset of APP, except ...

- Linked data compatible, Generic - doesn't require you to model your domain as a blog (feed, entry), Simpler, Solves more problems

Other OSLC specs

Adhere to Core spec and add domain-specific vocabularies

- Change Management
- Requirements
- Assets
- Tests
- Estimation
- Source Code Management/ versioning
- Reporting
- Architecture
- Project/portfolio
- Automation (e.g. build)

Linked Data Challenges

Most of the current web is read-only

- Most content created “conventionally” and then published read-only
- APP is an exception

Security

- Web authentication protocols are embryonic – e.g. OAuth
- Google – everything is public
- Enterprise search – typically everyone in enterprise can see

Ontologies (odious, pretentious word)

- Need to agree on common terms like “name”, “type”, “title”, “identifier”,
- Need to agree on domain-specific terms (defect, task, requirement, test case, build, ...)

There are many distractions

My “friends don’t let friends do ...” list

- XML
- Web Services
- Data formats specified in programming language technologies
- ATOM Publishing Protocol
- Data specified with object-oriented concepts (classes, instances)

Linked Data Challenges - detail

- Don't write back-links (they will get out of synch)
- Don't assume closed schema (let others add properties)
- Don't assume you can "move" data – URLs are forever
- Think of everything as a resource not operationally
 - "a list of bugs", "the first page of a list of bugs" and "bugs whose id is 8" are separate resources – not one resource with "arguments".
<http://example.com/bugs>, <http://example.com/bugs?oslc.paging=true>,
<http://example.com/bugs?oslc.where=dcterms:identifier=%228%22>
- Think of scope as global
 - Users are global, not defined by an application (accounts can be local)
 - "Type descriptions" are global (defects, requirements, ...)

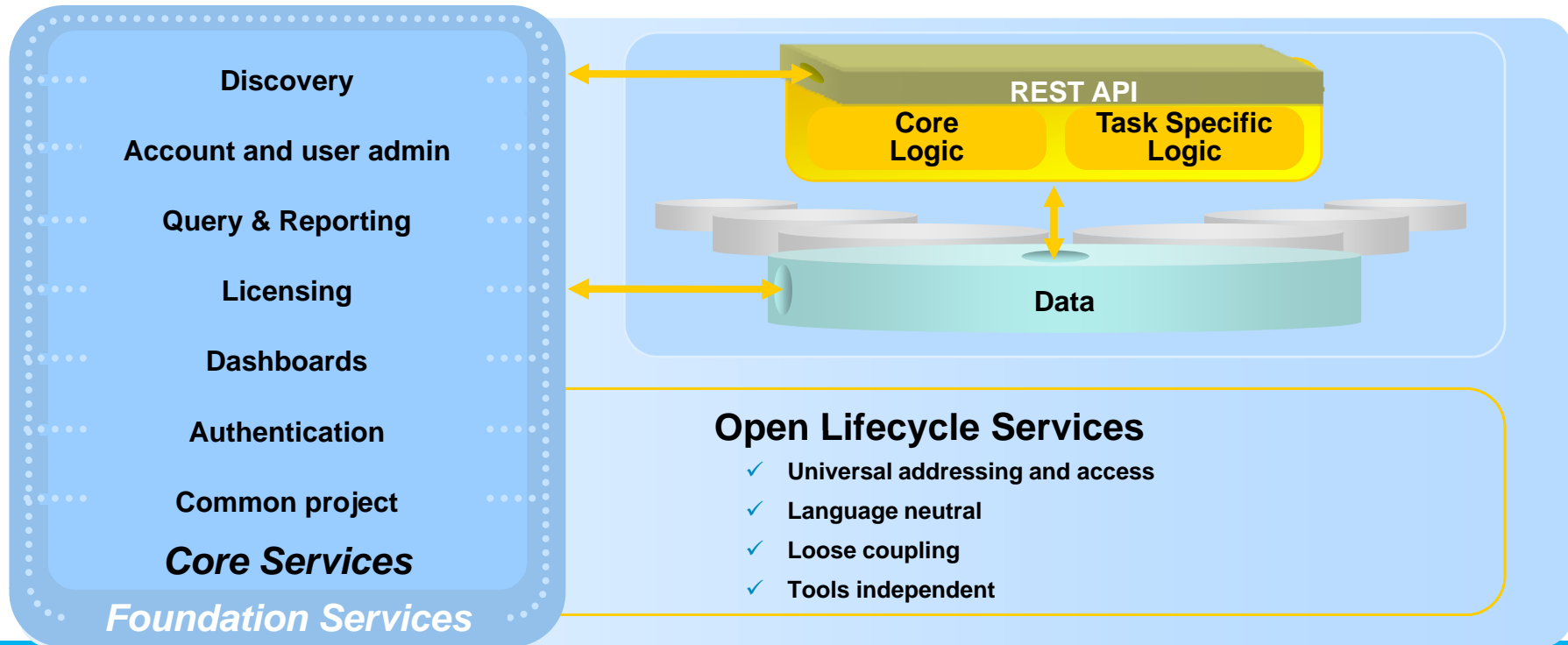
Linked Data Challenges - detail

- Think of “policy that changes with time” not “inherent characteristics”
 - E.g. Defects must have a priority between 1 and 3 – changeable policy.
- Write resource-centric UI, not desktop-application-in-a-browser
 - Much more composable – can adapt to change with time, scenario
 - Better performance
- Assume applications are “black boxes” – use protocols, not frameworks to integrate (c.f. Eclipse)

Linked Data Challenges – unsolved problems

- Versioning beyond SCM
 - WebDav versus Eclipse models
 - PLE “variants”

Jazz: Open, extensible, web-centric, integration platform



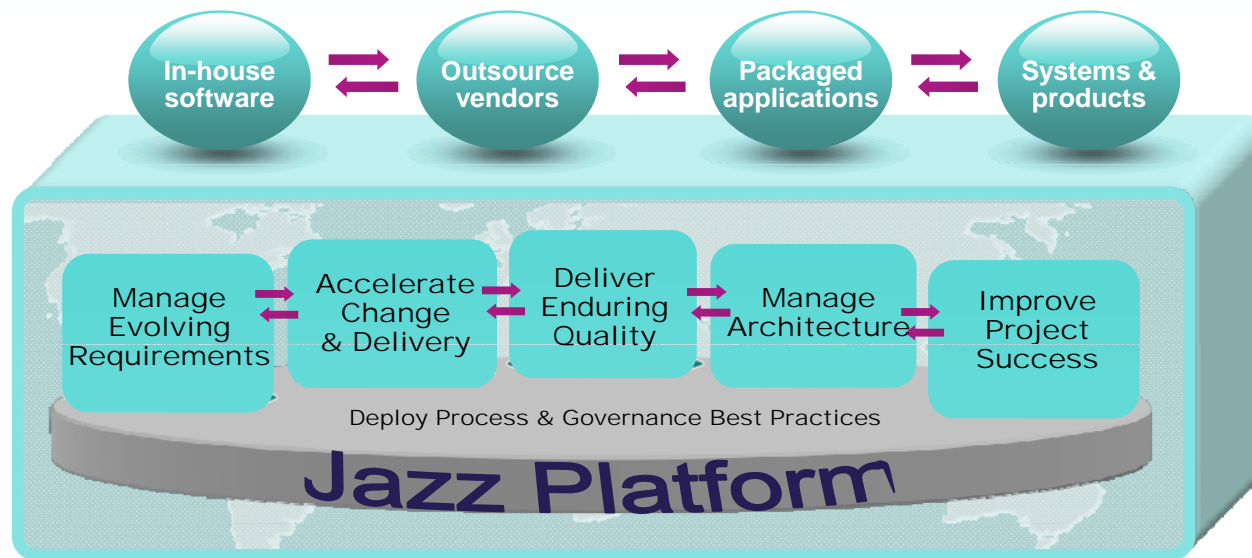
Let's **build** a smarter planet



IBM Rational Software Delivery Platform

Rational. software

***Solutions** to help customers achieve greater value and performance from their investments in delivering software*



- *Enterprise Modernization and Transformation*
- *Organizational Governance*
- *Skill Development and Community*
- *Implementation Services*

Let's **build** a smarter planet.

IBM.

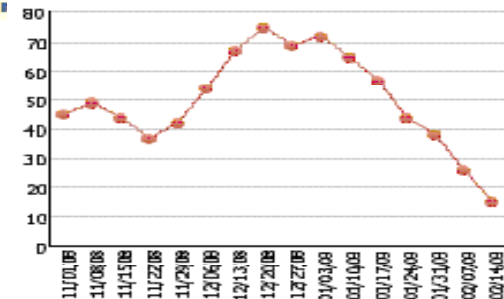
Executive Dashboards

Region: United States

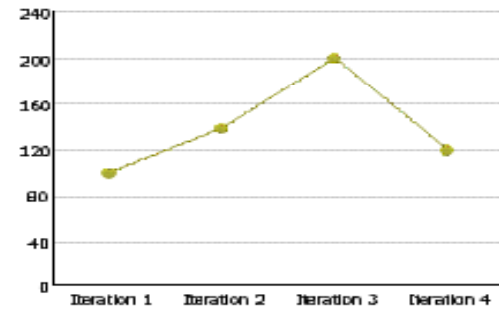
# of Head Count	Role						Total (Project)
	Architect	Developer	Doc	Marketing	Sales	Tester	
	2	4	5	2	2	4	47

Smarter Planet > Online Auction

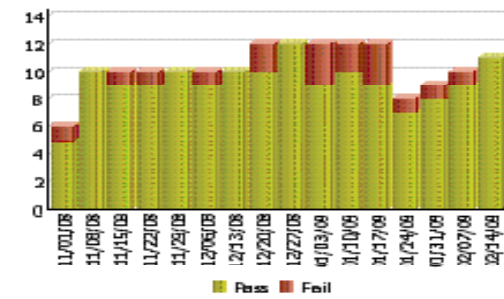
Outstanding Work



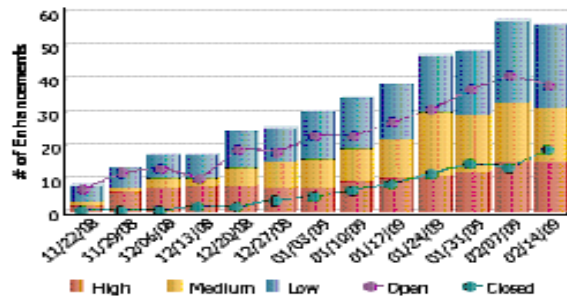
Iteration Velocity



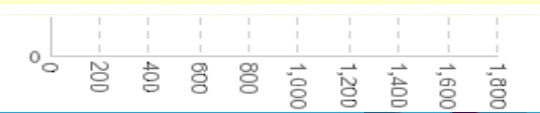
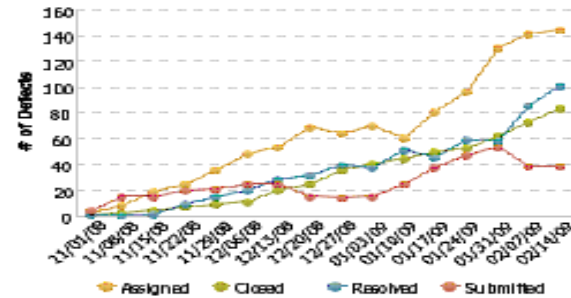
Build Health



Enhancement Request Backlog



Actual Defect Trends



Governance and Control of Software Delivery



Let's build a smarter planet





© Copyright IBM Corporation 2010. All rights reserved. The information contained in these materials is provided for informational purposes only, and 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, these materials. Nothing contained in these materials 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. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

Let's build a smarter planet

