

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.



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%

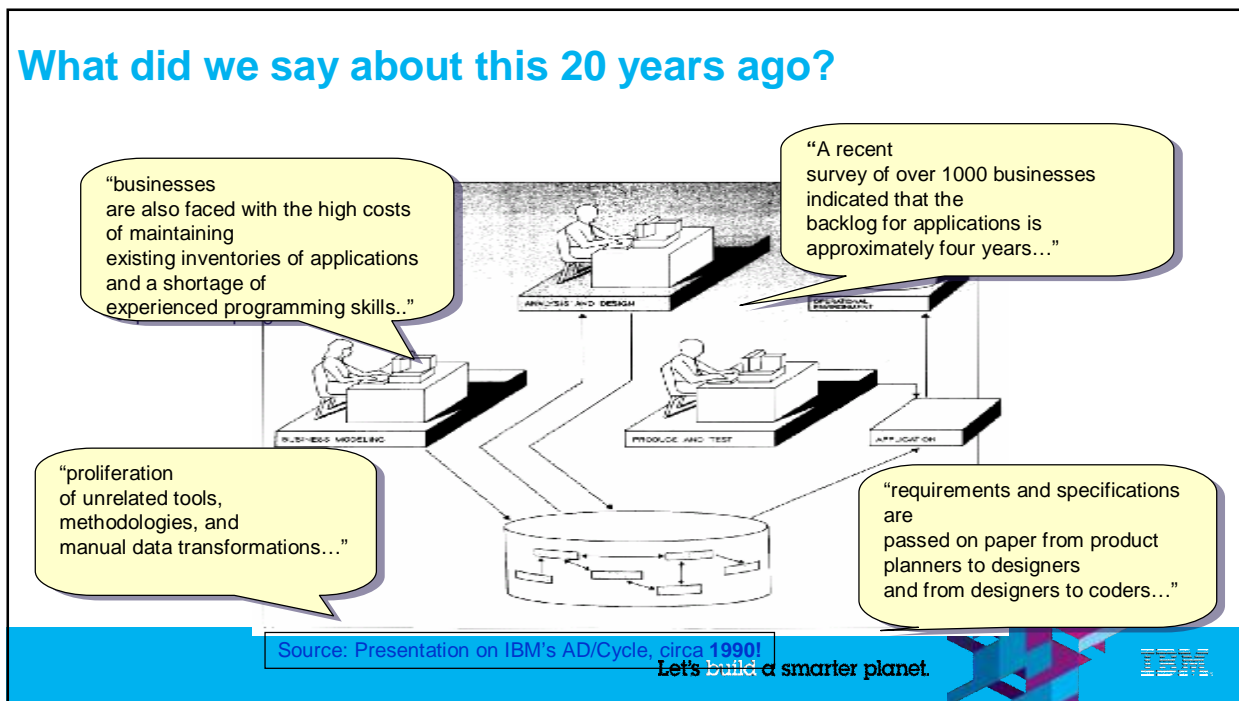
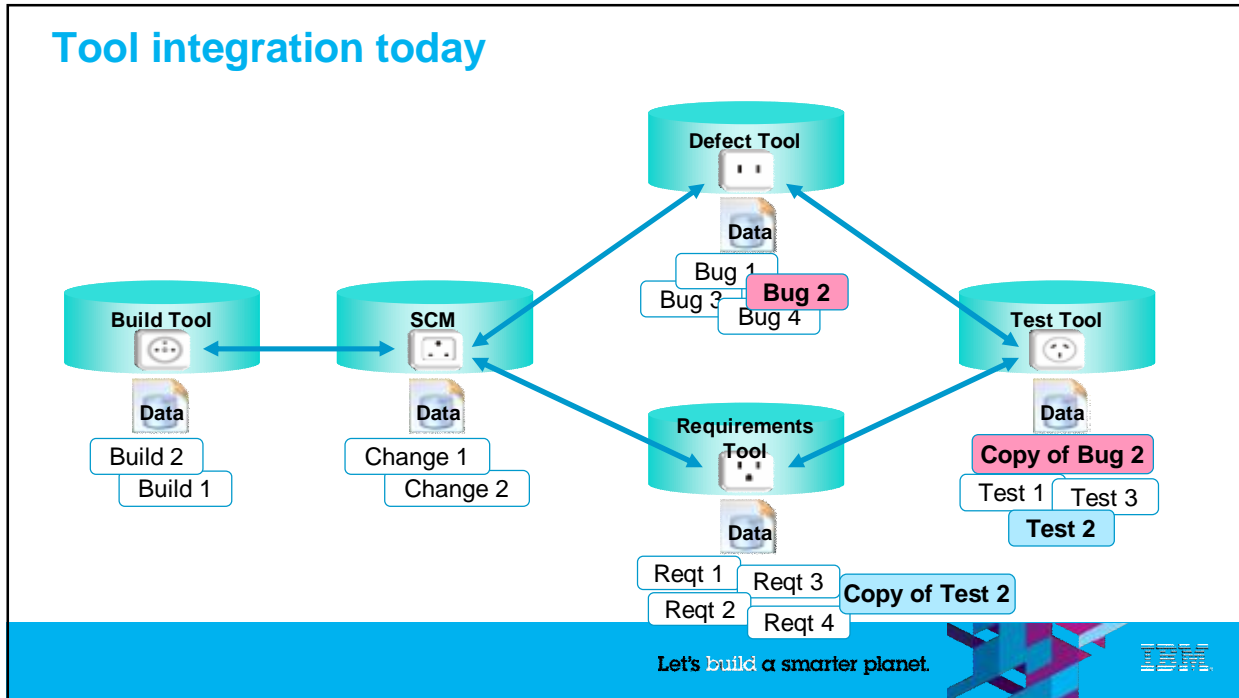
Lack the necessary internal expertise – 56%



Source: Forrester study commissioned by Wipro, 2008

Let's build a smarter planet.





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

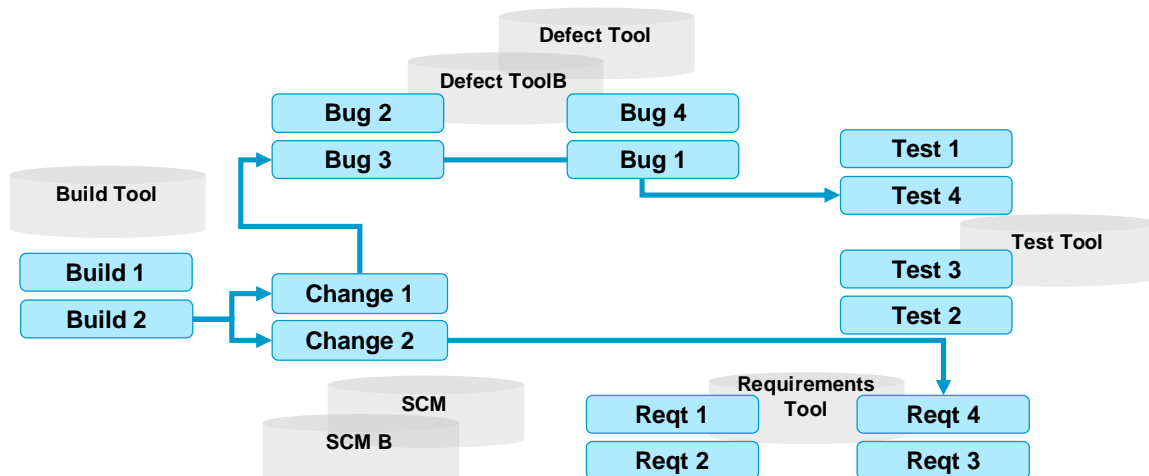
Works as well as other centrally-planned economies have worked

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

Let's build a smarter planet.



A new approach: Linked [Lifecycle] Data



Let's build a smarter planet.



What is Linked Data?

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs. so that they can discover more things.

Isn't this just REST?

Let's build a smarter planet.



Linked Data – a major transformation

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.

Adopting this simple model turns everything on its head.

- 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

Let's build a smarter planet.



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 on the world wide web

- 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

Let's build a smarter planet.



An example - <http://vh1.example.com/requirements/r1>

@prefix oslc-rm: <<http://open-services.net/ns/rm#>>.

@prefix dcterms: <<http://purl.org/dc/terms/>>.

@prefix rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#> >

<> rdf:type oslc-rm:Requirement.

<> dcterms:title "Requirement 1".

<> dcterms:description "The system should respond within 1 second".

Let's build a smarter planet.



An example - <http://vh1.example.com/testcases/tc1>

```

@prefix oslc-qm: <http://open-services.net/ns/qm#>.
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns# >
<> rdf:type oslc-qm:TestCase.
<> dcterms:title "TestCase 1".
<> dcterms:description "Verifies the system responds within 1 second".
<> oslc-qm:validatesRequirement <http://vh1.example.com/requirements/r1>.

```

Don't assume what sort of resource the requirement is – open system!!

Let's build a smarter planet.



An example showing why subject is important -

[http://vh1.example.com/testcases/defects?oslc.where=oslc_cm:inprogress="true"](http://vh1.example.com/testcases/defects?oslc.where=oslc_cm:inprogress='true')

```

@prefix oslc-qm: <http://open-services.net/ns/qm#>.
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns# >
<> oslc:nextPage <http://vh1.example.com/testcases/defects/.../page2>.
<http://vh1.example.com/defects>
  rdf:member <http://vh1.example.com/defects/00001>;
  rdf:member <http://vh1.example.com/defects/00002>;
  rdf:member <http://vh1.example.com/defects/00003>.

```

Let's build a smarter planet.



Another example showing use of subject - http://members.cox.net/martin_nally

@prefix dbpp: <<http://dbpedia.org/property/>>.
 @prefix dbpr: <<http://dbpedia.org/resource/>>.
 @prefix foaf: <<http://xmlns.com/foaf/0.1/>>.
 @prefix xsd: <<http://www.w3.org/2001/XMLSchema#>>.
 @prefix dcterms: <<http://purl.org/dc/terms/>>.
 <> dcterms:description "a web site for Martin Nally".
 <<http://martin-nally.name>>
 dbpp:birthname "Martin Paul Nally";
 dbpp:birthDate "1957-01-05"^^xsd:date;
 dbpp:birthPlace dbpr:Scotland;
 foaf:depiction <http://members.cox.net/martin_nally/Martin_Nally.6.email.jpg>.

Let's build a smarter planet.



Open Services for Lifecycle Collaboration *Specifications for linked lifecycle data*

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

Let's build a smarter planet.



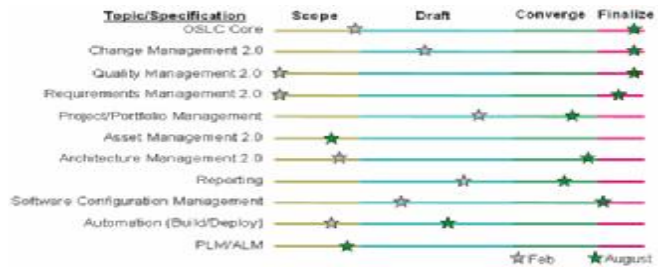
OSLC Community

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
APC	Northrop Grumman
Black Duck	Oracle
Boeing	OSM
BSD Group	Rally Software
Citigroup	Ravenflow
EADS	Shell
Emphasys Group	Siemens
Empulsys	Sogefi
Ericsson	SourceGear/Teamrise
Fokus Fraunhofer	State Street
Gabarrath	Tasktop (Eclipse Mylyn)
General Motors	Thales
Health Care Services Corp	Tieto
IBM	TOPIC Embedded Systems
Institut TELECOM	UrbanCode
Integrate Systems	WebLayers

15

Let's build a smarter planet.



OSLC Core Spec

Applies to all resources in an OSLC system.

Tries to answer some simple questions on how to use linked data

- 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 (e.g. link properties)
- Partial Update (there is a reason that SQL has no equivalent of PUT, only PATCH)

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

Let's build a smarter planet.



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)

Let's build a smarter planet.

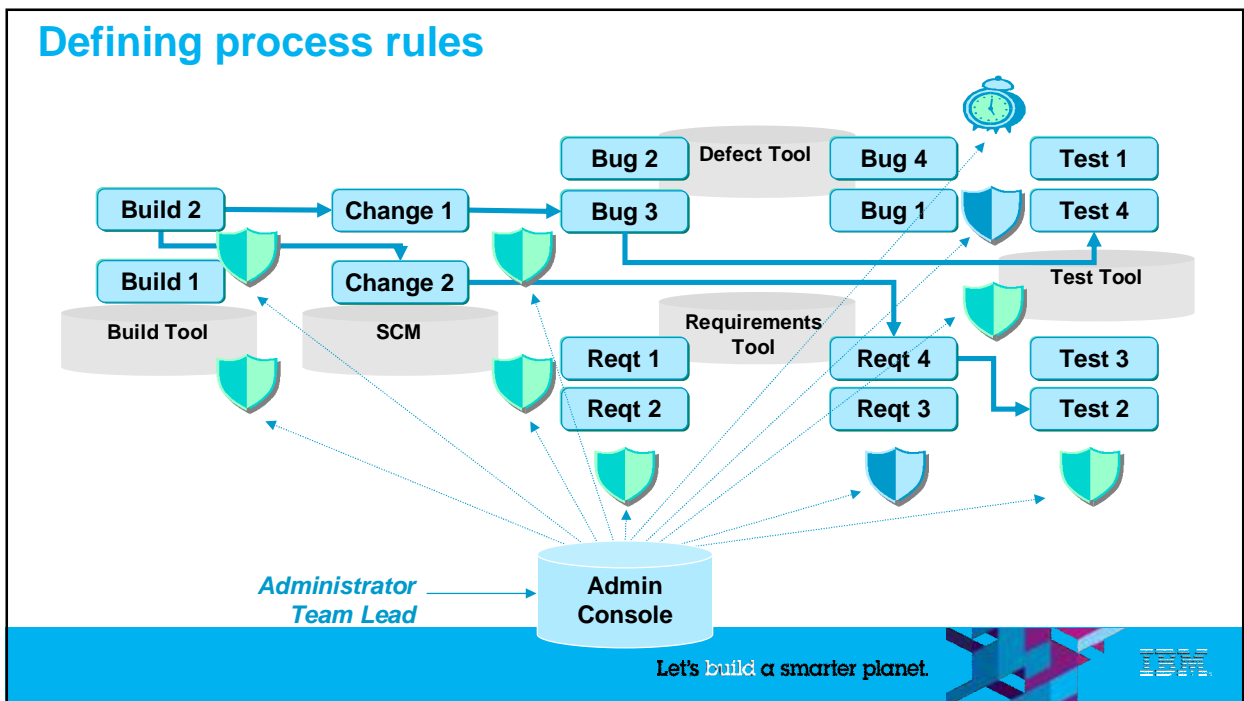
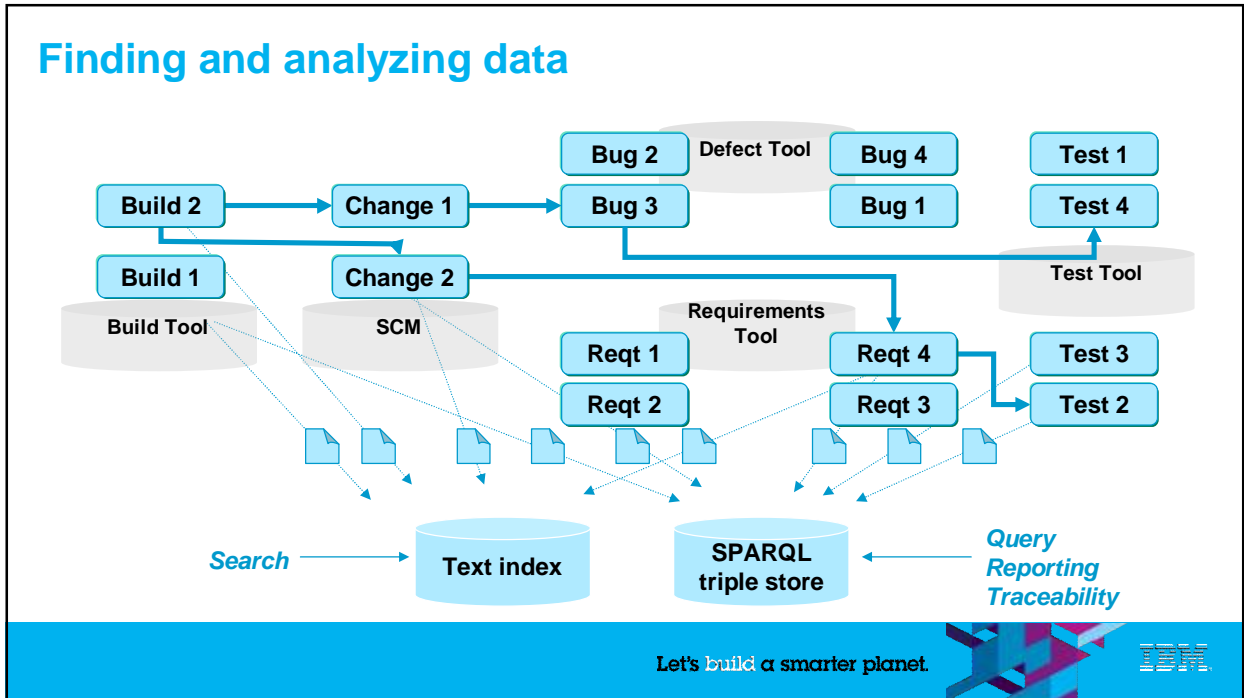


Ontologies (odious, pretentious word)

- Need to agree on common terms like “name”, “type”, “title”, “identifier”
 - Want to query across all resources, not just within types
 - We like Dublin Core. Maybe rdfs (for label). Maybe foaf for Person.
- Need to agree on some domain-specific terms
 - Don't try to define all the properties of a resource like defect
 - Every team/organization wants different ones
 - Focus on those properties that are important for integration scenarios
 - E.g. Is a defect closed?
 - E.g. What requirement does this test-case test?

Let's build a smarter planet.





friends don't let friends do ...

XML

- OSLC core spec has some “features” to keep the XML zombies happy L
 - RDF/XML mandated
 - forcing “blank nodes” to ease XML parsing (“Local Resource”) Web Services

ATOM Publishing Protocol

Data formats or interfaces specified in programming language technologies

- Data specified with object-oriented concepts (classes, instances)

Let's build a smarter planet.



Think conceptually, don't think like a programmer

“a list of bugs”, “the first page of a list of bugs” and “bugs whose id is 8” are independent resources – not one resource with “arguments”.

- <http://example.com/bugs>
- <http://example.com/bugs?oslc.where=dcterms:identifier=%228%22>
- <http://example.com/bugs?oslc.paging=true>

“oslc:pagination=true” is also “thinking like a programmer”. Better would be

- <http://example.com/bugs?oslc.firstPage>

Let's build a smarter planet.



Most of the current web is read-only

Most content created “conventionally” and then published

- Blogs, tweets, wikis are exceptions
- APP is a protocol for blogs (?)
- Doing “authoring on the web” for a new domain requires learning

Let's build a smarter planet.



Cool URLs last forever

Don't assume you can “move” data

- Use virtual host names, not ones tied to machines

Don't put any “meaning” into URLs

- It will change

Let's build a smarter planet.



Security

• Web authentication protocols are embryonic – e.g. OAuth

- Google – everything is public
- Enterprise search – typically everyone in enterprise can see

What is scope of “user”?

What is language for ACLs?

Let's build a smarter planet.



Miscellanea

Don't write back-links (they will get out of synch)

Don't assume closed schema

- Ideally, let others add properties to existing types

Don't assume what is at the other end of a “link”

Avoid “local resources”

- Users are global, not defined by an application (accounts can be local)
- “Type descriptions” are global (defects, requirements, ...)

Let's build a smarter planet.



Does the application own storage?

Typical Application model:

- Data access is through an application, application controls integrity
- Storage is an application concern, totally private and fixed

Traditional IDE model:

- Data in files, multiple tools work on the files, files may be all screwed up
- Permanent Storage (e.g. SCM versioning) is a peer application - file system is just a temporary shared cache between applications and permanent storage

What is web equivalent of IDE model?

Let's build a smarter planet.



Linked Data Challenges - detail

Think of “data policy that changes with time” not “inherent characteristics”

- E.g. Defects must have a priority between 1 and 3 – changeable policy.

Assume applications are “black boxes” – use protocols, not frameworks to integrate (c.f. Eclipse)

Let's build a smarter planet.



Fat client or resource-oriented UI?

You can write fat clients in the browser too

- That is exactly what most experienced programmers will do

Fat clients have good support for specific workflows

Fat clients have problems

- Long load times
- Closed systems (what do you do when a link leads to pdf, or html or other?)
- More difficult to evolve when workflows change

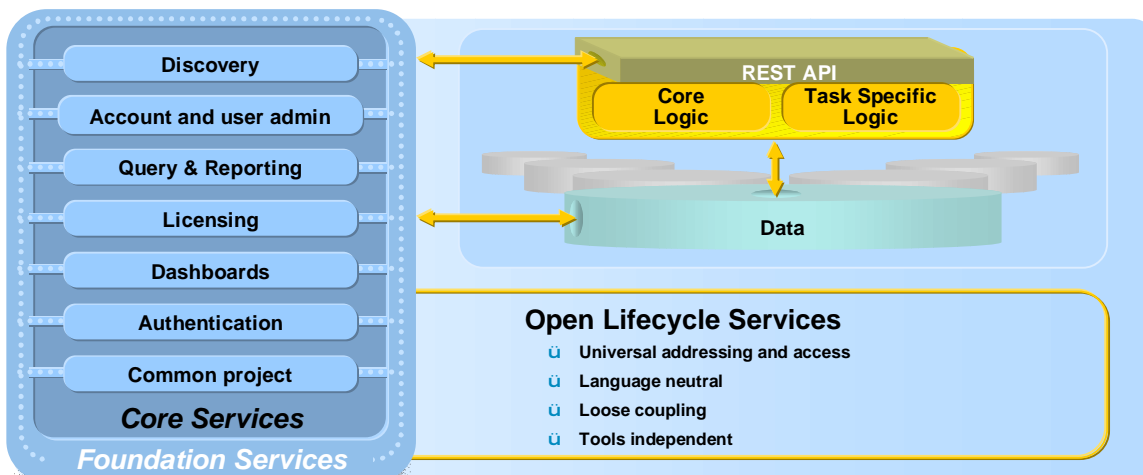
Another option is “page per resource” UIs

- Embrace page switches

Let's build a smarter planet.



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



Let's build a smarter planet.



IBM Rational Software Delivery Platform

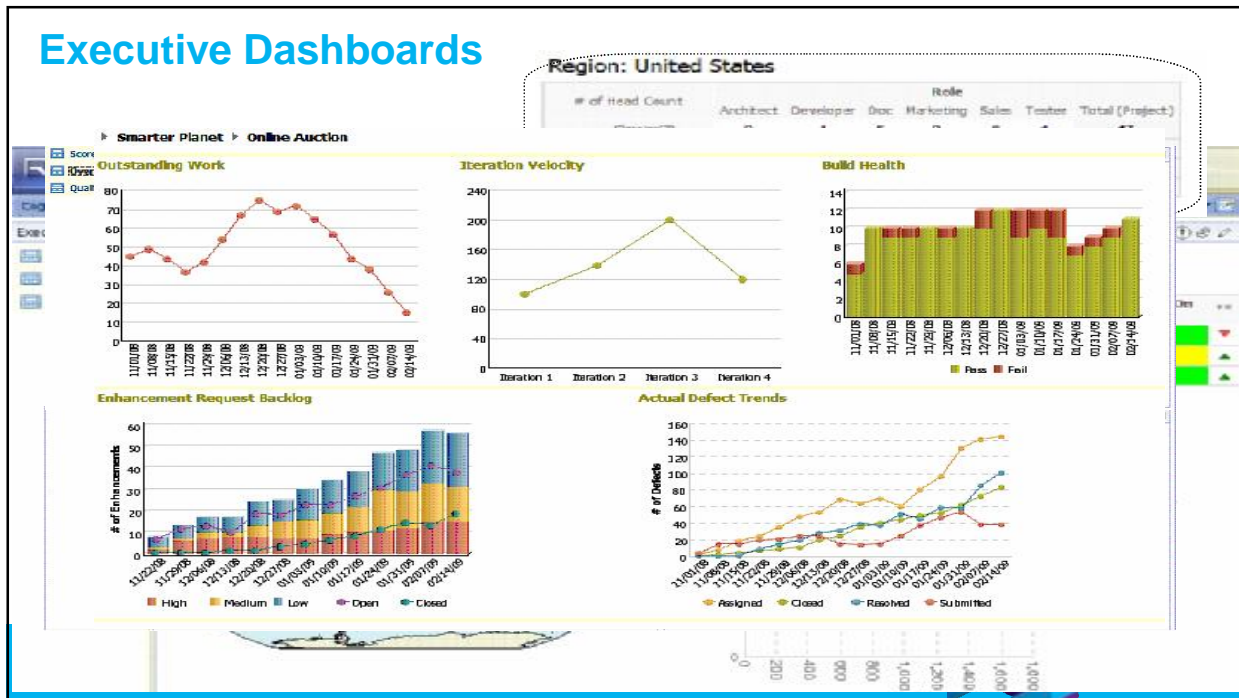
Rational software *Solutions to help customers achieve greater value and performance from their investments in delivering software*

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

Let's build a smarter planet.

Governance and Control of Software Delivery

Let's build a smarter planet.



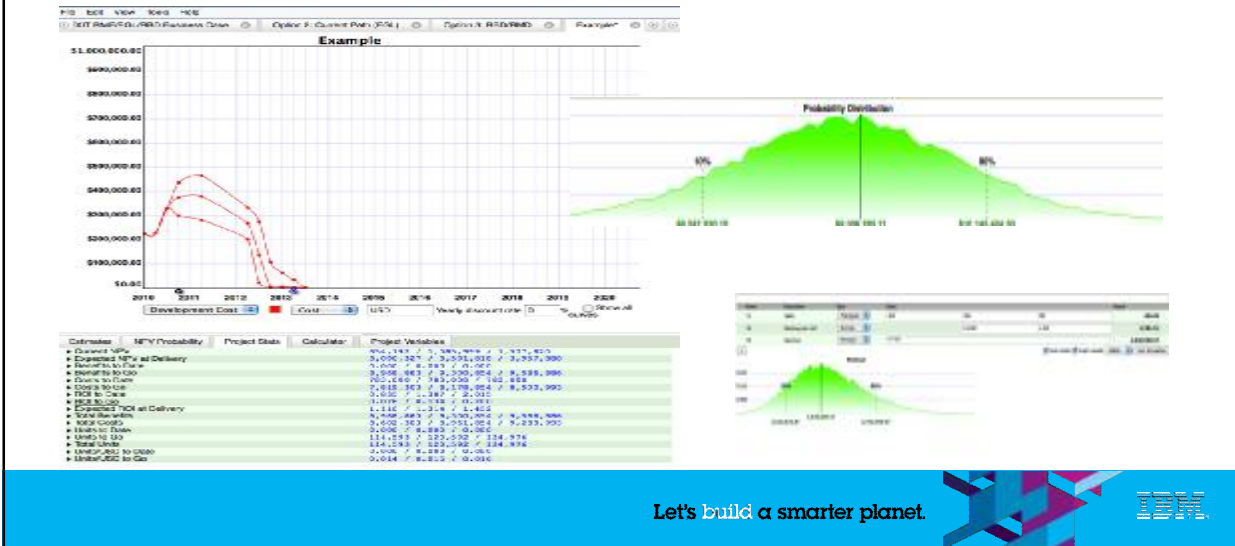
Something completely different

The future is uncertain – reason about probabilities

- Pierre de Fermat and Blaise Pascal, correspondence (1654)
- Christian Huygens (1657)
- Jacob Bernoulli (1713)
- Abraham de Moivre (1718)
- Thomas Bayes (1763)
- Pierre-Simon Laplace (1774)



Financier – using probability distributions to evaluate project and portfolio value



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.

