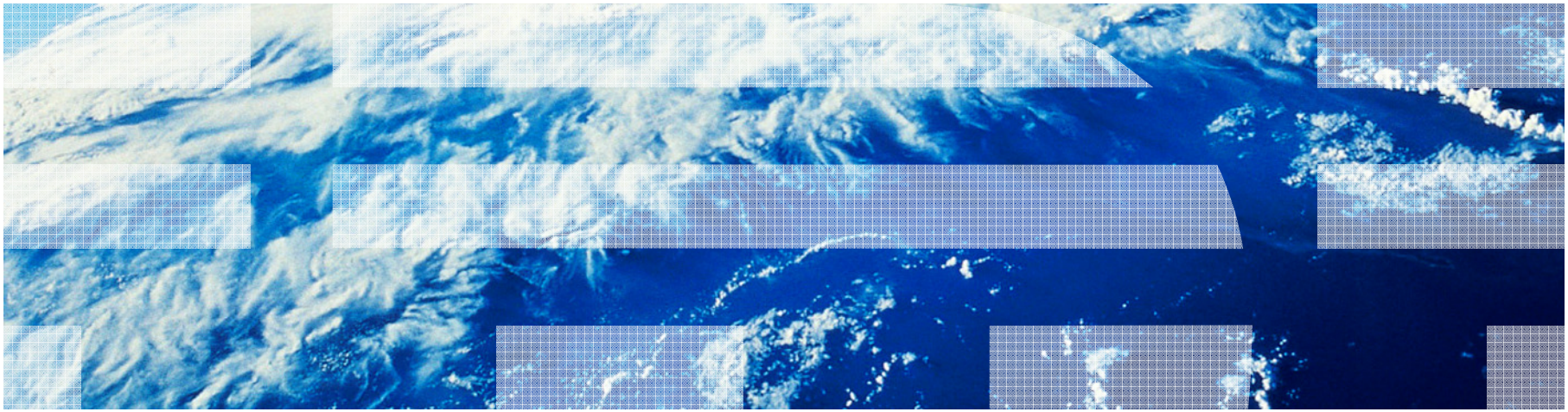
Current Trends and Future Directions in Technology for DevOps



Please note

The information contained in this presentation is provided for informational purposes only. 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. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. 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. 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. Product release dates and/or capabilities referenced in this presentation 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. 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, savings or other results.

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.

Outline

- Business promise of tools
- Business and Development
- Development and Operation

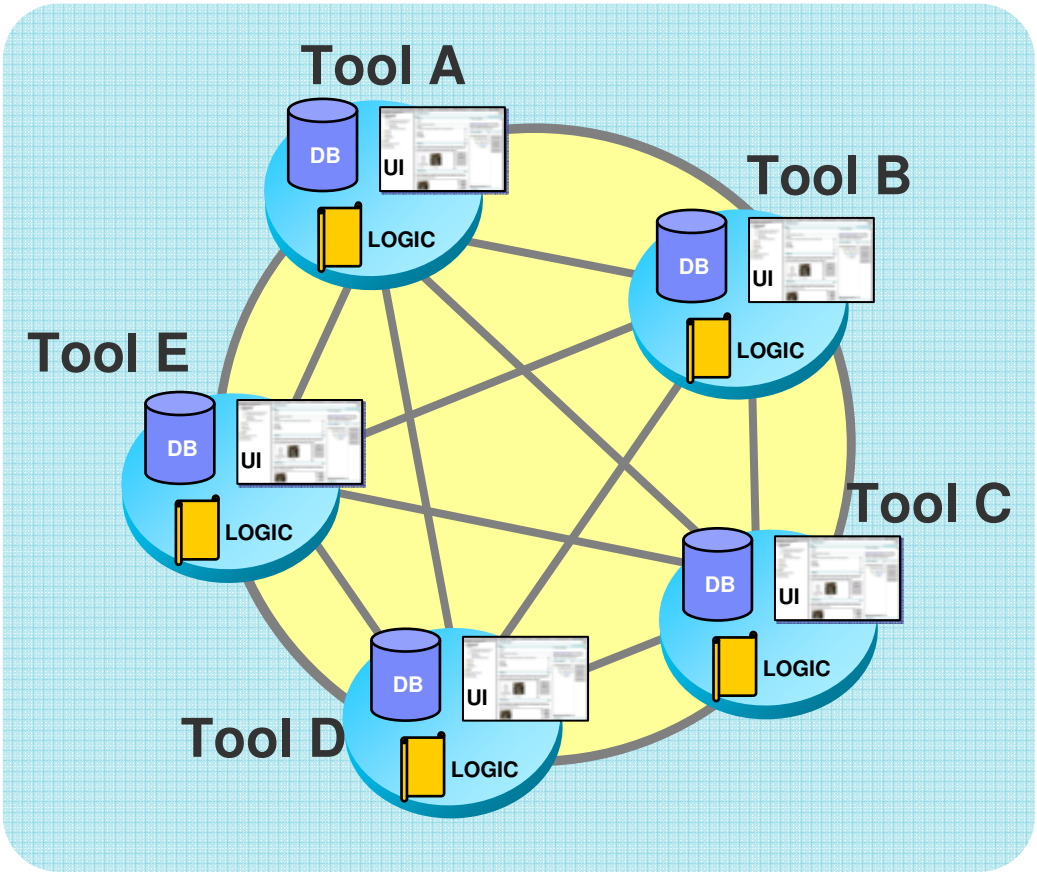
The Business Promise of Tools Is Widely Anticipated

Companies acquire tools with the best of business-centric aspirations

- Higher quality
- More customer satisfaction
- Aligning business and IT
- Faster time to market
- Lower costs/higher productivity
- More predictable delivery

Reality adds significant complexity

- Many tools from many vendors
  - **Heterogeneous environments** that are flexible for partners and suppliers
- Many teams in many places
  - **Distributed** development, cross site product development
  - Many levels of teams
    - PMO, Bus, dev teams, ops teams, etc
- Coherent process
  - **Flexible** and **robust** process
    - supporting Lifecycle / Agile Methods
  - **Measure** and **improve** effectiveness



What companies want to achieve

1. Communication of Knowledge and Integration of People
2. Better Process
3. Reality-based Measurements

What companies encounter instead

1. Distracted by day-to-day delivery pressures – 78%
2. Tools don't integrate properly – 62%
3. Lack the necessary internal expertise – 56%

Source: Forrester study commissioned by Wipro, 2008

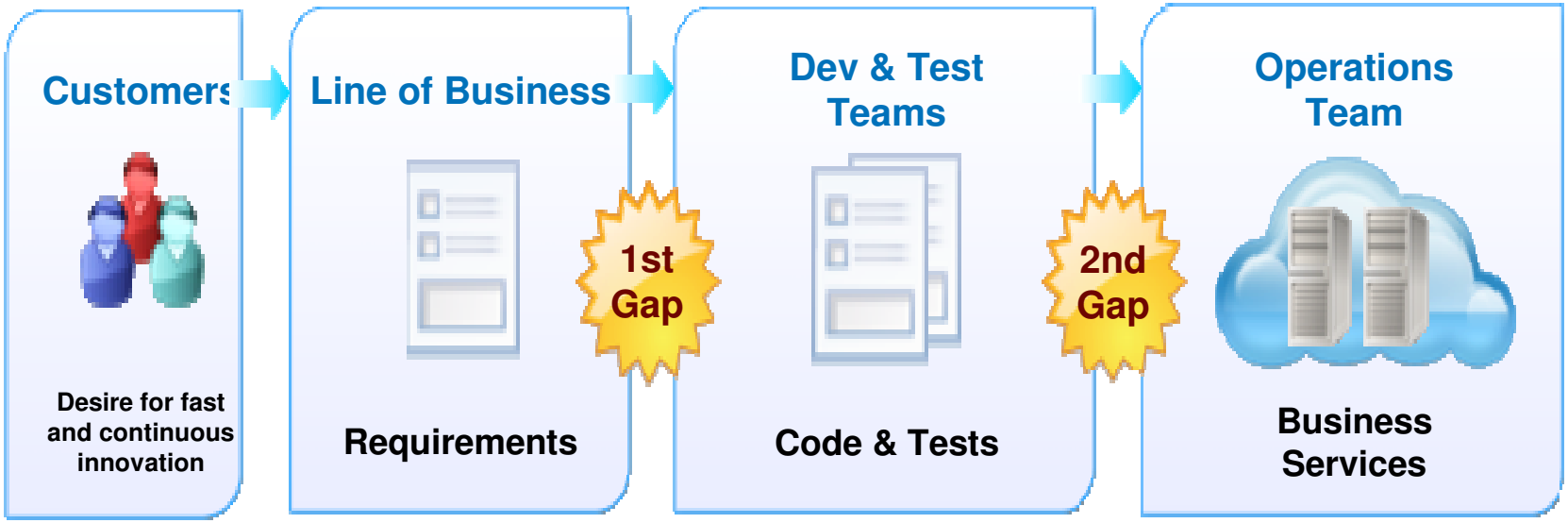
Delivery Challenges

Today's business and technical needs are pushing traditional delivery approaches to the breaking point

People

Process

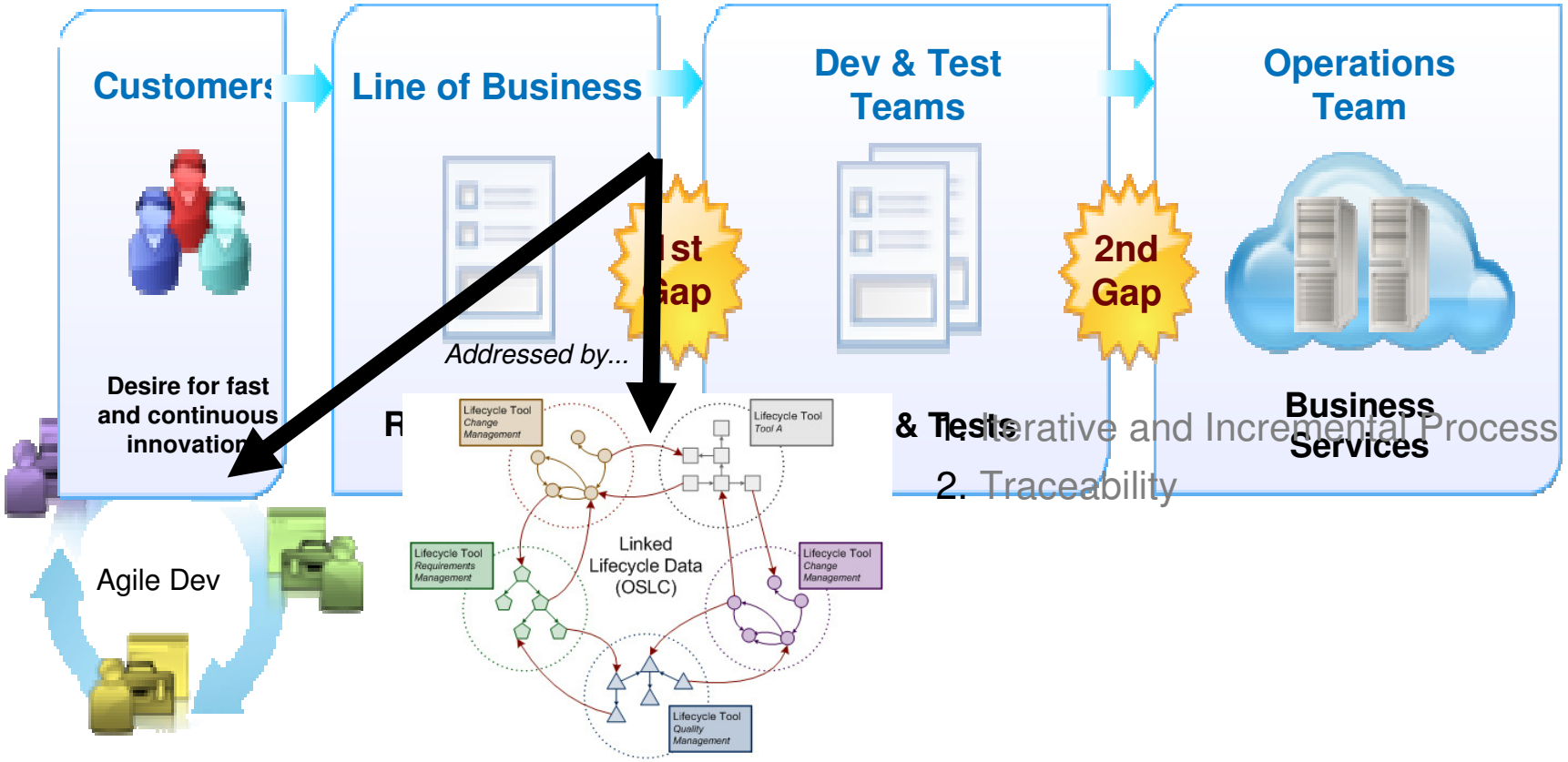
Information



"At some point, you take a step back, and you realize you have an awful lot of **siloed systems** that are **limiting transparency** across strategic projects."

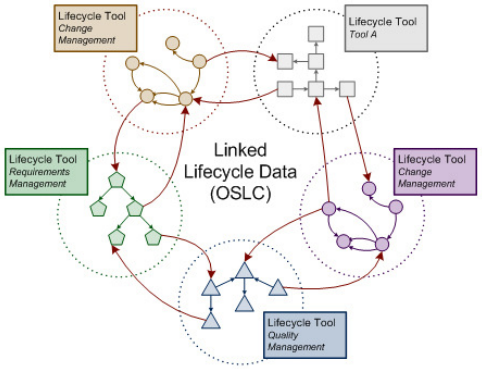
- Development Director
Temenos, Inc.

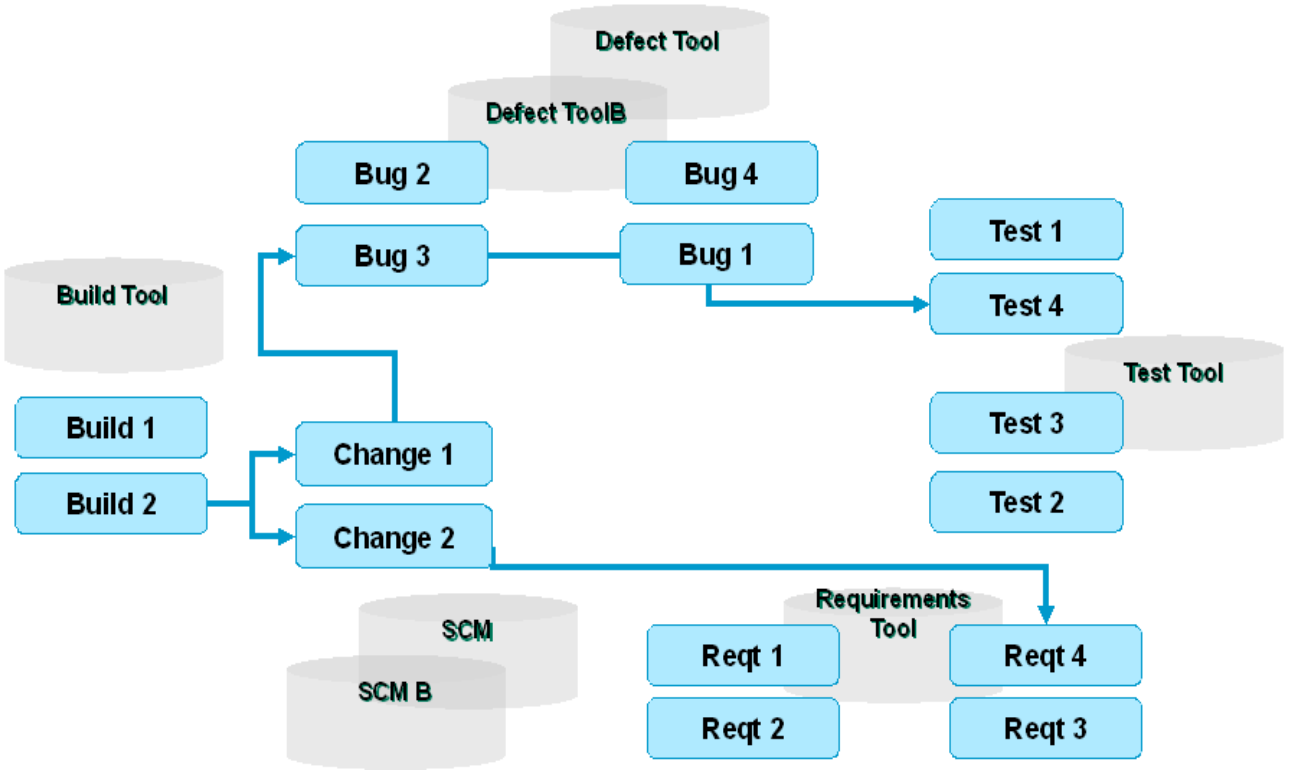
Addressing *BusDev* gaps



1. Iterative and Incremental Process
2. Traceability

Linked Lifecycle Data





- The data is the thing
  - Resources and relationships
  - Tools operate on the data
  - Tools execute the process
  - Tools expose their data in a common way (REST)
- Lifecycle integration:

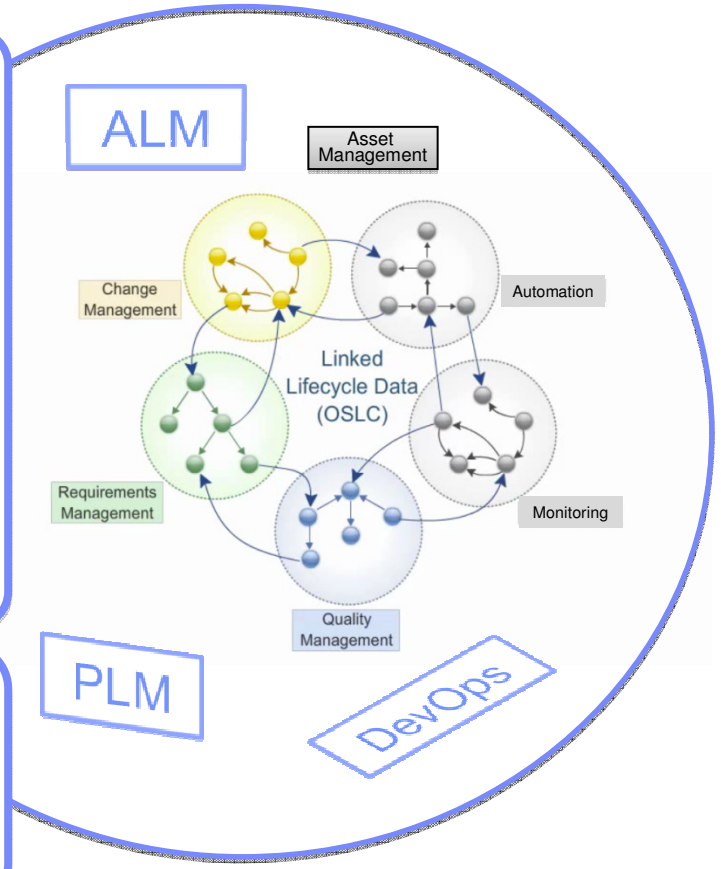
  Tracing, indexing, analyzing the web of lifecycle data where it lives
- Utilizes architecture of the internet
  - All data are resources with URLs
  - Open standards
  - Loosely coupled
  - Technology neutral
  - Scalable, extensible

Open Services for Lifecycle Collaboration (OSLC)

Working to improve the way software lifecycle tools share data

Open Services for Lifecycle Collaboration
Lifecycle integration inspired by the web

- Community driven and governed
  - 400+ registered community members
  - Workgroup members from 34+ organizations
- Wide range of interests, expertise, & participation
- Open specifications for numerous disciplines
- Defined by scenarios – solution oriented
- Implementations from IBM, BPs, and Others
- Based on W3C Linked Data



Inspired by the web **Proven**

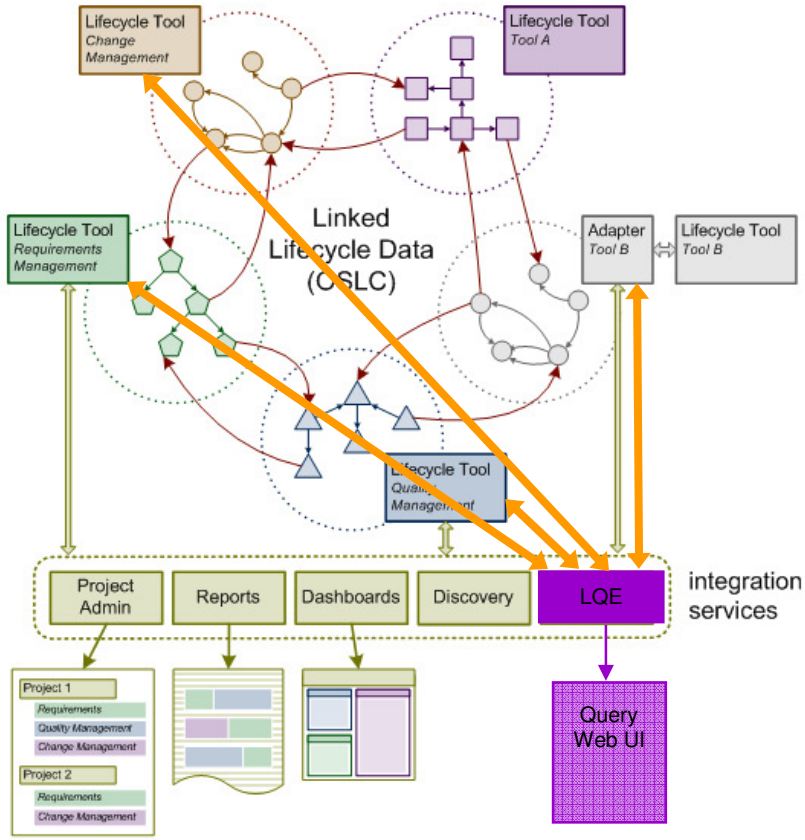
Free to use and share **Open**

Changing the industry **Innovative**

open-services.net

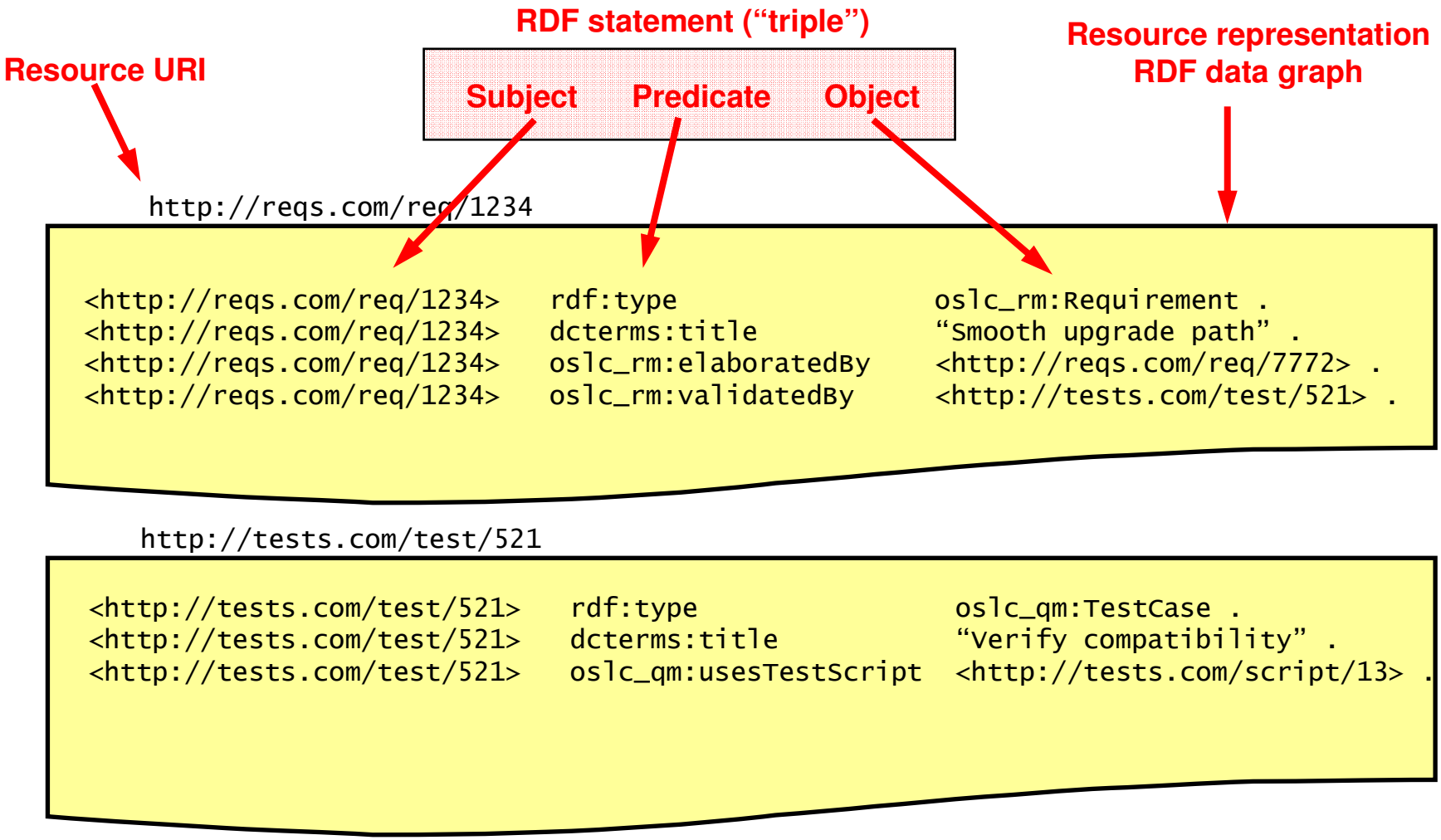
GET INVOLVED AND CONTRIBUTE!

New Integration Service - Lifecycle Query



Provides ability to run queries over linked lifecycle data aggregated from multiple lifecycle tools

Linked Lifecycle Data (LLD)



RDF standard www.w3.org/RDF/

SPARQL Query Language

- SPARQL is standard query language for RDF datasets

SPARQL query

```
SELECT ?x ?title1 ?y ?title2 WHERE {
  ?x   rdf:type            oslc_rm:Requirement .
  ?x   dcterms:title       ?title1 .
  ?x   oslc_rm:validatedBy ?y .
  ?y   rdf:type            oslc_qm:TestCase .
  ?y   dcterms:title       ?title2 . }
```

Query results

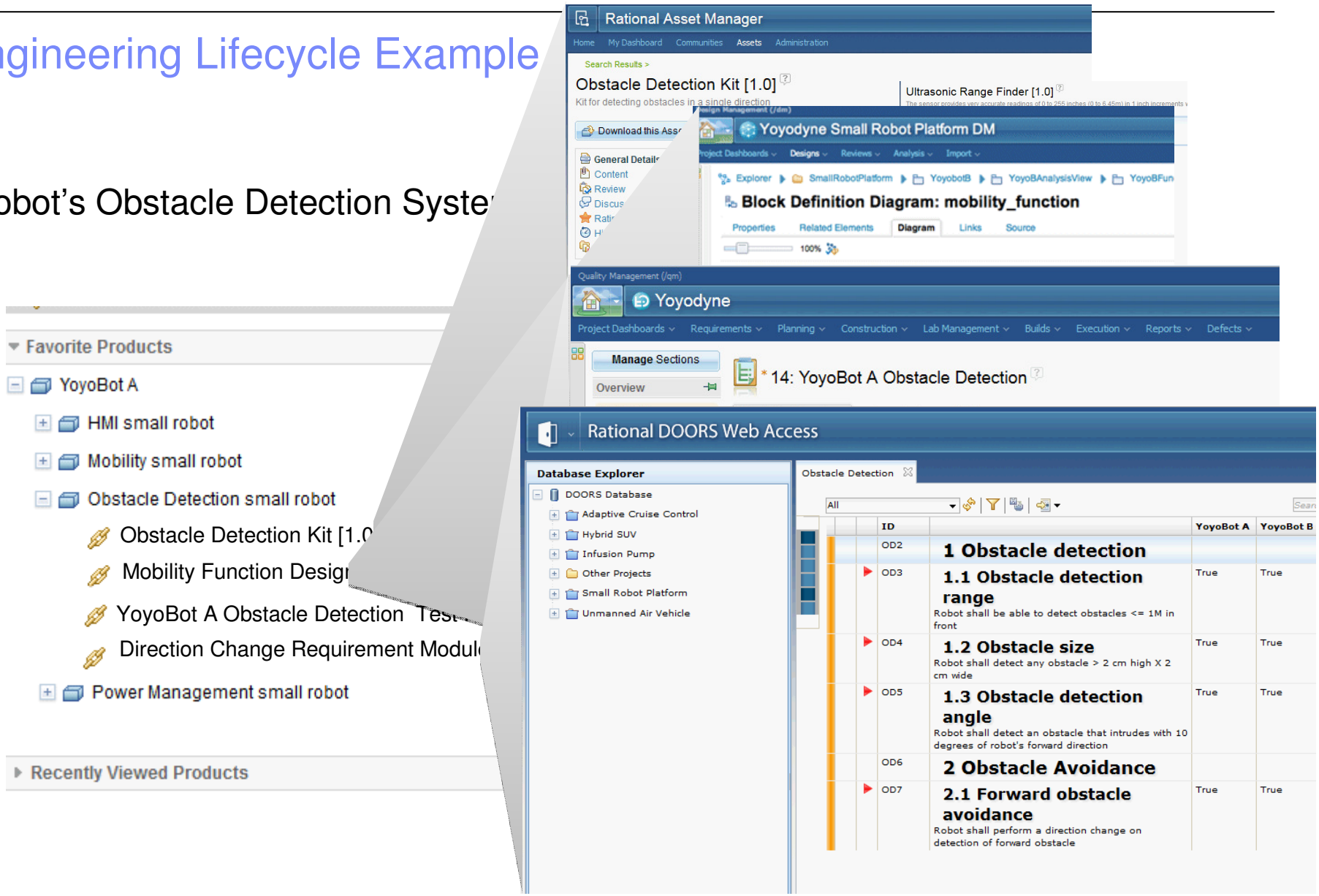
x	title1	y	title2
<http://reqs.com/req/1234>	"Smooth upgrade path"	<http://tests.com/test/521>	"Verify compatibility"

- Queries can mine linked lifecycle data aggregated from multiple lifecycle tools

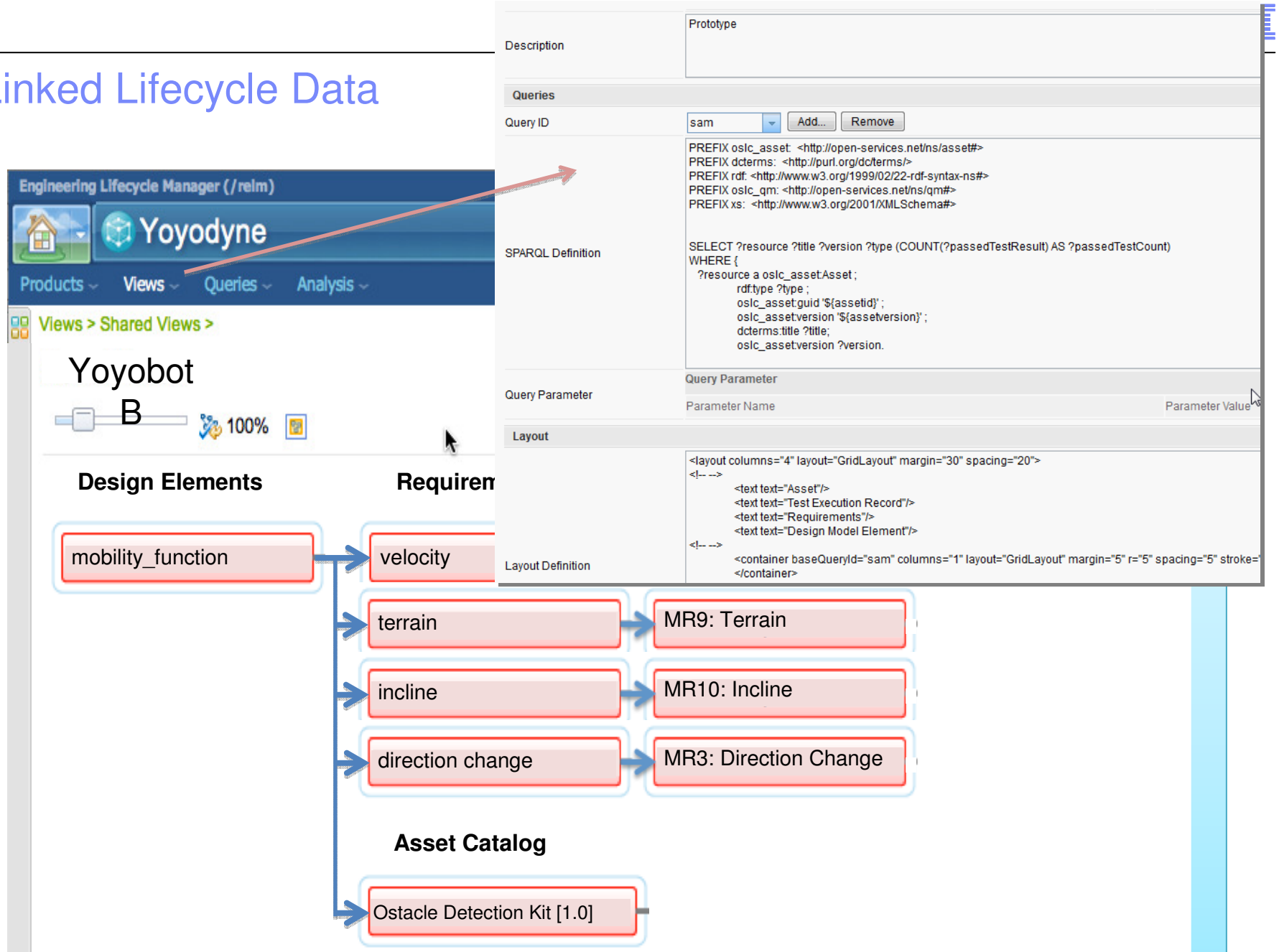
SPARQL standard www.w3.org/TR/rdf-sparql-query/

Engineering Lifecycle Example

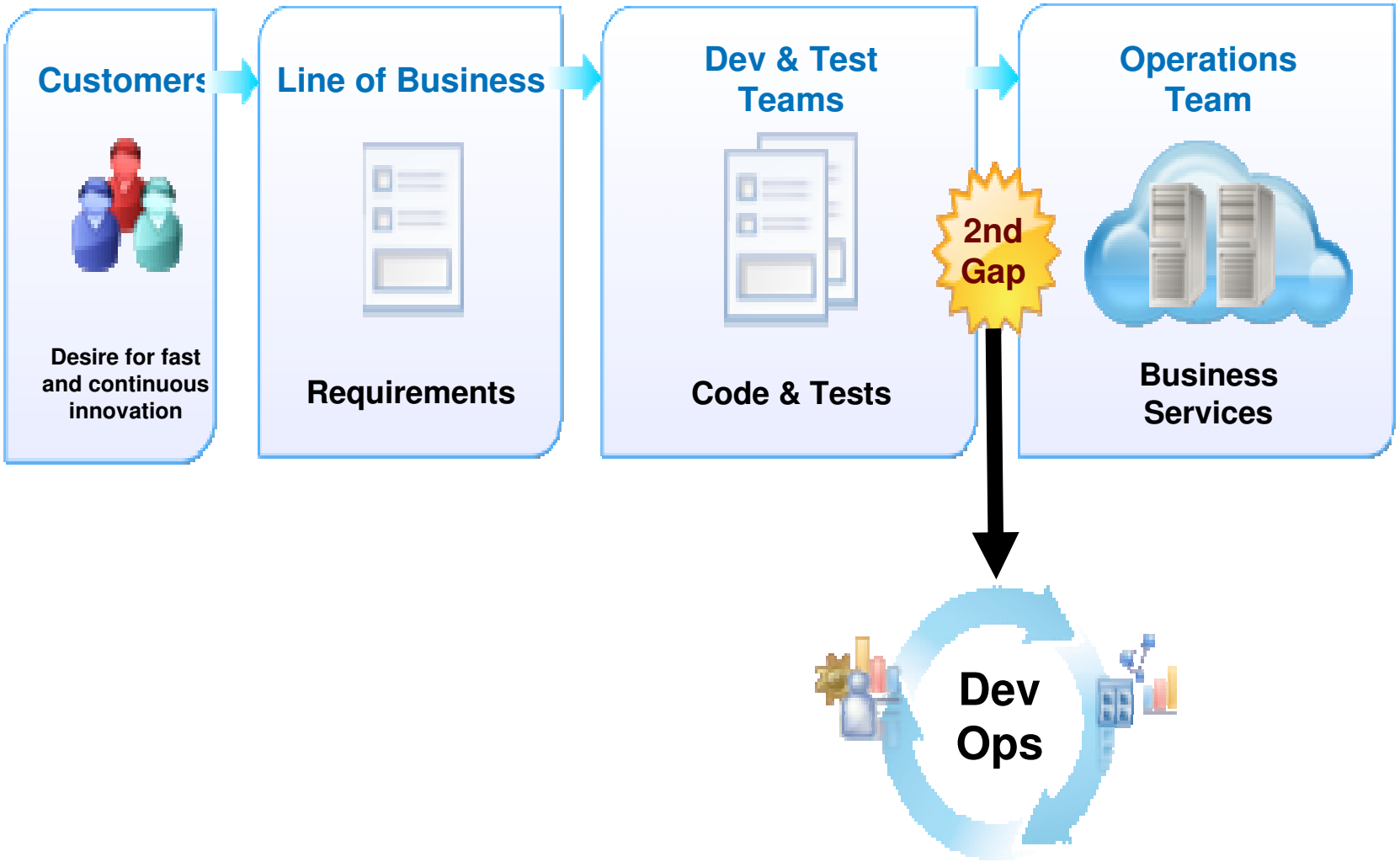
Robot's Obstacle Detection Syste



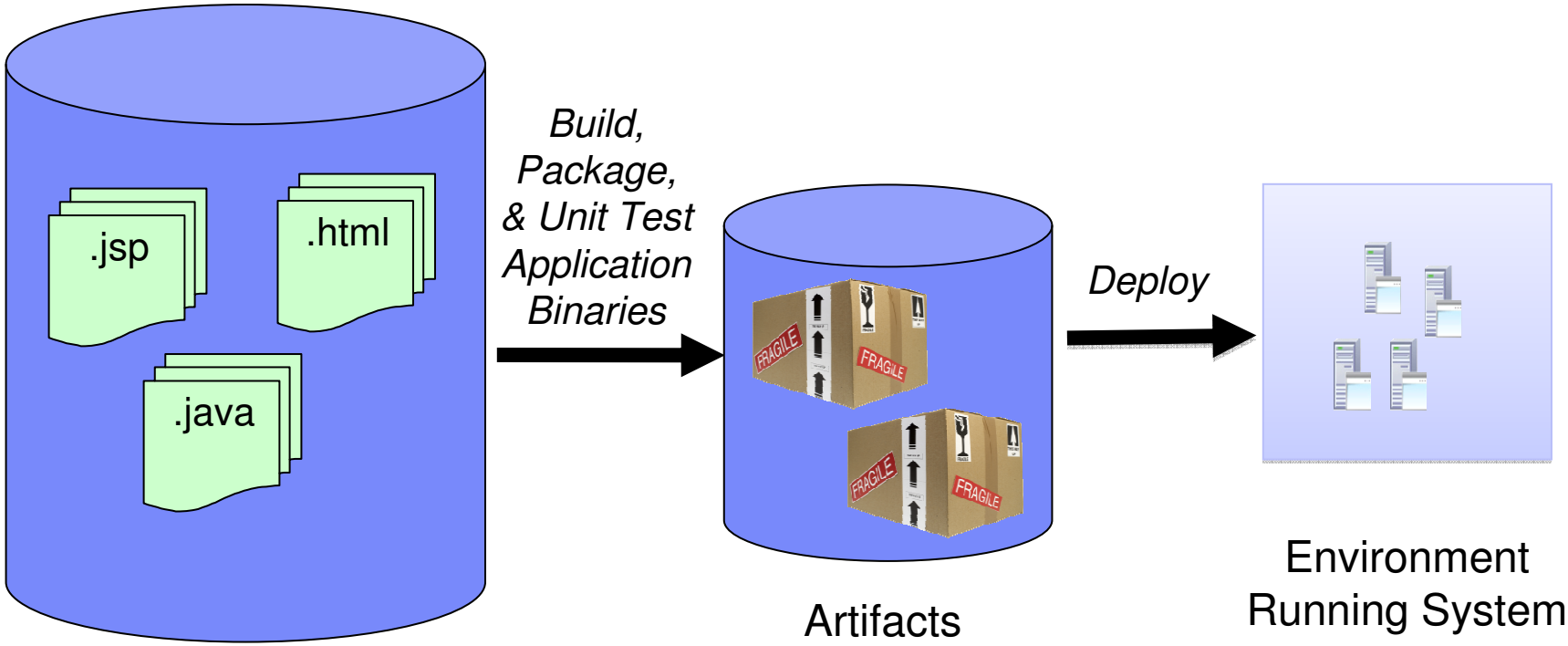
Linked Lifecycle Data



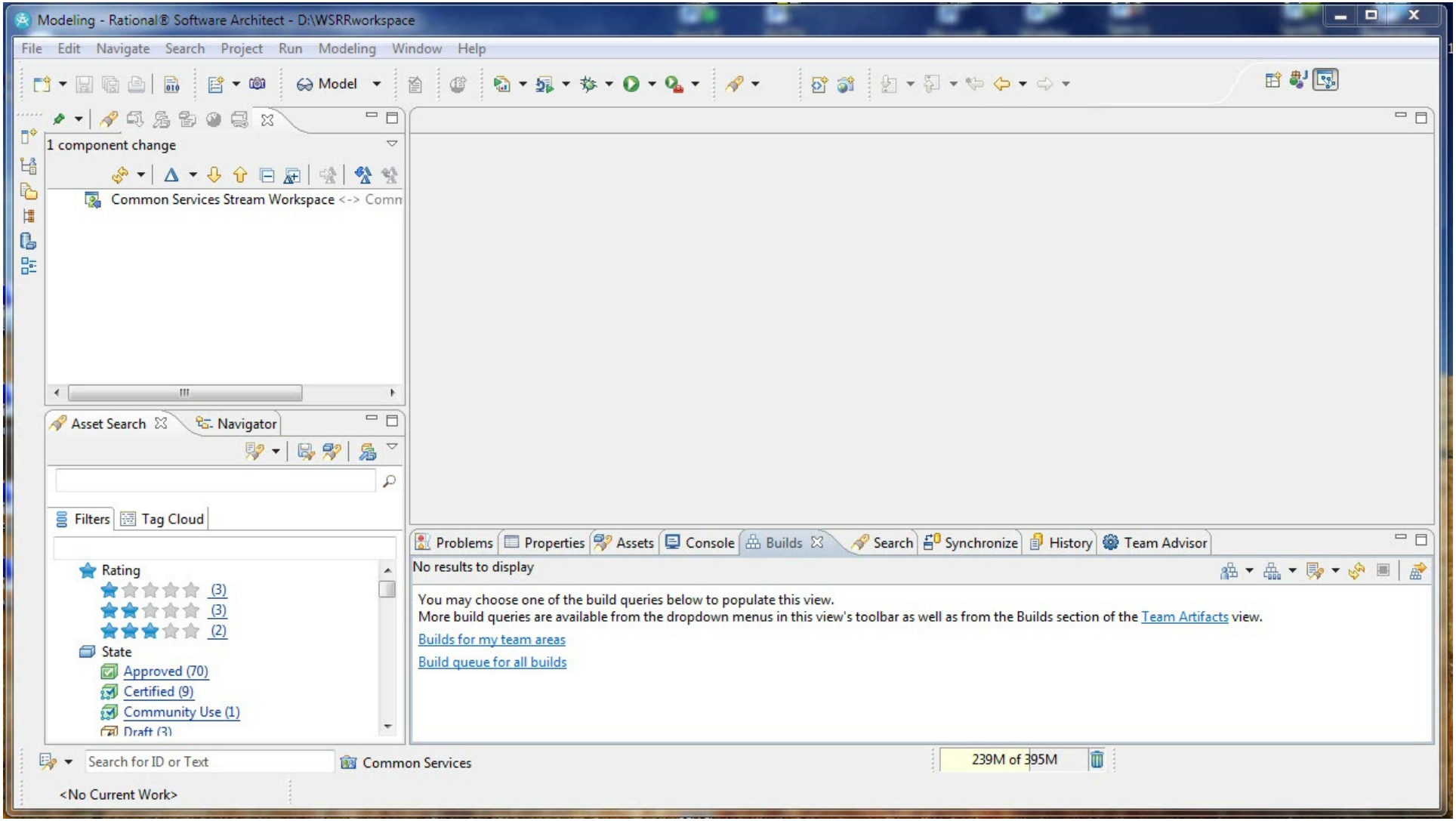
Addressing Application Lifecycle Management gaps



Automating development hand off today

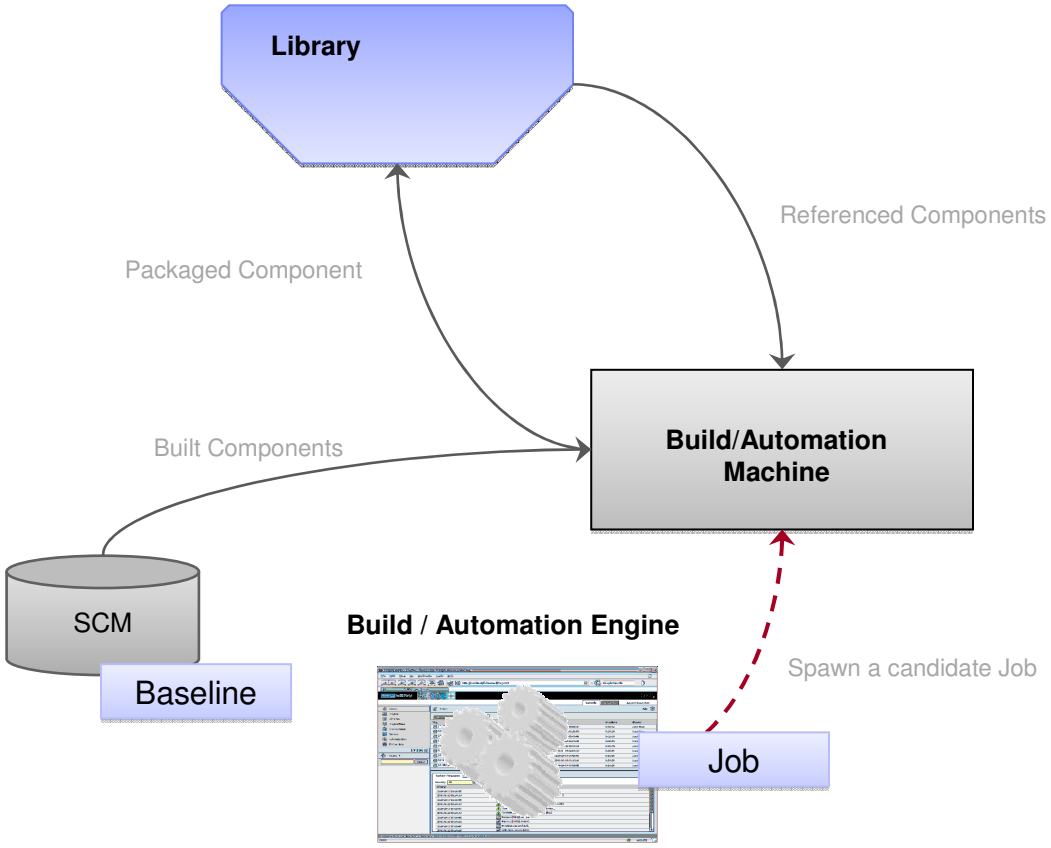


Development phase

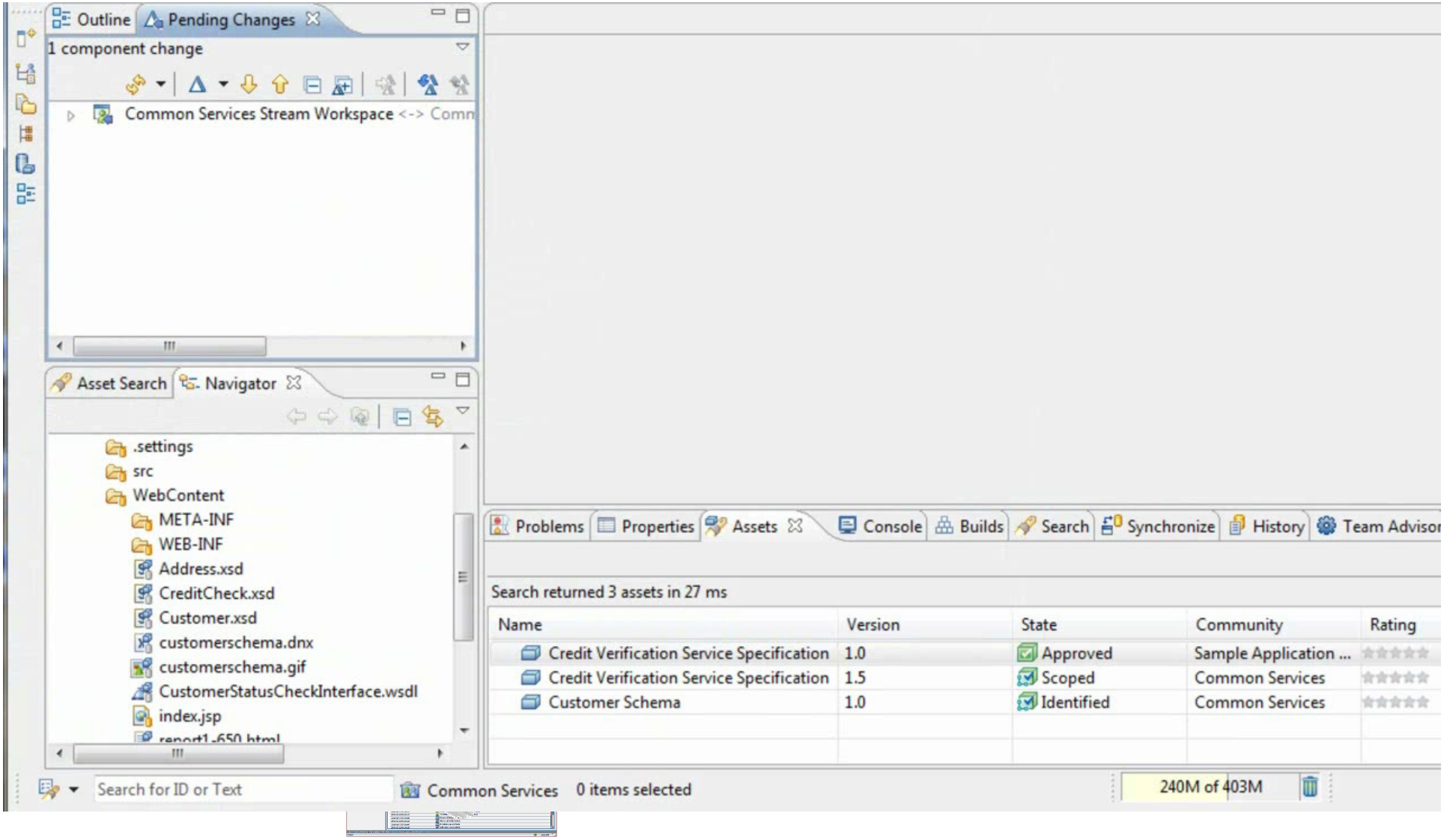


Build / Automation Phase

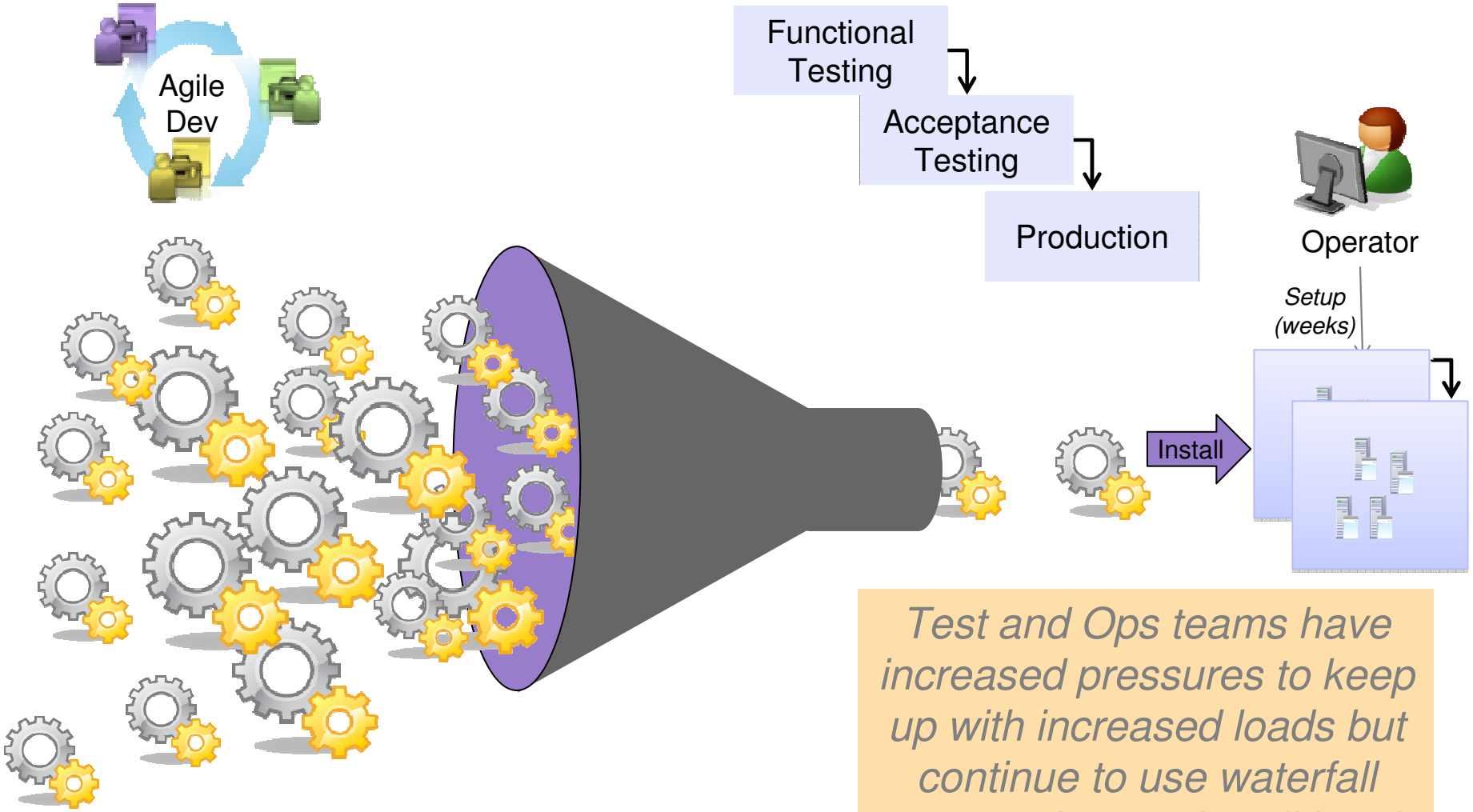
1. Track the Bill of Materials used in a build
2. Manage which build move onto the next stage



Deploy Automation Phase



With only Agile Development improvements…



CI builds are piling up

Test and Ops teams have increased pressures to keep up with increased loads but continue to use waterfall approaches and traditional tools.

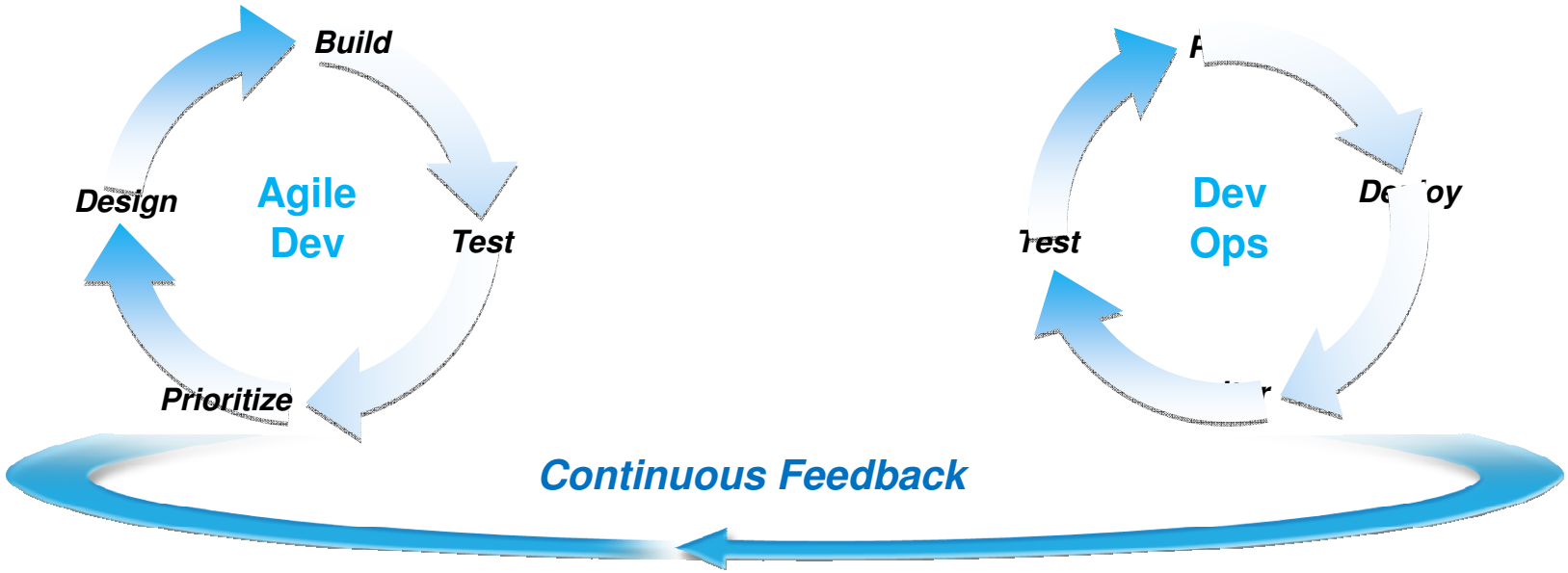
Adjusting Delivery Mindset

- Infrastructure Developer vs. Operator/Administrator
- Need to bring a software development mindset to the operational areas
- Replicate, where appropriate, standard architecture/development tools and methodologies
- Use an Agile approach to delivery of routines
- Continuous, incremental improvements and delivery of new functionality
- Automated unit and integration testing improves operational runtimes
- Source Control Management
- Automation routines and scripts are fundamental to Operations
- Managing Operations routines like source code offers several benefits:
  - Central point of truth as routines and environments change
  - Backup in case of loss
  - Identify possible regressions by comparing with prior versions

- Example Managed Assets:
- Perl, Jython, WSADMIN, ANT scripts, Service orchestration routines (opsware, buildforge, etc), Infrastructure Gold copies components

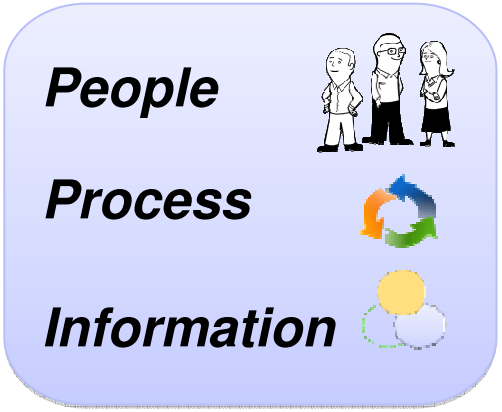
Agile Development and Delivery

Continuous Integration extends to Continuous Delivery



DevOps: Tighter alignment between Development & Operations
to increase application velocity with managed risk

DevOps Principles & Values



- Collaborate across disciplines
- Develop and test against a production-like system
- Deploy frequently
- Continuously validate operational quality characteristics

12 Principles for Better DevOps*

Collaborate

1. Do your Ops and Dev teams collaborate? Regularly?
2. Do you have agreed upon patterns for apps and platforms?
3. Do you have well defined delivery pipeline for apps and platforms?

Automate

4. Do your operation engineers understand how to developed well-structured reusable system configuration scripts?
5. Can you deploy a system in one step?
6. Do you provide Infrastructure and Platform as a Service for your development teams?
7. Can your developers launch, use, and destroy representative environments on demand without operator support?

*Based on "The Joel Test: 12 Steps to Better Code"
http://www.joelonsoftware.com/articles/fog0000000043.html

12 Principles for Better DevOps

Validate

8.Do you have automated tests to validate your application and platform function and security?

9.Do you validate platform software against expected KPIs, before deploying your application?

10.Do you deploy your applications daily and verify them?

Manage and Control

11.Do you use source control?

12.Do you have an issue tracking system for operations, linked to a bug database used for development?

Installation Instructions

RedHat Linux

1. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

2. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Apache Web Server

1. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

2. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,

3. adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem.

Python

1. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur?

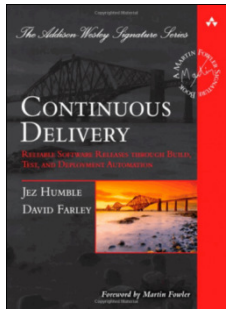
2. Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur,

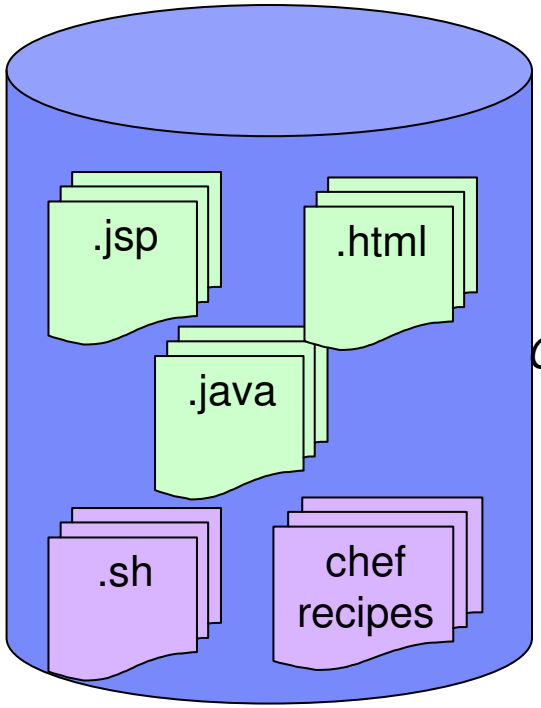
3. vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

```
#!/usr/bin/env ruby

class DevopsDeployer
  def initialize(build_url, build_id)
    @log = Logger.new(LOG_FILE)
    @log.level = LOG_LEVEL

    @iaas_gateway = IaasGateway.new(HsltProvider.new(),
LOG_FILE, LOG_LEVEL)
    @server_instance = nil

    rtc_build_system_provider = RtcBuildSystemProvider.new(
RTC_REPOSITORY_URL, RTC_USER_ID, RTC_PASSWORD_FILE)
    @build = rtc_build_system_provider.resolve_build(
build_url, ENV['buildResultUUID'], build_id)
    @build_system_gateway = BuildSystemGateway.new(
rtc_build_system_provider, LOG_FILE, LOG_LEVEL)
  end

  def add_build_stamp
    template_file = WEB_APP_ROOT +
"/app/templates/pages/page.html"
    @log.info "Adding build ID stamp #{@build.id} to \
#{template_file}"

    # Read in the file's contents as a string, replace
    # the build_id, then overwrite the original contents
    # of the file
    text = File.read(template_file)
    new_text = text.gsub(/\{\{ build_id \}\}/,
"<a href=\"#{@build.uri}\">#{@build.id}</a>")
    File.open(template_file, "w") { |file|
      file.puts new_text
    }
  end

# ...
```

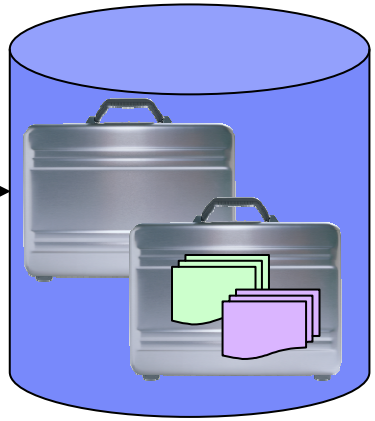
Delivery Pipeline

Using the same tools and methodologies to manage and deliver software and deployment configuration changes.

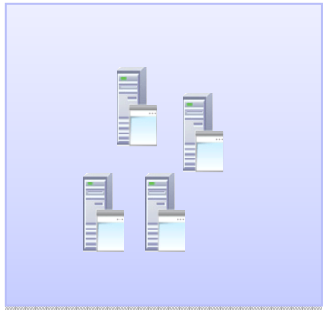




Build, Package, & Unit Test Application Binaries & Platform Configuration



Deploy



Source Artifacts

Source Control Management

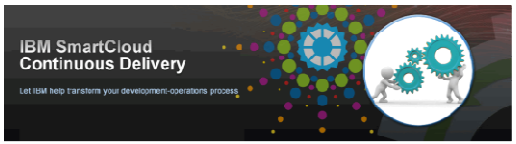
Deployable Artifacts

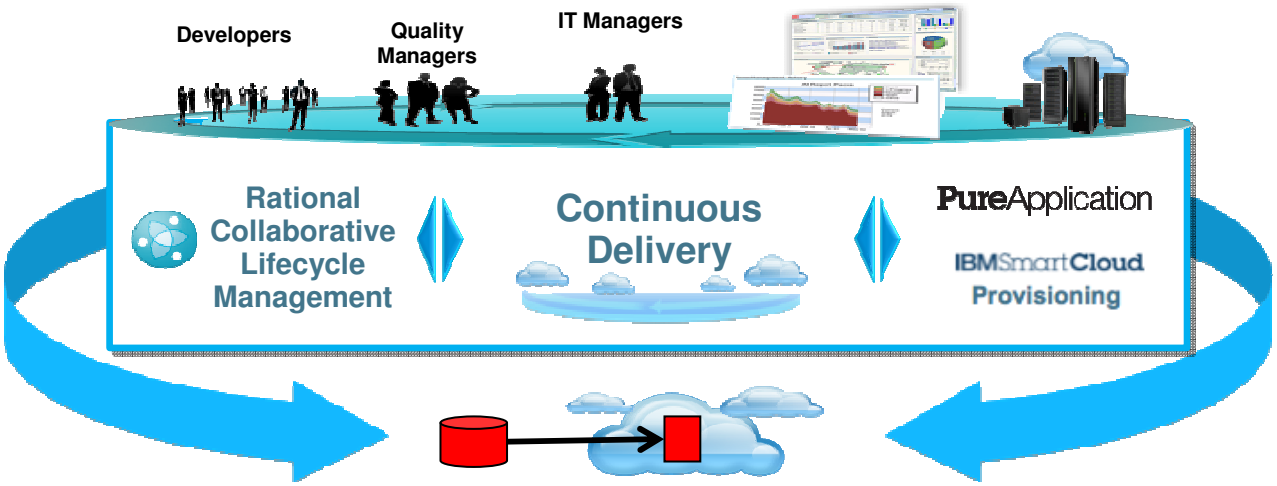
Library

Environment Running System

IBM SmartCloud Continuous Delivery

Extending Agile disciplines through delivery





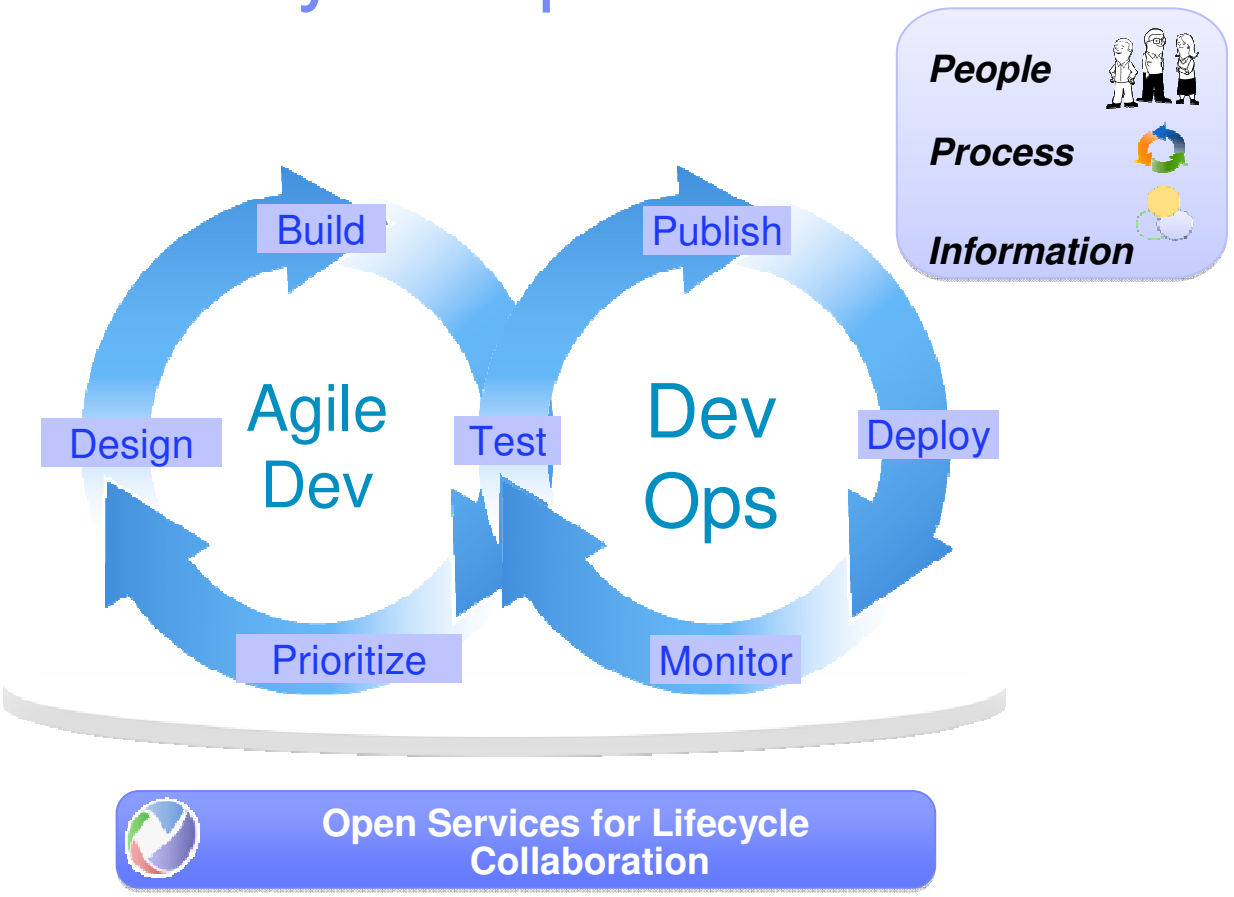
Client Value

- Reduce risk, improve quality; manage change from development to deployment
- Improve efficiency, accelerate delivery; automated handover between processes
- Optimize resources; workload pattern composition delivery

Targeted Entry

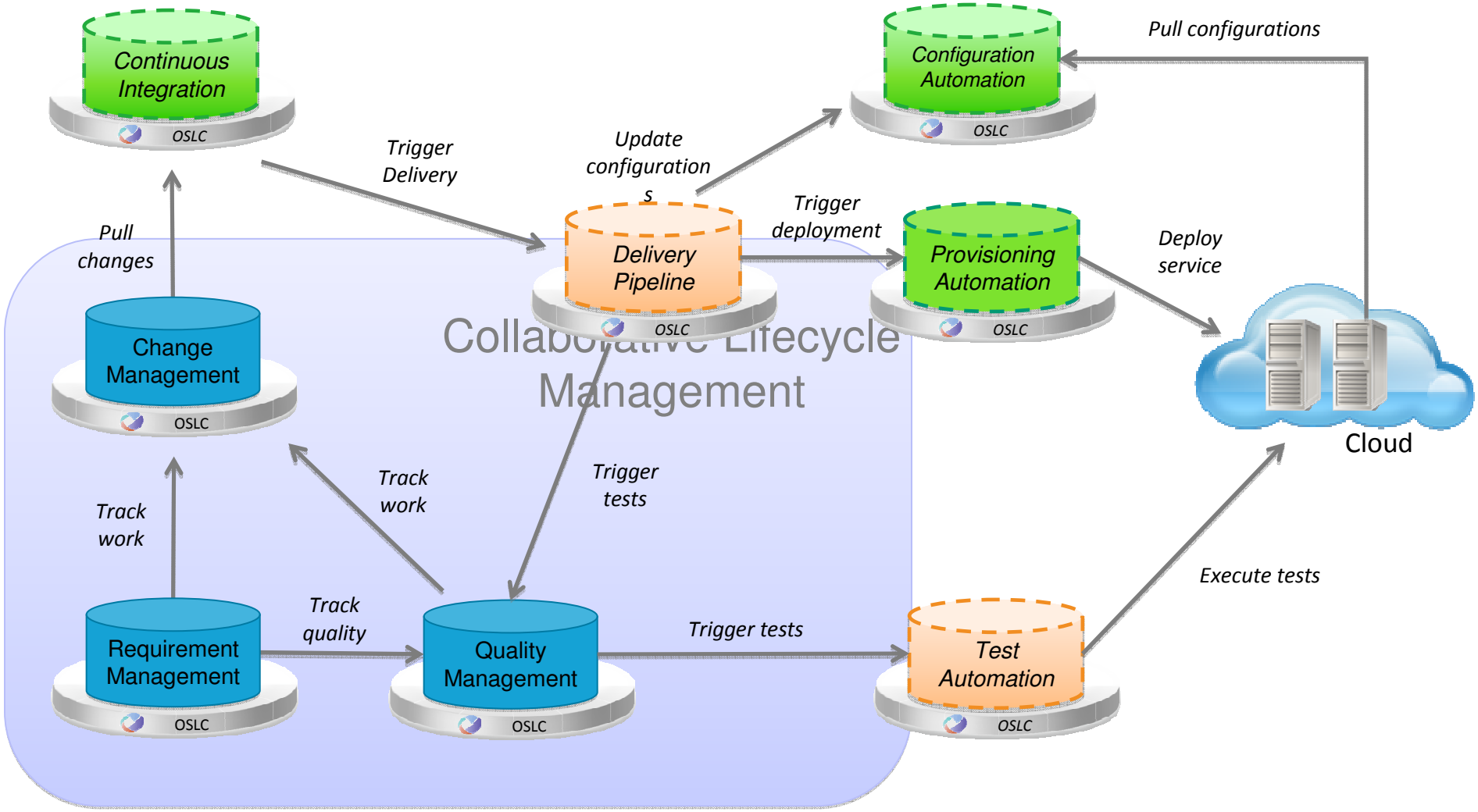
- Development team extending Agile into rapid workload deployment in the cloud
- Operation teams delivering scalable, continuous delivery services to the development organization

End-to-End Lifecycle Optimization



Follow us on the Enterprise DevOps and Jazz Team Blogs

Lifecycle Management Reference Architecture



Adopt in any order, at any time