

Taking Small Bites to Deliver the Big Picture Joshua Brusse

© Copyright 2012 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice.

What Bothers Operations The Most?

Problems in production:

- Known Errors
- Problems
- Outages
- Performance Degradation
- Installation Errors
- An unexpected impact of a change on other business services
- Knowledge not being transferred, just "Stuff coming over the wall"
- lack of monitoring

A lack of confidence that "it will be alright on the night"



Key Cause of this "Discomfort"

The Quality Of Change That Is Deployed Into Production



The Key Symptoms of Poor Quality



Code from external (and internal) developers, all too often, **IS** poor



Defects **ARE** Delivered into Production



Release Schedules **ARE** impacted, Releases **ARE** late



Projects **DO** Run Late



Lack of visibility **IS** driving lack of communication based **ON** facts



Testing **IS** a Bottleneck, **IS** Expensive & Takes **TOO** long



A Traditional AppsDev Solution

Spend time & money "fixing" Testing...

This has been tried time and time again: QA Team, Q&T, TCoE, Test Practice etc

.... is not the only Answer

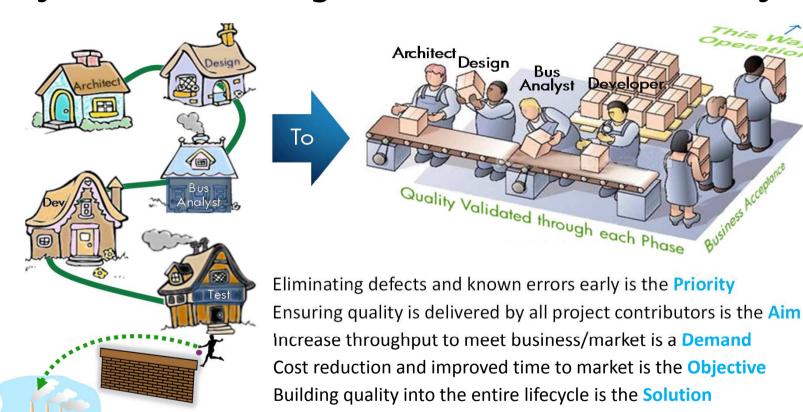
Because too many problems still hit production...

...and Ops will take care of them!!





It isn't just about Testing - it's about the entire lifecycle



Reduced risk and increased confidence in production is the Output

relopment Company, L.P. The information contained herein is subject to change without notice.

The Right Answer

Identify &
Address
Problems
early by
introducing
Quality into
the Entire
Lifecycle



Give Applications Development Teams Accountability for Quality Reject the "Ops will handle it" mantra



Give Change Managers Visibility of progress based on Quality as well as Time and Budget



Give Release & Deployment Managers Fact based Information to determine Readiness to Deploy



Use SLAs based on Metrics to Monitor SLA achievement and proactively support Services



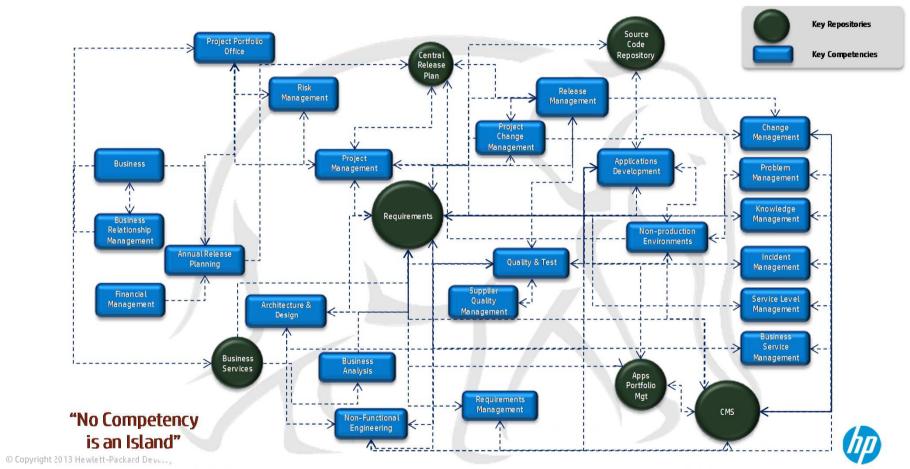
Dev & Ops – What glues them together?

AppsDev Operations Defined by Requirements Run on Servers -**Design Solution** Connected by Networks -**Build Code** End User Experience -**Provide Business Functionality** Supported by Operating Systems – **Tested for Functionality** Source of Problems -Services **Tested for Performance** Services Performance Monitored -**Defects Resolved** Supplied by Apps Dev - Fact Based Dev & Ops Sign-off Supported by Operations – Delivered to Operations Operations Early View -**Federated Data & Tooling** Tracks the overall performance of applications Visibility of the build/supply/run lifecycle based on requirements Services (supported by 1 or more Apps) the common "denominator" • Informs Operations of:

- ✓ Fact based proof of readiness to deploy
- ✓ Source of problems
- √ Impact analysis of Change/Incident
- ✓ Pipeline of work for resource & infrastructure planning



Enterprise Release Management-Reference View



Key Repositories

Requirements

Key source of referential information about the applications required to support business services.

Links to repositories that describe aspects of each requirement:

- **Functional**
- **Technical**
- Infrastructure
- Code
- Design
- **Business Processes**
- Tests
- **Defects, Problems & Incidents**
- **Business Processes & Services**

Business

Central Release Plan

All scheduled changes to production managed using the CRP.

Requirements are the basis of measuring progress and assess readiness for deployment.

Source Code Repository

Source code should be consolidated to enterprise rated and established Source Code Management (SCM) systems.

Source code to be linked to relevant requirements

Applications Portfolio

Key bridge between applications and the operations infrastructure repository,

CMS

CMS represents the sustained and current view of the configuration of all significant components of the IT environment.

CMS



Source

Code Repository





Business Services defined and stored in an originating repository.

Requirements

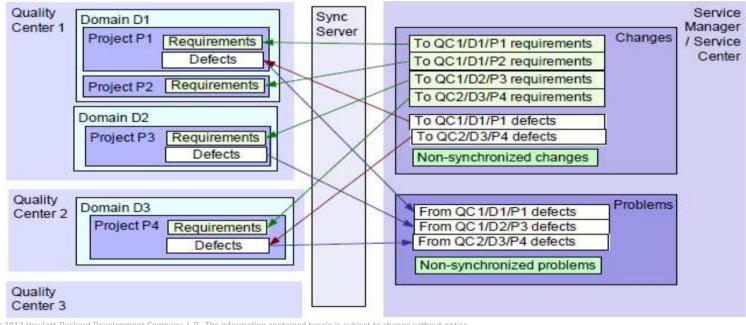
Linked to the central requirements repository

An Example of Federating DevOps Data

Using the ALM/QC synchronizer for Service Manager

ALM and SM makes the linkage of the following entities possible:

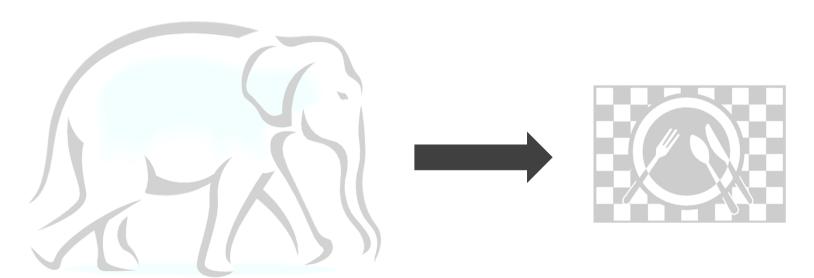
- SM Change => ALM Requirement or Defect
- SM Problem <=> ALM Defect





Having Determined that Change is Required,

How do we make it happen?



Having designed the elephant -

How do we eat it?

Answer: One Bite at a time



Requirements Management – A Place to Start



After elicitation and design. BAs have visibility of testing and quality outcomes

Applications Development

Code coverage to meet requirements verified for quality

Quality and Test

Test Execution coverage measured by coverage of requirements

Change Management

Changes are monitored and tracked to ensure that projects are to delivered to satisfy requirements

Project Managers

Project Managers to have "hard facts" to support schedule, budget and quality objectives based on progress against requirements.

Projects and other changes

PPO to have portfolio and enterprise view of metrics. SLAs and progress

and Portfolio Office

Project

SERVICES DEFINED BY REQUIREMENTS

- Services and tooling to support the development and management of requirements for the enterprise
- Provision of traceability of requirements to business processes
- Ensure that linkages and sharing with other competencies is sustained via tooling and information from other repositories (such as SCM, CMS, project portfolios etc).
- Provision of metrics and reporting for enterprise level visibility of requirements creation, development and delivery
- Provision of reporting that facilitates visibility for the impacted competencies

Architecture and Solutions

Impact analysis upon which Solutions Design is drawn from requirements based assessments of change

Release Management

Confirm readiness to deploy based on requirements coverage and completion

Non Production Environments

Environment Requirements driven through requirements management that is informed by Ops to provision Fit for Purpose Dev/Test Environments

Operations

Requirements Management informed by repositories such as CMS. Confirm change is ready for deployment with minimal risk



Key Steps – A bite at a time?

- **Define, Plan and Manage a Transformation Programme**
- **Maturing Foundational Repositories**
 - CMS (automated discovery, dependency mapping leading to service models)
 - Requirements CMS & Requirements will mature together
- Uplift Key Competencies to utilise CMS and Requirements e.g.:
 - **Estimations & Resource Management**
 - **Business Analysis & Technology Architects**
 - **Build/Test & Release Management**
 - Implementation & Transition to Production
- **Mature Additional Repositories**
 - **Business Services**
 - **Applications Portfolio Management**
 - Central Release Plan
 - **Source Code Repositories**
 - · Quality & Test, Project Portfolio, Resource etc
- Establish meaningful, sustainable and relevant metrics and report on them
- **Manage for Continual Service Improvement**



Outcomes of an Enterprise Release Management Strategy



Reduce Costs



Reduce Risk



Improve time to market



Change Impact Analysis – Getting the business what it wants and giving operations what it expects



Release Management **Confidence** – **Hitting the Mark**



Meaningful and relevant information at anytime - from a single factual source



"Right first time" deployments -Fliminate failure



Early warning of Success (or Threats) – "Know Early, Know Often"



A call to Action

Monday Morning

Identify the problems you get in production that are due to applications problems delivered by Apps Dev Calculate the cost to your organisation of what it takes to fix these problems (either by direct intervention, emergency fixes/releases or change requests for future releases)

Go ask the Apps Development Manager what he/she is going to do about it

Next 90 Days

Formalise the "Monday Morning" findings and discussion Develop a roadmap that addresses the Quality issues Identify your key repositories and rate them based on their effectiveness Develop an "uplift plan" with solid ROI outcomes for the business case

Next Year

Implement the CMS & Requirements repositories and Metrics & Reporting Enable the entire AppsDev organisation to ensure that all of them are accountable for quality

