



Getting Enterprise Release Management Right

Taking Small Bites to Deliver the Big Picture
Joshua Brusse

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

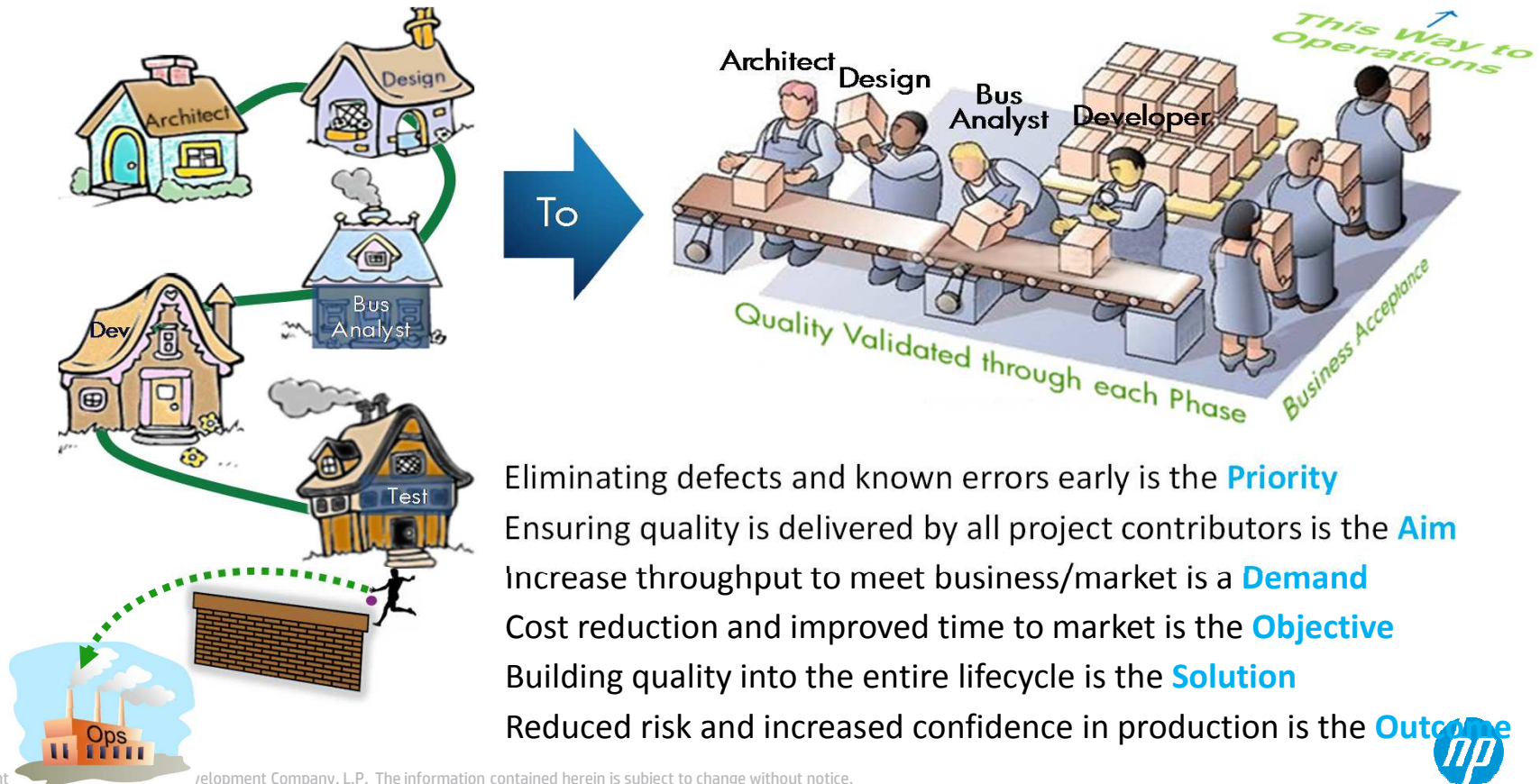
..... is not the only Answer

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



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



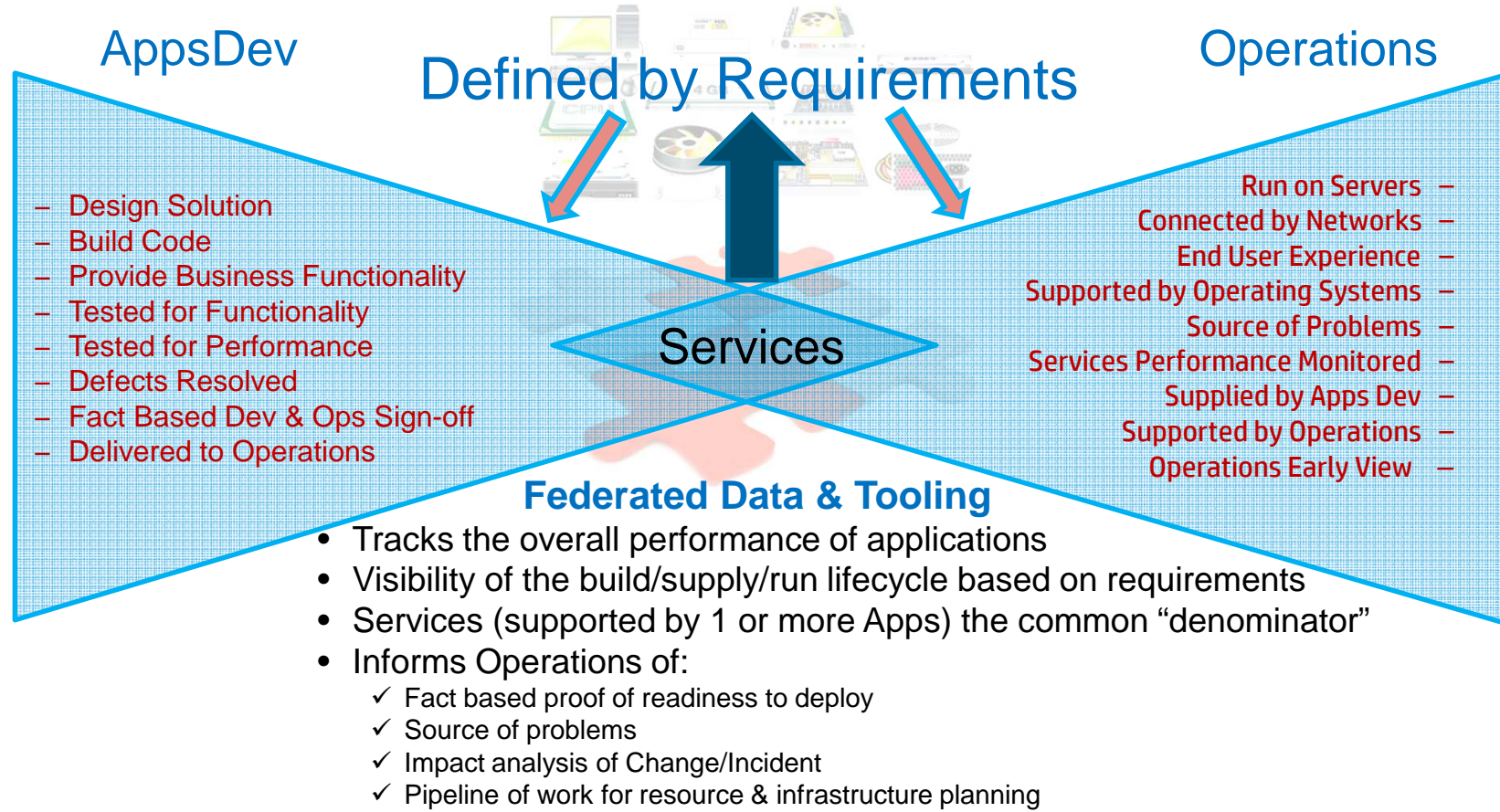
Give Release & Deployment Managers Fact based Information to determine Readiness to Deploy



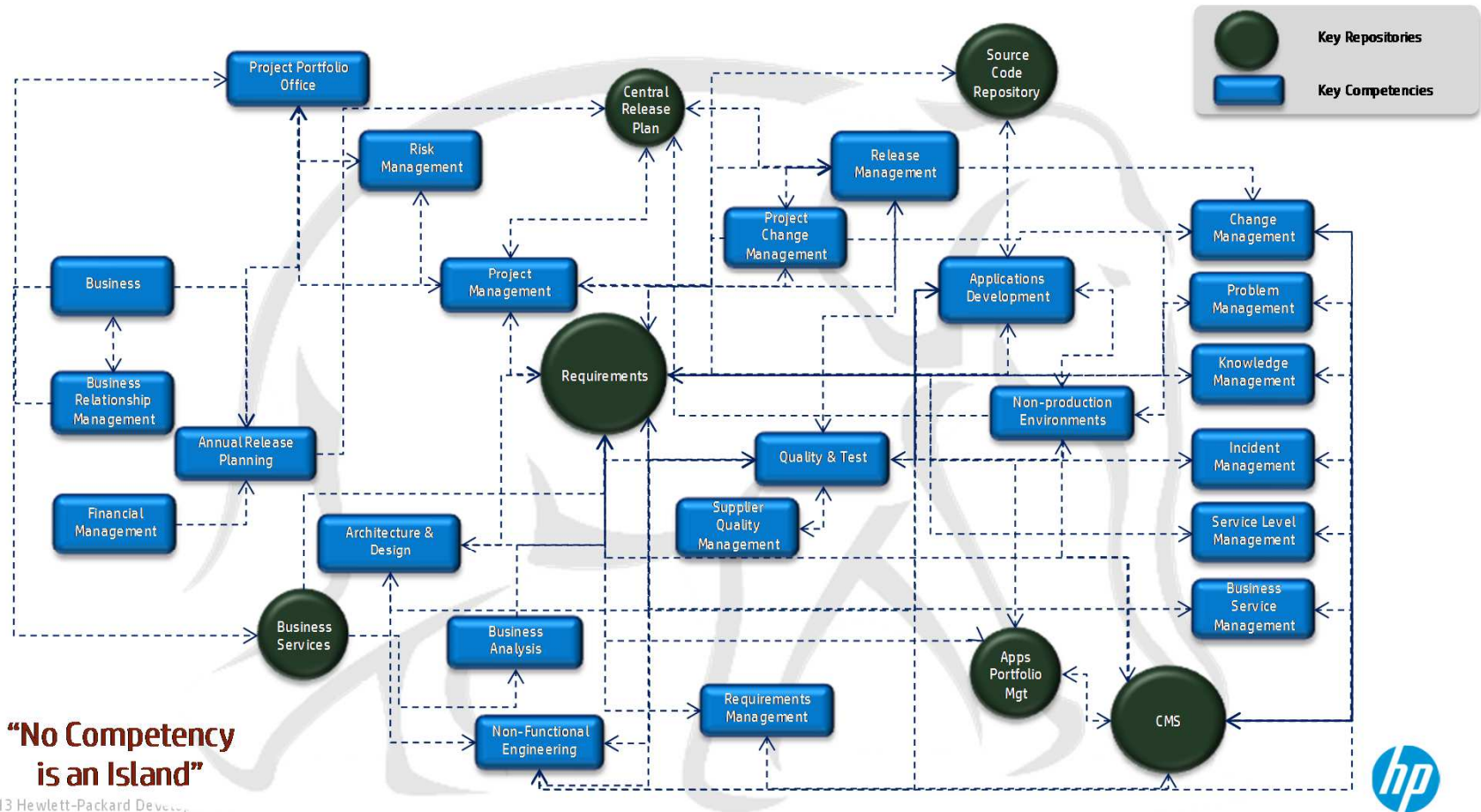
Use SLAs based on Metrics to Monitor SLA achievement and proactively support Services



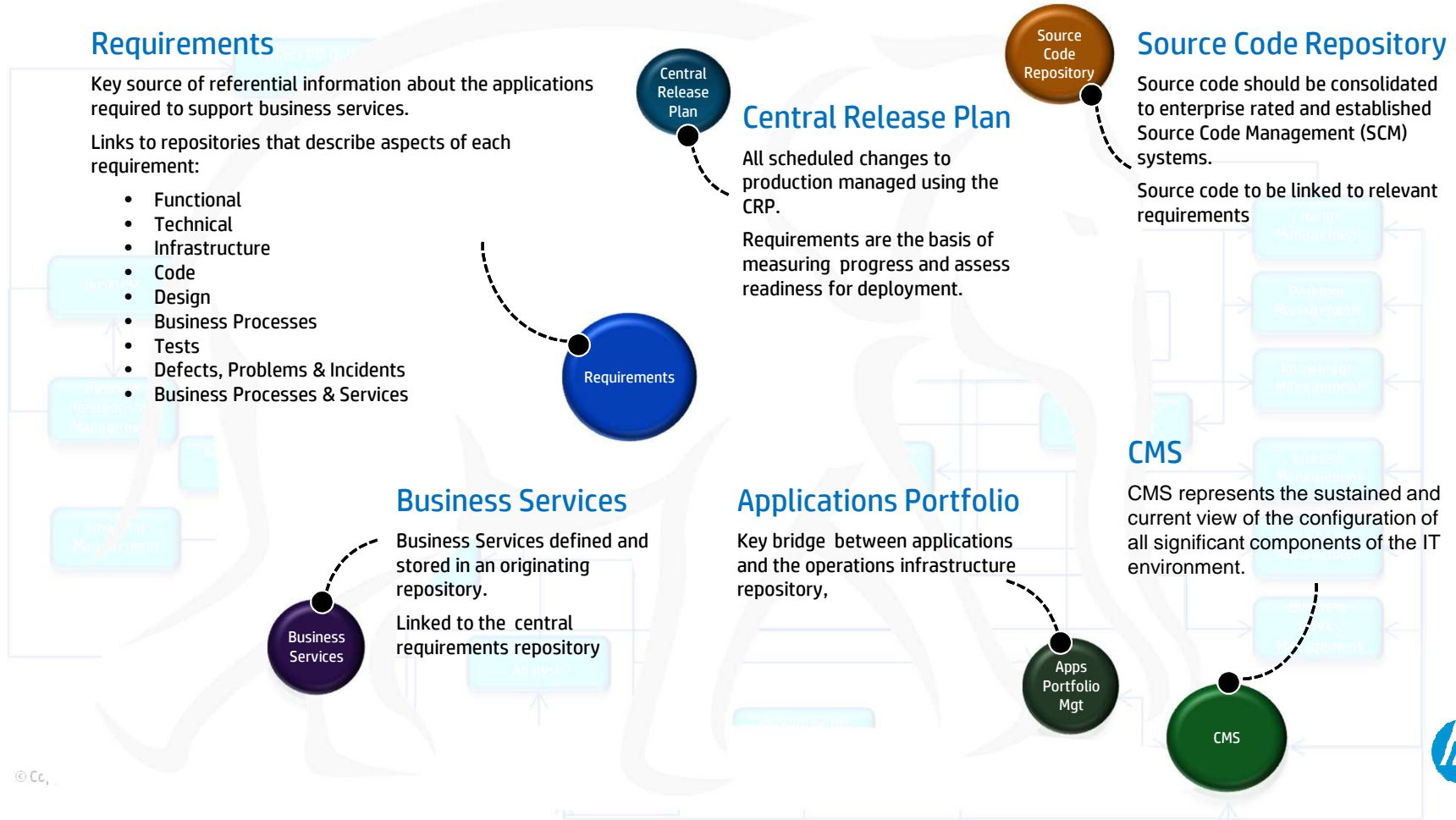
Dev & Ops – What glues them together?



Enterprise Release Management– Reference View



Key Repositories

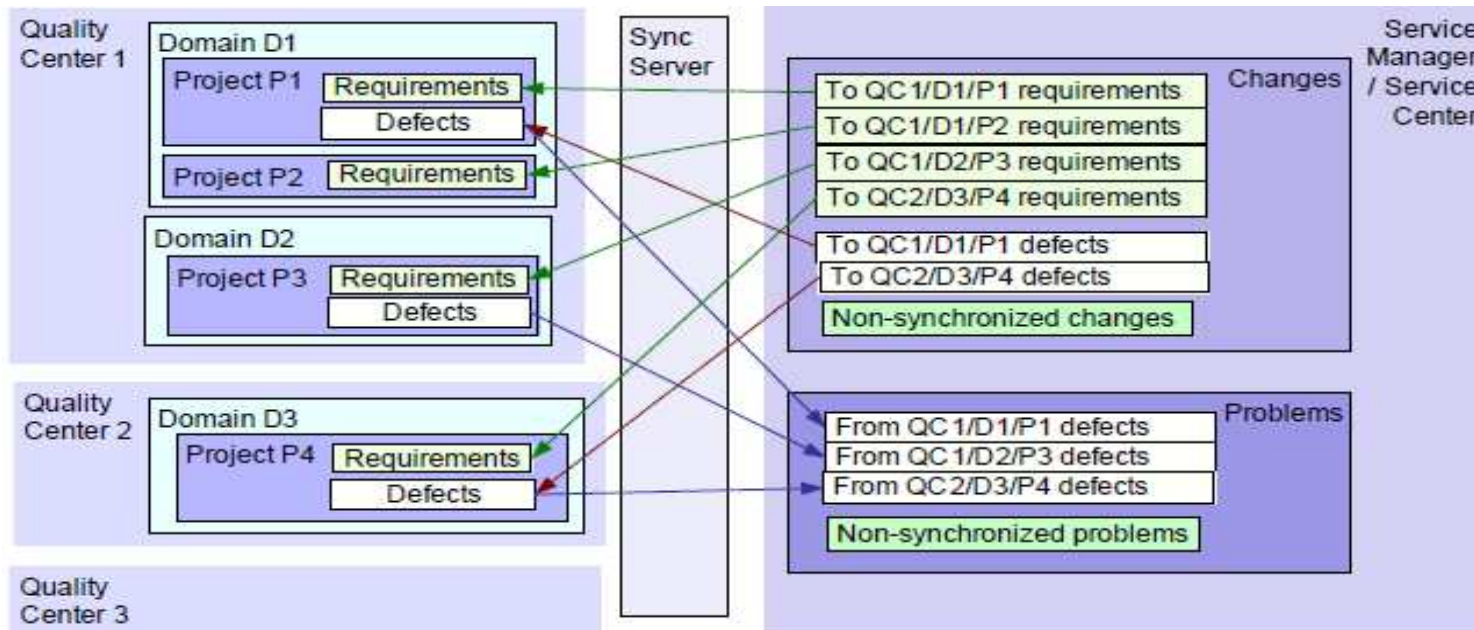


An Example of Federating DevOps Data

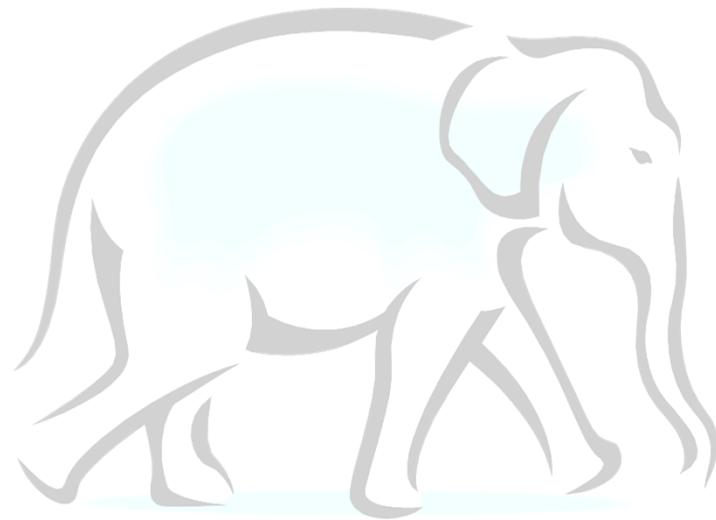
Using the ALM/QC synchronizer for Service Manager

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?



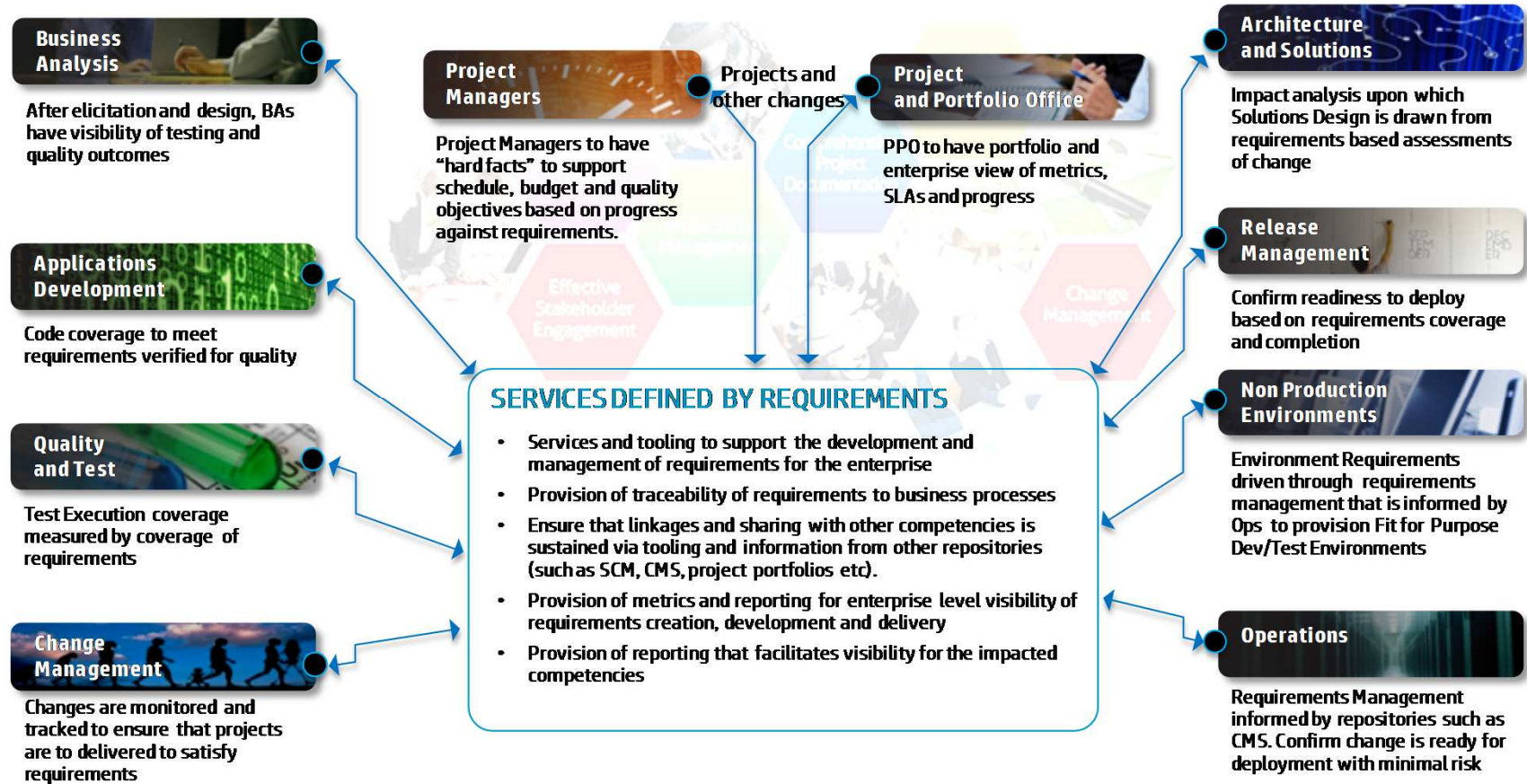
Having designed the elephant -

Answer: One Bite at a time

How do we eat it?



Requirements Management – A Place to Start



Key Steps – A bite at a time?

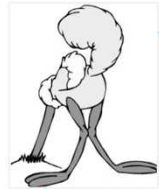
- 0 **Define, Plan and Manage a Transformation Programme**
- 1 **Maturing Foundational Repositories**
 - CMS (automated discovery, dependency mapping leading to service models)
 - Requirements – CMS & Requirements will mature together
- 2 **Uplift Key Competencies to utilise CMS and Requirements e.g.:**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

