

Правильный способ внедрения управления релизами в масштабах предприятия: как сложить большую картину по частям

Джошуа Брюссе
Старший архитектор, HP Software





Что больше всего напрягает эксплуатационные подразделения?

- Проблемы в продуктивных средах:
 - Известные ошибки
 - Проблемы
 - Простои
 - Деградация производительности
 - Ошибки при инсталляции
 - Неожиданное воздействие изменения на другие бизнес-услуги
 - Отсутствие передачи знаний, вместо этого «они просто перебрасывают штуки через стену»
 - Недостаточность мониторинга
- Недостаток уверенности, что «сегодня за ночь ничего не случится»



Причина этого «дискомфорта»

*Качество изменений,
которые развёртываются
в продуктивной среде*



СЕРВИСНОЕ МНОГООБОРЬЕ
ITSM: от фитнеса – к спорту высоких достижений



Основные симптомы низкого качества



Код от внешних
(и внутренних)
разработчиков слишком
часто **ПЛОХОЙ**



Дефекты
РАЗВОРАЧИВАЮТСЯ
в продуктиве



Графики релизов
МЕНЯЮТСЯ,
релизы **ЗАПАЗДЫВАЮТ**



Проекты завершаются
С ОПОЗДАНИЕМ



Недостаточная прозрачность
ВЕДЁТ
к недоставу коммуникаций
на базе **ФАКТОВ**



Тестирование **ЯВЛЯЕТСЯ**
бутылочным горлышком,
оно **ДОРОГОЕ**
и **СЛИШКОМ** долгое



Традиционное решение разработчиков

Потратить время и деньги на «правильное» тестирование

Это делали много раз: команда QA, Q&T, TCoE, практики тестирования и т.д.

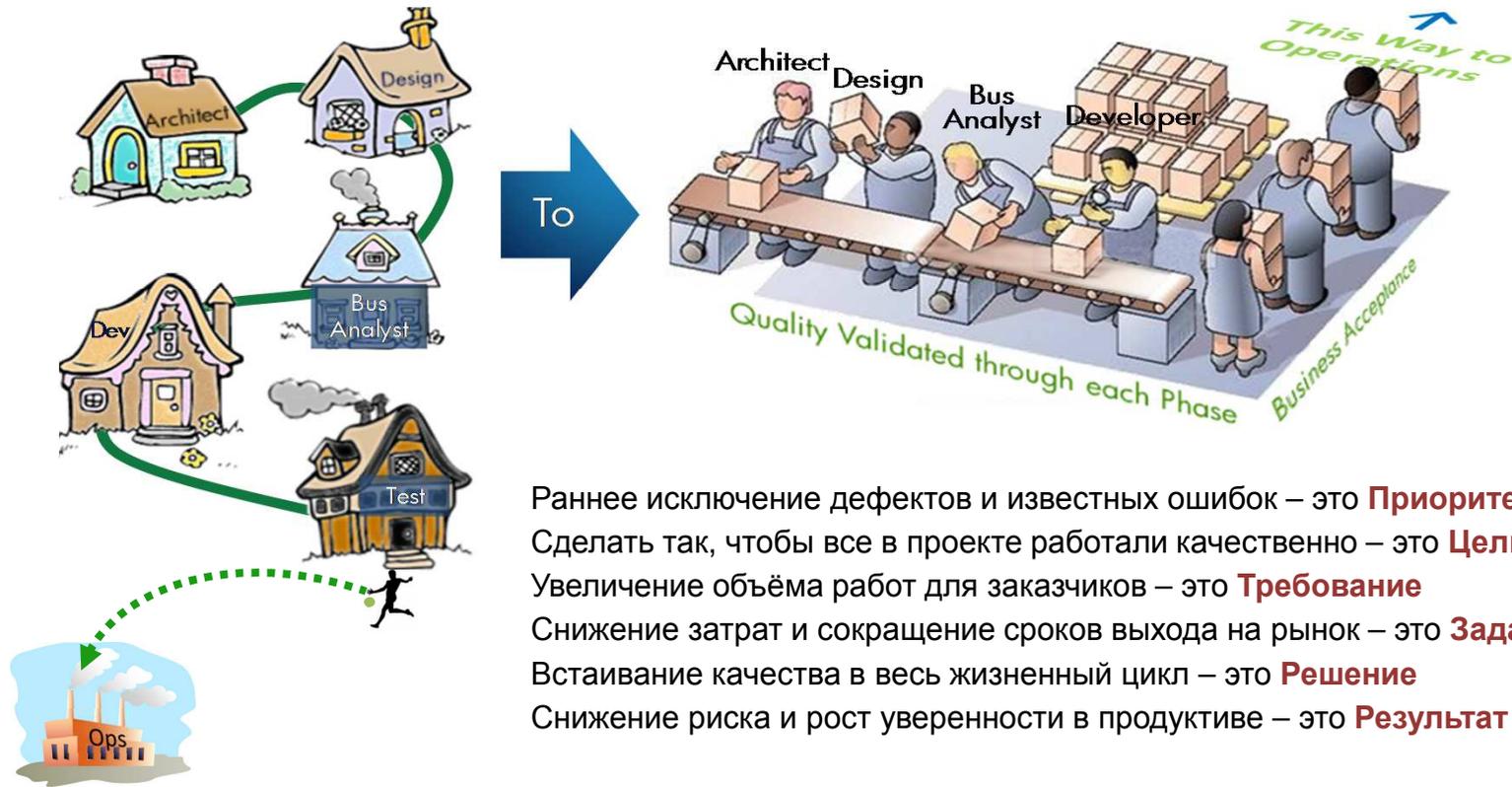
... не единственный возможный ответ.

Так как слишком много проблем уходит в продуктив

... а с ними уже будет разбираться эксплуатация!



Дело не в тестировании, это вопрос всего жизненного цикла



Раннее исключение дефектов и известных ошибок – это **Приоритет**
 Сделать так, чтобы все в проекте работали качественно – это **Цель**
 Увеличение объёма работ для заказчиков – это **Требование**
 Снижение затрат и сокращение сроков выхода на рынок – это **Задача**
 Встраивание качества в весь жизненный цикл – это **Решение**
 Снижение риска и рост уверенности в продуктиве – это **Результат**

Правильный ответ

**Раннее выявление
и решение
проблем за счёт
включения
качества в весь
жизненный цикл**



Передать ответственность за качество разработчикам приложений; Отменить мантру «Разберутся в эксплуатации»



Дать менеджерам изменений прозрачность с точки зрения качества, а также сроков и бюджета



Дать менеджерам по релизам и развёртыванию фактическую информацию для определения готовности к развёртыванию



Использовать SLA, основанные на метриках, для мониторинга достижения уровня SLA и проактивной поддержки услуг

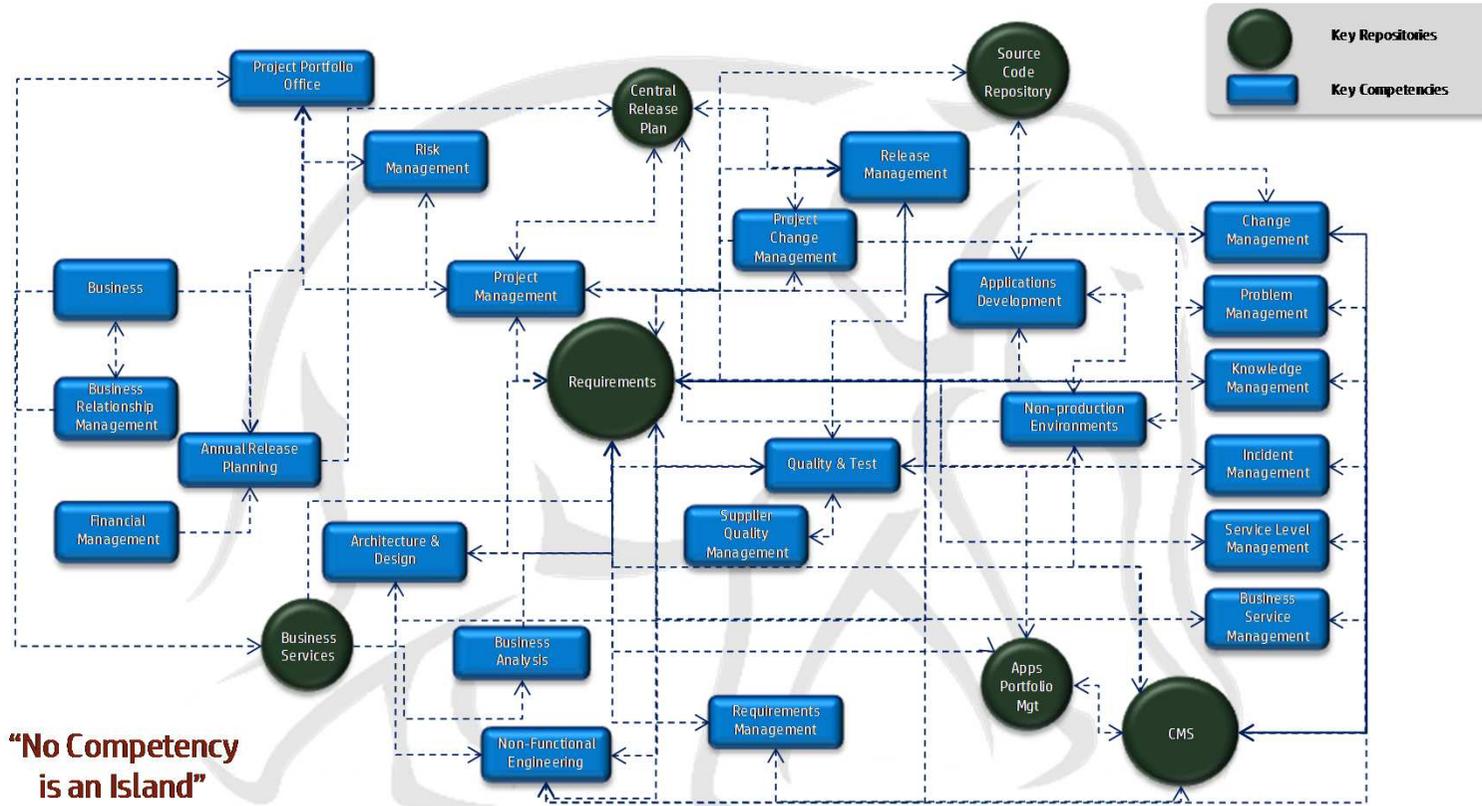


Разработка и эксплуатация – что их объединяет?





Референсный взгляд на управление релизами на предприятии



Основные репозитории

Требования

Основной источник информации о приложениях, которые необходимы для поддержки бизнес-услуг.

Связан с репозиториями, которые описывают разные аспекты каждого требования:

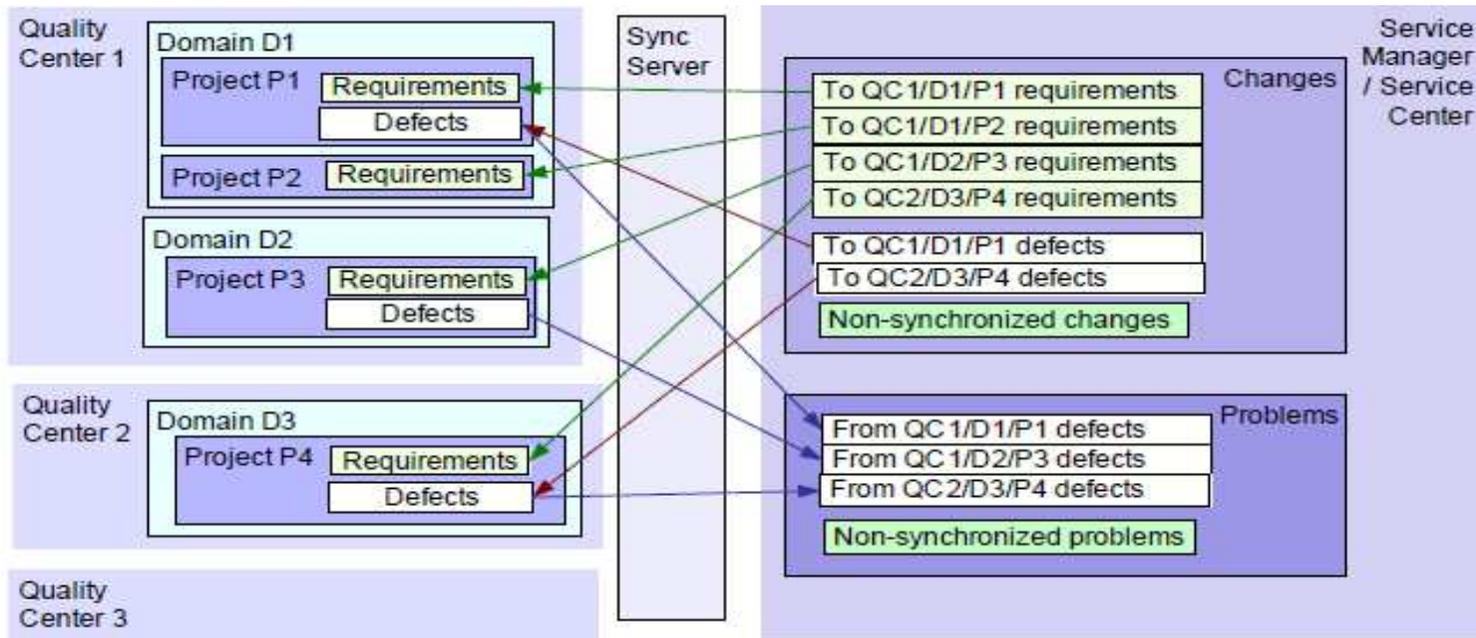
- Функциональные
- Технические
- Инфраструктурные
- Кодовые
- Проектные решения
- Бизнес-процессы
- Тесты
- Дефекты, проблемы и инциденты
- Бизнес-процессы и услуги





Пример федерации данных DevOps

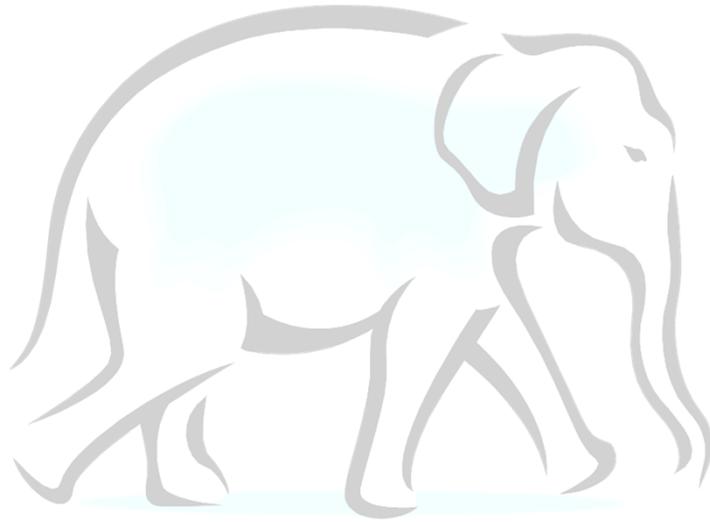
- Использование ALM/QC синхронизатора для HP Service Manager, который связывает следующие сущности:
 - Изменение в SM => Требование или дефект в ALM
 - Проблема в SM <=> Дефект в ALM





Мы поняли, что изменения необходимы

И как теперь их реализовать?



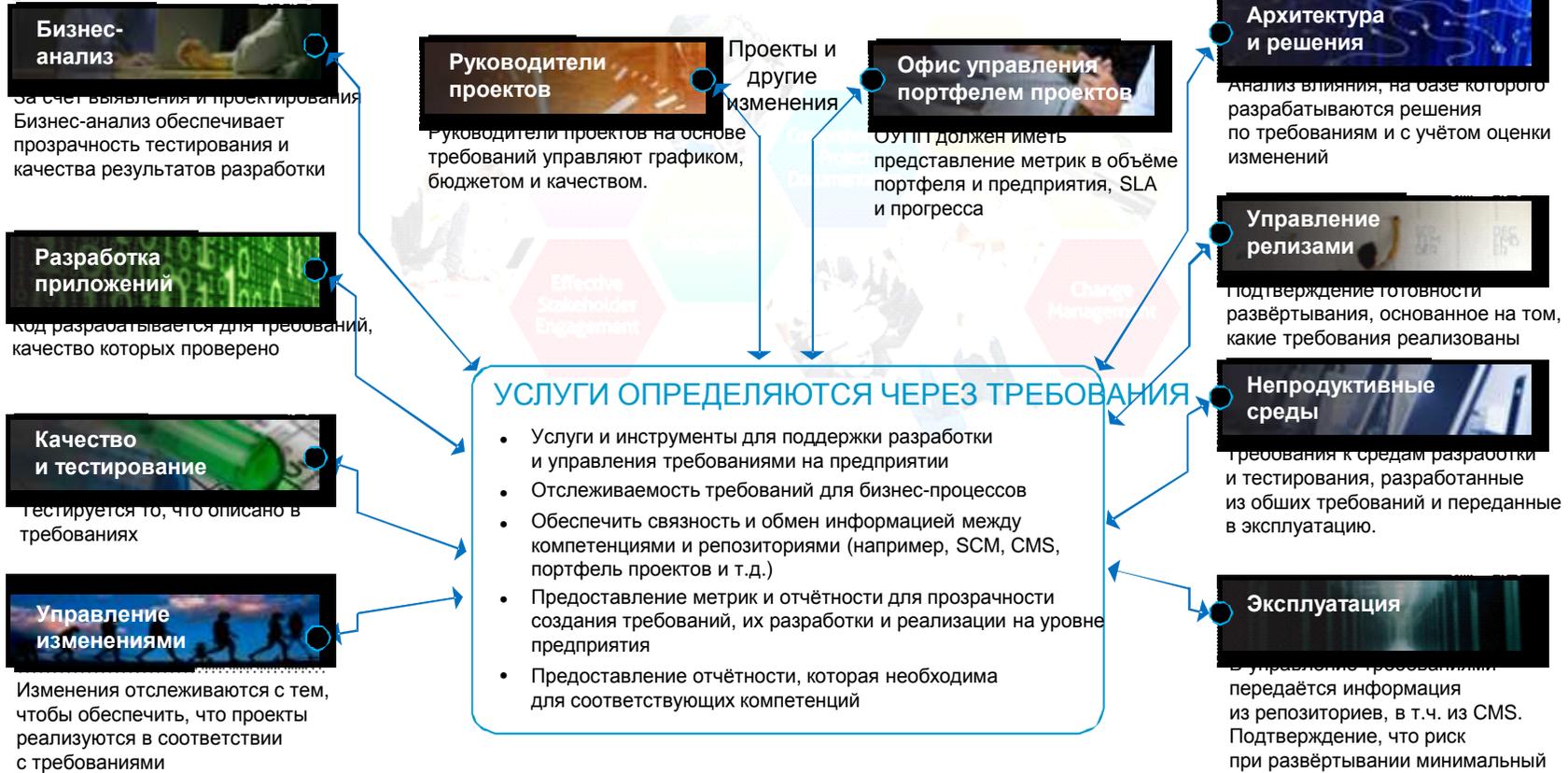
Вот у нас есть слон –

И как мы будем его есть?

Правильно! По частям



Управление требованиями – с этого можно начать





Основные шаги – как «есть слона по частям»?

- 0** Определить, спланировать и управлять программой преобразования
- 1** **Повышение уровня зрелости базовых репозиториях**
 - CMS (автоматизированная инвентаризация, учёт связей и построение ресурсно-сервисных моделей)
 - Требования – CMS и требования будут накапливать зрелость вместе
- 2** **Обеспечить наличие компетенций для использования CMS и требований, в т.ч.:**
 - Управление ресурсами и оценочными показателями
 - Бизнес-аналитики и технические архитекторы
 - Управление сборкой/тестированием и релизами
 - Внедрение и перевод в эксплуатацию
- 3** **Повышение уровня зрелости дополнительных репозиториях**
 - Бизнес-услуги
 - Управление портфелем приложений
 - Централизованный план релизов
 - Репозитории исходных кодов
 - Качество и тестирование, портфель проектов, ресурсы и т.д.
- 4** **Выработать понятные, актуальные и практичные метрики – и отчитываться по ним**
- 5** **Управлять постоянным совершенствованием услуг**



Результаты стратегии управления релизами на предприятии



Снижение затрат



Снижение риска



Ускорить время выхода решений на рынок



Анализ влияния изменений – Дать бизнесу то, что ему нужно, а эксплуатации – что она ожидает



Уверенность в управлении релизами – в яблочко!



Значимая и практичная информация в любое время и из единого источника фактов



«Правильные сразу» развёртывания – не переделываем



Раннее оповещение об успехе (или угрозе) – «Получать информацию заранее и часто»



Призыв к действию

- ***Утром в понедельник***

- Выявить проблемы, которые возникают в эксплуатации из-за проблем в приложениях, которые пришли от разработчиков
- Посчитать затраты, которые вы несёте на устранение этих проблем (сразу в эксплуатации, через срочные патчи/релизы или запросы на изменения к будущим релизам)
- Спросить руководителя разработки, что он/она собирается со всем этим делать

- ***В ближайший квартал***

- Формализовать выводы и обсуждения задач для «понедельничного утра»
- Разработайте план мероприятий, как устранить вопросы по качеству
- Определите основные репозитории и проранжируйте их по результативности
- Разработайте «план роста» с понятным ROI и бизнес-кейсом

- ***В следующем году***

- Внедрить репозитории CMS и требований, а также метрики и отчётность
- Обеспечить, чтобы все в разработке понимали, что каждый из них отвечает за качество



***Спасибо за внимание!
Вопросы?***



hp.com