

# Среда Промышленной Разработки Программного Обеспечения

Евгений Веселов



# Интегрированность с Окружением

- Главная Черта Современного ПО – Интегрированность с Окружением:
  - По функциям (другие пакеты, другие способы применения)
  - В пространстве (файлы, базы данных, интра-экстра-сети, Web-сервисы, почта, workflow, ...)
  - Во времени (свои версии, версии других пакетов, версии платформы)

# Главные источники затруднений для разработки

- По функциям – как предусмотреть многоплановость?
- В пространстве – случайность поведения среды
- Во времени – совместимость назад и вперед, установка side-by-side, upgrade, patch, deployment, изменчивость требований

# Методология Промышленной Разработки

- Когда-то давно:
  - Формальные спецификации
  - Доказательство правильности
  - Структурирование
  - Нисходящее проектирование
- В недавнем прошлом:
  - Формализация производственного процесса
  - Управление проектами и ресурсами
  - Системное моделирование (UML)
- Сейчас:
  - Agile Software Development Ecosystem (активная коммуникация, личная мотивация, разделение ответственности, командный дух)

# Впечатления из Производственного Цеха

- Конвейер сборки программы
- Три-баланс организации разработки
- Многофазный цикл фиксированной продолжительности
- Кривая людских ресурсов

# Конвейер Сборки Программы

- Задача:
  - Необходимость обеспечить параллельную работу многих разработчиков
- Проблемы:
  - Изменения разрушают сделанное ранее
  - Множественность и подвижность участников разработки -> отсутствие цельного видения у отдельных разработчиков
  - Изменчивость платформ, средств разработки и смежных программ в процессе разработки

# Технологические Принципы

- Поддержание целостности сборки (build)
- Авто-тестирование сборки (automation)
- Линейная последовательность внесения изменений

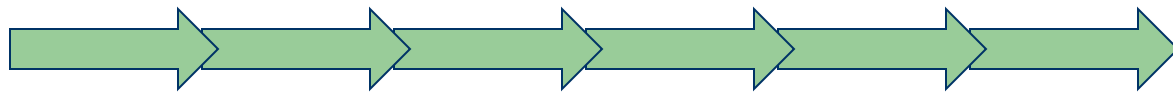
# Как Это Происходит

- Основа – система управления версиями (source code control)
- Подготовка изменения: check-out, кодировка, отладка, проверка, структурный просмотр
- Постановка изменения в очередь – вход в конвейер
- Принятие изменения: компиляция, функциональная проверка, проверка производительности, занесение в базу или отвержение, оповещение

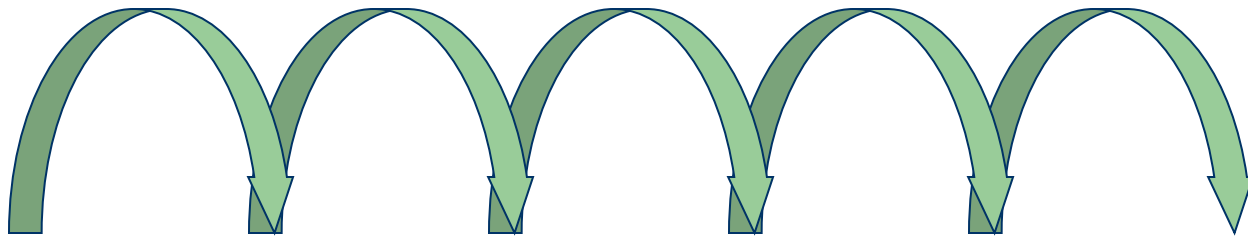


# Структура Конвейера

- Классический конвейер:



- Конвейер программной сборки:



# Конвейер Программной Сборки

- Построение программы:
  - 2 (2+) супермашины – debug & retail builds
- Функциональная проверка:
  - 100-200 машин – параллельный прогон функциональных авто-тестов
- Проверка производительности:
  - 200-300 машин – параллельные группы тестов производительности
- Приемка в главную базу данных

# Обслуживание Конвейера

- Клиент on-line монитора:
  - Текущий статус очереди
  - Изменение приоритетов, досрочное снятие
  - Ход исполнения построения и сборки
- Супервизор очереди:
  - Раннее обнаружение конфликтов
  - Слияние изменений
- Сервер авто-тестирования:
  - Освобождение клиентского ресурса
  - Сертификация изменения для ввода в конвейер

# Результат конвейера

- Постоянное поддержание действующей сборки:
  - за счет незначительного усложнения индивидуальных изменений
  - группа никогда не заблокирована в дальнейшей разработке
  - при постоянном гарантированном контроле качества сборки

# Три-баланс («Triage») оргструктуры разработки

- В доисторические времена:
  - Постановщики-алгоритмисты
  - Программисты-кодировщики
- В прошлом:
  - Разработка как единый процесс конструктивного доказательства правильности программы
- Вариант действующий на практике:
  - Разработчики: конструирование архитектуры, алгоритмы, код
  - Тестеры: обнаружение ошибок
  - Program Managers: представитель потенциального клиента

# Характер работы в triage

- Разработчики:
  - Архитектура и Design Documents
  - Алгоритмы
  - Код, авто-тесты
  - План-графики работы
- Тестеры:
  - Test-cases
  - Баги – инициация и завершение workflow
  - Авто-тесты
- Program Managers:
  - Спецификации: 1-liner, 1-pager, full spec
  - Координация работы
  - Выбор Adds-Cuts

# Принципы три-баланса

- Организационная независимость трех взаимодействующих групп
- Противопоставление целевых установок и критериев поощрения
- Консенсус (не использование «решающего голоса»)
- Количественное соотношение:
  - Разработчики : Тестеры : Pr.M. = 5 : 6 : 1

# Фазовый цикл разработки

- Планирование :1
- Разработка :1
- Стабилизация :1
  
- Этапы определяемые фиксацией времени



# Дуга Людских Ресурсов (Curve)

- Стимулы и критерии:
  - Партнерство
  - «Видимость», влияние на других
  - Владение направлением (“ownership”)
- Дуга напряженности:
  - 30% – Передовики
  - 45% - Основной состав
  - 25% - Отстающие