



НАСЛЕДИЕ ДРЕВНИХ

Как учесть все ограничения legacy-систем страховой компании и запустить процесс Continuous Deployment?

НЕМНОГО О СЕБЕ

<> 10 ЛЕТ В СФЕРЕ ИТ

<> В ЕРАМ С 2008 ГОДА

<> НАЧИНАЛ КАРЬЕРУ В КОМПАНИИ С
ПОЗИЦИИ РАЗРАБОТЧИКА

<> СЕЙЧАС МЕНЕДЖЕР ПРОЕКТОВ

<> УСПЕШНЫЙ ОПЫТ ВНЕДРЕНИЯ AGILE



ЕРАМ – НАДЕЖНЫЙ ПАРТНЕР



ОКРУЖЕНИЕ И ПРЕДПОСЫЛКИ



1

Agile - чтобы выжить

2

Расширение состава:
от 1 до 10 команд (65 человек)

3

Смесь методологий (SCRUM, Kanban) и различные сроки поставок

4

Большое количество legacy-кода в ядре

5

Обязательное приёмочное тестирование заказчиком

ОГРАНИЧЕНИЯ И ПРОБЛЕМЫ

- 1 **Много команд** - сложно работать с общей кодовой базой
- 2 **Много фич** - нужно тестировать, не мешая друг другу
- 3 **Частые релизы** в тестирование и прод - невозможность делать ручную
- 4 **Time to Market** - **изменения должны выходить быстро**
- 5 **Недостаток времени** на регрессионное тестирование - необходимо выявлять максимум ошибок автоматически
- 6 Пользователи по всей России - **нет технологического окна** для обновления ПО

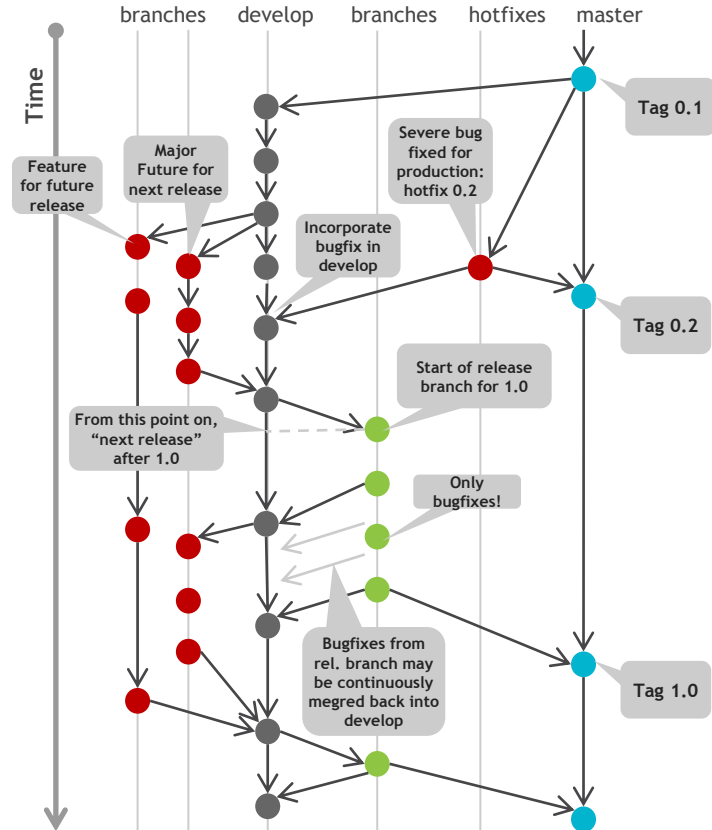
ЧТО БЫЛО СДЕЛАНО

- BRANCHING STRATEGY
- FEATURE TOGGLING
- QUALITY GATES
- BLUE GREEN DEPLOYMENT



BRANCHING STRATEGY

- Основана на **gitflow**
- Каждая User Story (feature) имеет **свою ветку** в git
- Полноценное **тестирование** ветки на выделенном стенде
- **Слияние** feature-ветки **с develop** после тестирования
- Проведение **смек-тестирования** основной ветки



FEATURE TOGGLING

- Каждая новая функция (feature) должна быть отключаема
- Код **попадает в сборку**, а функционал ожидает завершения приёмочного тестирования
- Код **не накапливается** на стороне разработки, а регулярно внедряется на промышленную среду
- Периодическая **чистка кода**



```
if (FeatureManager.GetFeatureEnableState("PolicyKasko"))  
{  
    //on  
}  
else  
{  
    //off  
}
```


QUALITY GATES

PRE-DEPLOY

- **Юнит-тесты** на новый код - базовый контроль
- **Sonar** - статический анализ кода
- **Автотесты** бизнес-логики - контроль **стабильности** legacy-кода
- Peer **code review** - финальный контроль

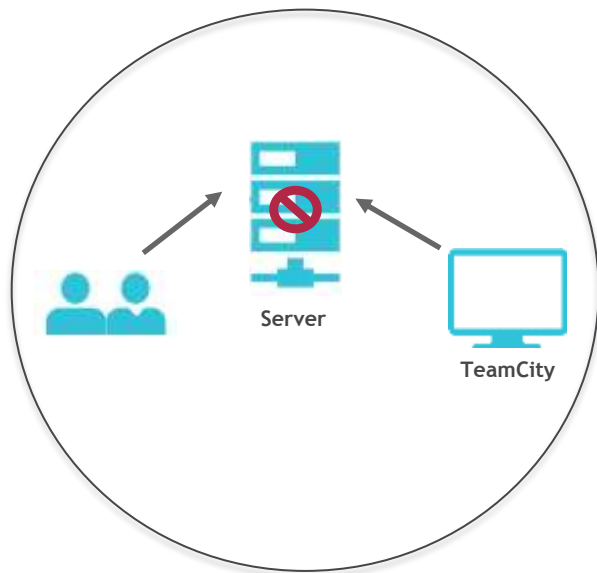
POST-DEPLOY

- Ручной и автоматический **мониторинг массовых ошибок**



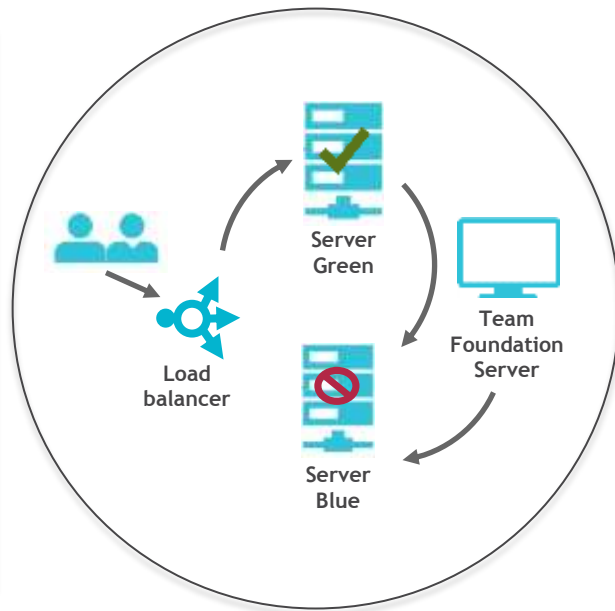
КЛАССИЧЕСКОЕ РАЗВЁРТЫВАНИЕ

- Во время **установки** сервер **недоступен**
- **Откат** происходит **некоторое время**, приложение снова недоступно
- Нет возможности проверить работоспособность билда до его установки для всех пользователей



BLUE GREEN DEPLOYMENT

- Билд устанавливается на неактивный сервер
- В случае необходимости отката он происходит **мгновенно** путём переключения
- После установки билда можно **проверить его работоспособность** на боевом окружении, до того как направлять на него пользователей.



ЧТО В ИТОГЕ

1

Нет зависимости от сроков приёмочного тестирования заказчиком

2

Надёжность внесения изменений в legacy-код

3

Максимально частая интеграция кода

4

Непрерывное внедрение незаметно для пользователя





СПАСИБО!

НИКИТА ЕВСЕЕНКО



www.epam-group.ru

www.epam.com