



Mstislav (Slava) Martynyuk

Solution Architect

Company Founder

Camunda Certified Engineer

<https://reunico.com>

MMartynyuk@reunico.com

Как мы разработали с нуля и внедрили за 2 года систему с 70+ бизнес-процессами

*А потом поняли, что прошли Customer Success Path от Camunda

И как нам помогли в этом Scrum и Agile

У нас было:

- бизнес-процесс, подготовленный аналитиком заказчика в Camunda Modeler;
- разработчик, архитектор и менеджер проекта;
- GitLab - для хранения исходного кода и управления задачами;
- Camunda Standalone Engine;
- 50 пользователей;
- комната для работы команды и проведения встреч
- ограниченный бюджет :-)



Но не было понимания, что будет представлять из себя
результат нашей работы



Christopher Columbus on Santa
Maria in 1492, Emanuel Leutze,
1855

Scrum

- Спринт (Sprint)
- Бэклог (Backlog)
- Пользовательская история (User Story)
- Demo (Demo)
- Планирование спринта (Sprint Planning)
- Scrum Board
- Работающий продукт важнее документации
- Взаимодействие бизнеса и ИТ важнее ТЗ

Спринты 1, 2: Proof of Concept

- Провели верификацию и валидацию требований, подготовили пользовательские истории и сформировали бэклог
- Развернули песочницу (Camunda BPM Standalone Engine)
- Смоделировали исполняемый бизнес-процесс с использованием TaskList
- Базовая интеграция с сервисами предприятия (уведомления через почту и аутентификация в Active Directory)
- Валидировали бизнес-процесс с заказчиком и участниками процесса, внесли изменения

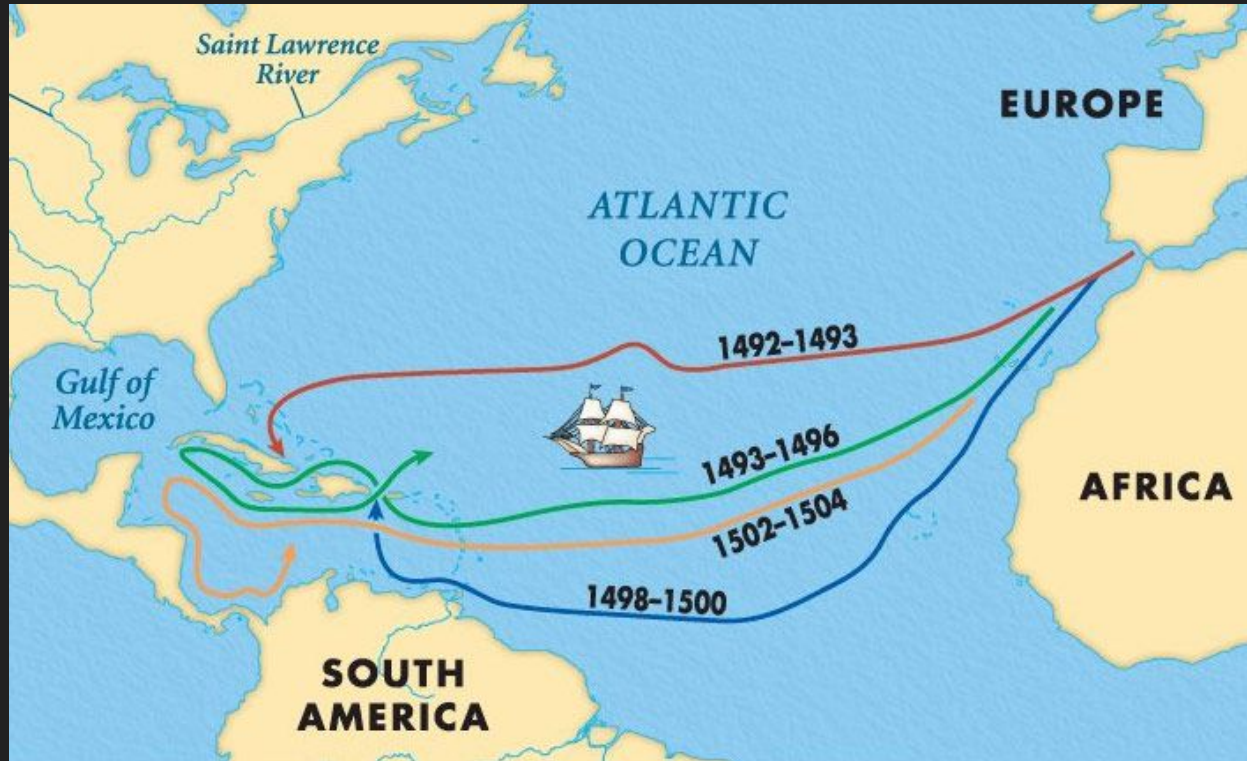
Спринты 3,4,5,6,7,8: Pilot

- Решение для хранения бизнес-данных (Spring Boot, Angular)
- Перенесли процесс на Camunda Embedded Engine
- Вспомогательные сервисы (загрузка данных, формирование отчетности)
- Custom Task List (Angular) + Camunda REST API
- Миграция данных
- Приемка
- Запуск в прод

Эксплуатация, оценка, улучшение



Применение полученного опыта для других процессов



Применение полученного опыта для других процессов

- Усиление команды: +аналитик, +2 разработчика, +DevOps
- COVID!!! Команда перешла в онлайн

Добавлены артефакты Scrum:

- Daily Meeting
- Ретроспектива
- Покерное планирование (оценка задач в майках - S, M, L, XL)

Архитектура:

- ++ jHipster, Kubernetes, Elasticsearch, Kafka, PostgreSQL (Cluster)

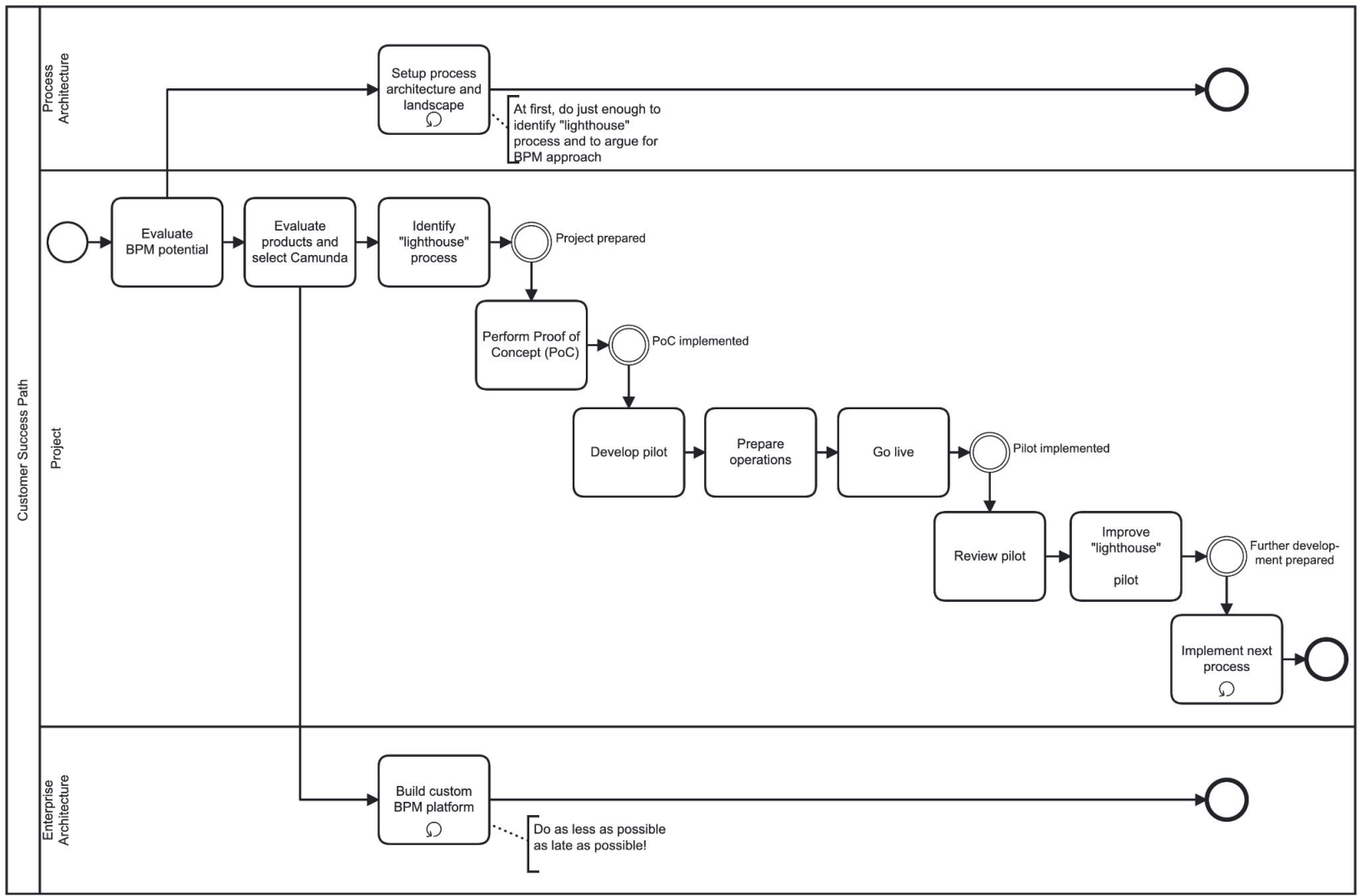
Итог спустя 2 года:

- Идеальная команда: 2 разработчика, тимлид разработки, скрам-мастер, аналитик, QA
- Одна подсистема (со вспомогательными сервисами): 3 месяца
- 9 подсистем
- 1300 пользователей
- 19 микросервисов
- 10 точек интеграции
- 4 стенда (prod, UAT, pre-prod, dev)
- 3000+ задач
- 2703 MR
- аудит кода
- защита архитектурного решения

Существующие вызовы спустя 2 года:

Меняются (в порядке убывания):

- люди (проектная команда, заказчики, затраты на вовлечение снижены благодаря BPMN);
- бизнес-процессы (улучшения, приспособление к изменениям среды);
- технологии (в меньшей степени).



Что надо делать?



- Следовать Customer Success Path
- Использовать гибкую методологию
- Процесс для пилота выбрать исходя из соотношения усилия/результат
- Обеспечить тесное взаимодействие команды разработки и бизнеса
- Избегать параллельной разработки (идеально, если команда запустившая пилот, делает последующие процессы в том же составе)
- Прототипировать и моделировать (модели процессов, скетчи экранных форм, диаграммы, etc)

Чего стоит избегать?



- Моделировать в BPMN все процессы организации (достаточно ограничиться верхнеуровневым описанием: SIPOC, карта процессов)
- Заранее проектировать инфраструктуру и подсчитывать необходимые мощности (CPU, RAM, storage)
- Точной оценки трудозатрат (оценивайте работы исходя из структуры процесса - пользовательские задачи, используемые сервисы)
- Начинать крупный проект без PoC или пилота

Спасибо за
внимание!

