Архитектура

Sergey Kuntsevich
Digital CTO

Планирование

- Планируем 5 из 6 итераций (по 2 недели). Скрам мастера ответственны за определение capacity команды на каждую итерацию
- От каждой команды ожидается постановка целей на инкремент программы и каждую итерацию. Не забываем о «целях без уверенности» там где существуют неустранимые риски. Цели без уверенности должны быть обоснованы
- Норма: 5-7 целей с уверенностью на команду. 1-3 целей без уверенности.
- Команды ответственны за их формулировку и получение Business Value от владельцев бизнеса и Product Management Team.
- **Метрика Flow Predictability** – для всех команд будет отслеживаться и иметь ключевое значение для оценки результатов нашей работы
- Миссия Core поезда и Core команд – это внедрение лучших технологий и практик, а не разработка enablers для бизнес-фич

Релизы и поддержка

- Резерв на поддержку – 20% от Capacity команды.
- Под поддержкой понимаем неожиданные проблемы с продом/задачи от офиса генерального директора
- Не понимаем под поддержкой – блокирующие дефекты релизов.
- Не понимаем под поддержкой – старые проблемы.
- После каждого спринт-планинга создается резервирующий время enabler story на фиксированное количество Story Points
- При возникновении запроса на поддержку – создается *enabler-story*, оценка SP корректируется соответствующим образом
- Критерии приемки историй и релизов находятся по URL: https://alkorco.atlassian.net/wiki/spaces/QA/pages/2939846695/Release+DoD+Quality+Criteria

Web Architecture Guidelines

- Код, который выводится куда-либо, должен пройти проверку QA.
- Новые версии и новые имплементации спринговых эндпоинтов описываются в openapi до начала разработки и согласовываются с фронтами и бэкендом.
- DDM - все новые скрипты, которые добавляются через интерфейс ddm, должны проходить проверку и доработку при необходимости со стороны frontend
- DDM - события для аналитики должны быть зафиксированы в критериях приёмки задач.
- Функционал web приложения разрабатывается на vue, нокаут - в исключительных ситуациях.
- Новый функционал не вызывает JS ошибок в instana, текущие ошибки исправляются в процессе работы (EDP-17360)

iOS Architecture Guidelines

- Основная архитектура проекта - MVVM. Для связи View с ViewModel используется биндинг на основе Combine (до перехода на iOS 13 - OpenCombine)
- Для новых фич предусмотреть фича-тогл. Отказ от фича тогл по согласованию с Архитектором
- Тестовое покрытие. Новый код должен быть покрыт Unit-тестами как минимум на уровне логики. ViewModel - это логика.
- UI контролы должны содержать AccessibilityId и быть доступны для UI-тестирования.
- Верстка UI осуществляется в коде
- Навигация через Router. Любой вновь создаваемый View Controller должен быть зарегистрирован в фабрике. Навигации через segues следует избегать.
- Включение новых сторонних зависимостей должно быть согласовано с архитектором.

Andorid Architecture Guidelines

- Основной язык разработки - Kotlin, весь новый код пишется на нем, старый код на Java постепенно переводится на Kotlin с покрытием тестами при переводе.
- Основная архитектура приложения - MVVM. Используется ViewModel и LiveData от Android Jetpack, для взаимодействия с потоками и стримами данных - RxJava.
- Для новых фич предусмотреть фича-тогл. Отказ от фича-тогл по согласованию с архитектором
- Код покрывается тестами, минимально - unit тесты для логики из ViewModel и helper классов, желательно - еще и покрытие happy flow нового функционала UI тестами на Kaspresso.
- Зависимости в коде предоставляются через Hilt, используются интеграции с Core модулями.
- Для комплексных layout'ов используется ConstraintLayout вместо множества ViewGroup с большим уровнем вложенности.
- Подключение новых библиотек в gradle конфигах - через согласование с архитектором.

Back-end Architecture Guidelines

- Каждая новая фича должна оцениваться для варианта ее реализации вне Oracle ATG.
- Реализация новых (ранее не существовавших) сервисов или фич на Oracle ATG отдельно согласуется с Архитектором
- В случае необходимости изменения логики внутри REST MVC, вместо доработки должна быть выполнена модернизация на Spring REST. Коммиты в старые REST должны блокироваться на code review

Engagement Architecture Guidelines

- Для всех больших новых фич предусмотреть фича-тогл (флаги на уровне картриджей, включатели в бсс на уровне сайта, флаги на уровне компонентов, в зависимости от ситуации). Отказ от feature toggle по согласованию с техническим лидером
- При разработке нового функционала с новыми картриджами полностью уходить от подхода передавать данные в самих картриджах (только конфигурационные настройки). Все данные должны получаться с помощью существующих/новых Spring сервисов.
- Запланировать изучение протокола OData и соответствующих библиотек для Java, Js перед началом реализации первого этапа НПЛ.

Dev Ops Architecture Guidelines

- Правила для работы с GIT:
- Имя репозитория должно соответствовать регулярному выражению: [a-z][a-z0-9-]+;
- Репозиторий не должен содержать:
  - Связок логин/пароль;
  - Токенов;
  - Сертификатов;
  - Приватных ключей;
  - JKS файлов;
  - Архивов и иных бинарных файлов, например, JAR
  - Имен и путей с пробелами и киррилицей.
- Репозиторий должен содержать файл .gitignore, в котором указаны пути к результатам сборки проекта;
- Репозиторий должен иметь защищенные ветки:
  - master
  - develop
  - Веткри команд типа TEAM_NAME/develop
- Защизенные ветки должны иметь следующие ограничения:
  - Allowed to merge: Developers или Developers or Maintainers;
  - Allowed to push: No one.
- Изменения в защищенные ветки должны вноситься исключительно через merge request'ы;
- Merge допускается только в случае успешного завершения пайплайна.