

Технологии программирования

Лекция №11

ИС, весна 2022

Управление
разработкой

Документирование

- Зачем?
- Bus factor
- Шаринг знаний
- Аудитория

Тулинг для документации

- Комментарии в коде
- Гитхаб вики
- Медиавики
- Конфлюенс
- Ридми

Meeting notes

- Фиксация результатов
- Шаринг информации вне митинга
- Шаринг знаний
- Фиксирование плана, согласования действий

Architecture decision record

- Время
- Статус
- Контекст
- Решение
- Последствия

Эстимейты

- Оценка или обязательство
- Сообщать о проблемах эстимейтов
- Декомпозиция
- Закон больших чисел
- Умножение на два
- Покер планинг

От идеи до прототипа

- Ценности проекта
- Ориентир на пользователей
- Домен
- Глоссарий

User stories

- As a [type of user],
- I want [an action]
- so that [a benefit/value]

As is / to be процессы

- Формализация текущего процесса, описание его этапов, входных и выходных параметров
- Проектирование итогового процесса, проверка соблюдения процесса, соответствии параметров
- Анализ изменений между процессами

Постановка задачи

- Идентификатор задачи
- Название и описание
- Состояние - открыта, в процессе, выполнена
- Назначенный ответственный
- Приоритет.

Тестирование

- TDD
- Юнит тесты
- Интеграционные тесты
- Ручное тестирование

Что есть баг

- Не смотря на то, что баги - это зона ответственности QA, у разработчиков должно быть выстроено понимание как с ними работать
- Баги как и задачи должны трекаться и быть рядом с кодом
- Описание баги всегда должно оперировать двумя состояниями - ожидаемое поведение и текущее
- Ассоциация с определённой версией продукта,

Мейлстоуны

- Релиз вовремя
- Прототипы и MVP
- Фидбек пользователей

Рост стоимости задач

- Проблема на уровне компиляции
- Проблема на уровне тестов
- Проблема на уровне ручного тестирования
- Проблема выявлена кастомером
- Проблема, которая включается поддержку форвард совместимости

Meetings

- Дейли
- Викли
- Планинг
- Ретро

Code review

- Валидация
- Шаринг знаний
- Оценка сверху вниз - от архитектуры до кодстайла
- Код ревью - это саджесты, а не перекладывание ответственности
- Токсичность в ревью

Github как инструмент для всего

- Профиль, почта, GPG
- История комитов, аннотейты
- Управление ветками
- Связь репозиториев гит и гитхаба
- Пермишены

Github для управления задачами

- Пул реквесты
- Ревью
- Ишуи
- Мейлстоуны
- Проджекты

Github action

- Встроенные механизмы CI
- Настройка билда и тестов
- Настройка деплоя
- Настройка статических анализаторов

Проектирование взаимодействия между компонентами

- API, проектирование
- Компоненты в рамках одного процесса
- Компоненты в рамках одной ноды
- Транспорт для более глобального взаимодействия
- Swagger

ORM

- Место базы в системе
- Процедурные монолиты, скрипты апгрейда
- Микро-орм, орм, миграции

Чеклист запуска проекта

- Кто есть команда, какие есть роли у в команде? Дев, КА, аналитики, маркетинг, саппорт
- Кто является ответственным за всё?
- Кто будет пользователем?

Чеклист запуска проекта

- Как будет построена коммуникация между командой?
- Какие митинги будут, с какой периодичностью?

Чеклист запуска проекта

- Какая цель, какую проблему решаем?
- Фреймворк, язык, архитектура, кодстайл, база?
- Как будет приложение тестироваться?
- Как будет выглядеть система контроля версий, как там будет всё организовано?
- Какой процесс мёрджа комитов, ревью, автотестов?
- Какой процесс деплоя?

Чеклист запуска проекта

- Когда проект должен быть готов?
- Когда первый мейлстоун?
- Когда нужно будет PoC/Demo?

Чеклист запуска проекта

- Как взаимодействуют разные модули?
- Как взаимодействует код и база?
- Какие есть другие зависимости и как с ними происходит взаимодействие?