


ЖИЗНЕННЫЙ ЦИКЛ СОЗДАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Постоев Дмитрий
postoev@okbsapr.ru

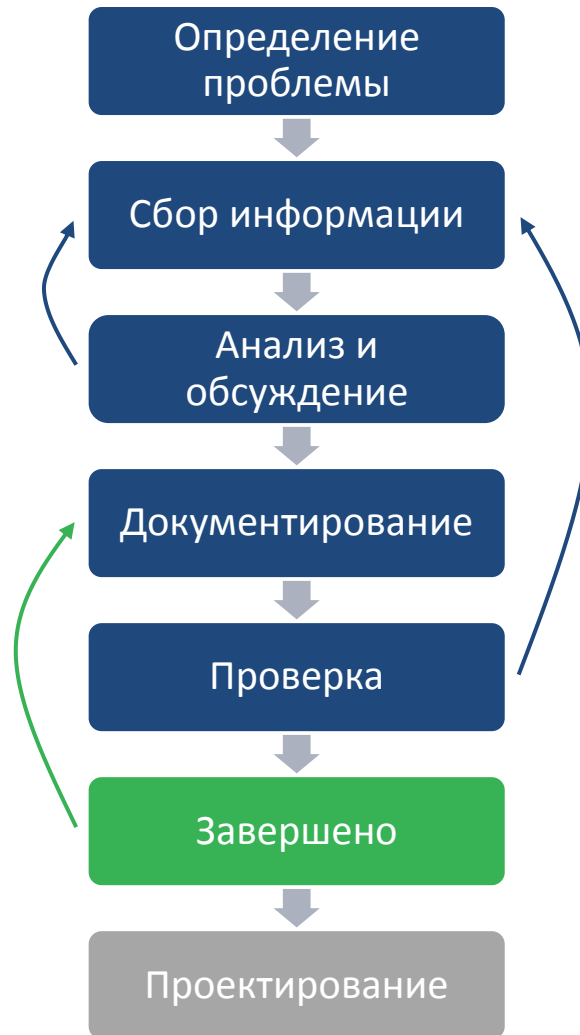
Часть 4:

Формирование требований

Технические требования – набор правил и условий, которым должен соответствовать и которые должен поддерживать будущий продукт. Помимо этого включают в себя процедуры, позволяющие убедиться в том, что итоговый продукт удовлетворяет указанным условиям.^[1]

[1] Professional Software Development Ch. 1, Steve McConnell

Процесс формирования требований



Определение проблемы

1. Зачем делаем? (выявление проблемы)
2. Что делаем? (определение продукта)
3. Для кого делаем? (определение ключевых заинтересованных лиц)

Зачем делаем? Принцип «пяти почему»

Основная задача — выявление ключевого источника проблемы.

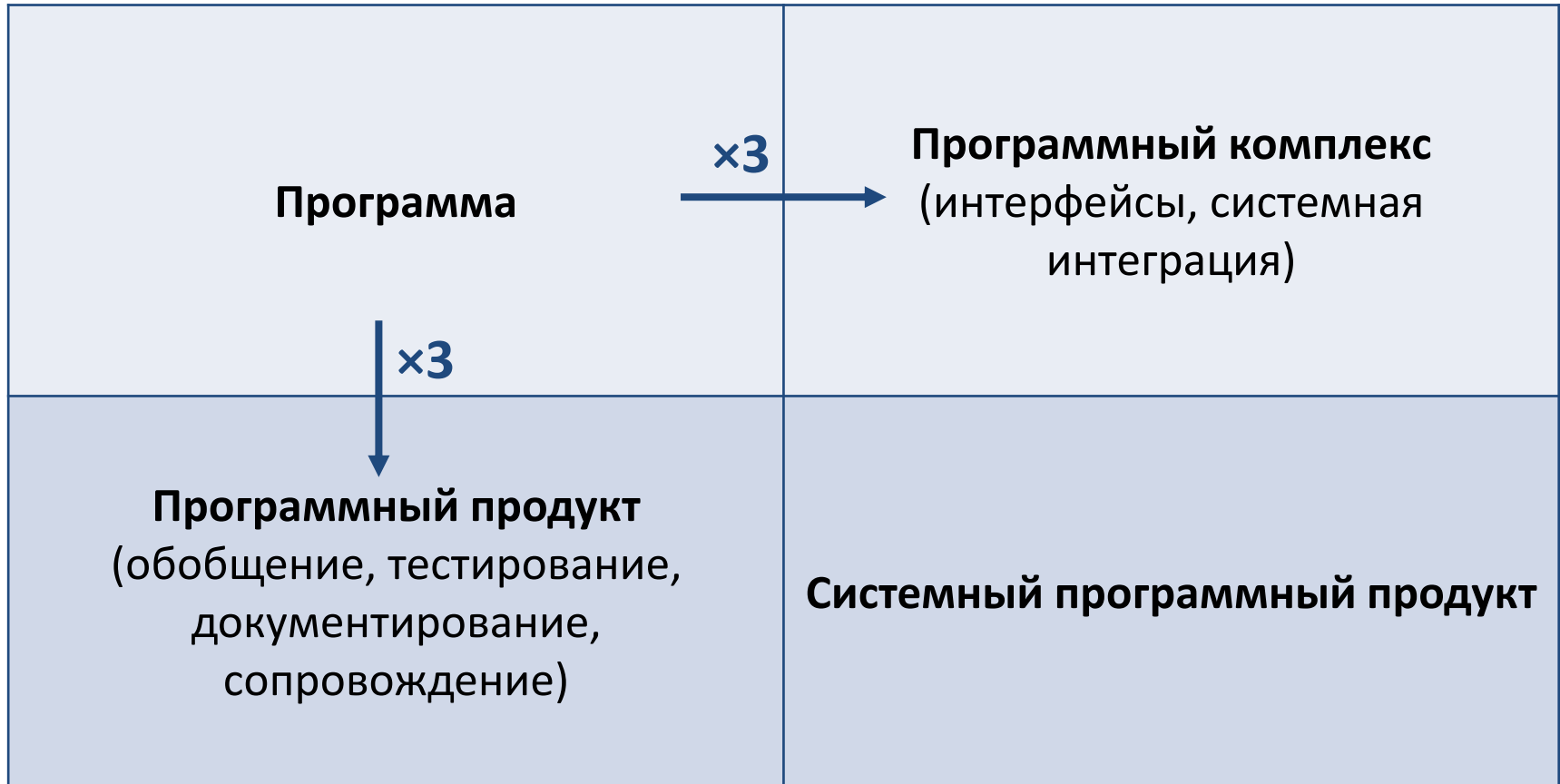
Проблема: не заводится двигатель

1. **Почему?** — сел аккумулятор
2. **Почему?** — не работает генератор
3. **Почему?** — проблемы с ремнем генератора
4. **Почему?** — ремень был в порядке до этого момента, но ни разу не менялся
5. **Почему?** — автомобиль не проходил должного технического осмотра по рекомендуемому графику (первопричина)

Зачем делаем?

1. Описывайте максимально конкретно (без общих слов)
2. Определите свой целевой рынок (аудиторию)
3. Оцените размеры рынка и проблемы
4. Определите текущее (плохое) решение
5. Обезличьте ваше описание

Что делаем? Виды программных продуктов



Для кого делаем?

Стейкхолдеры – человек, группа людей или организация, кто может влиять, заинтересован или может быть затронут решением, деятельностью или результатом проекта.^[1]

Кто относится:

- Клиенты или их представители
- Пользователи ПО
- Руководство компании, руководители проектов
- Спонсоры
- Команда разработки
- Внешние поставщики, консультанты

[1] Стандарт ISO 21500

Сбор информации

- выявление потребностей клиентов, пользователей и других заинтересованных в проекте людей

Виды собираемых требований:

- К продукту
- К процессу
- Функциональные
- Нефункциональные
- Системные

Стратегии сбора информации

Product-centric



Usage-centric



Сбор информации. Что необходимо учесть

1. Ключевых заинтересованных лиц надо опросить как можно раньше
2. Конфликты между клиентами или отделами должны быть разрешены
3. Постоянное взаимодействие с экспертами
4. Изучение скрытых требований и заинтересованных лиц

Анализ и обсуждение требований

- классификация информации, собранной на предыдущем этапе, идентификация недостижимых и непрактичных целей, уточнение потребностей и разрешение возможных конфликтующих требований.^[1]

[1] «Guide to the software engineering body of knowledge», Alain Abran, James W. Moore; Pierre Bourque, Robert Dupuis, 2015

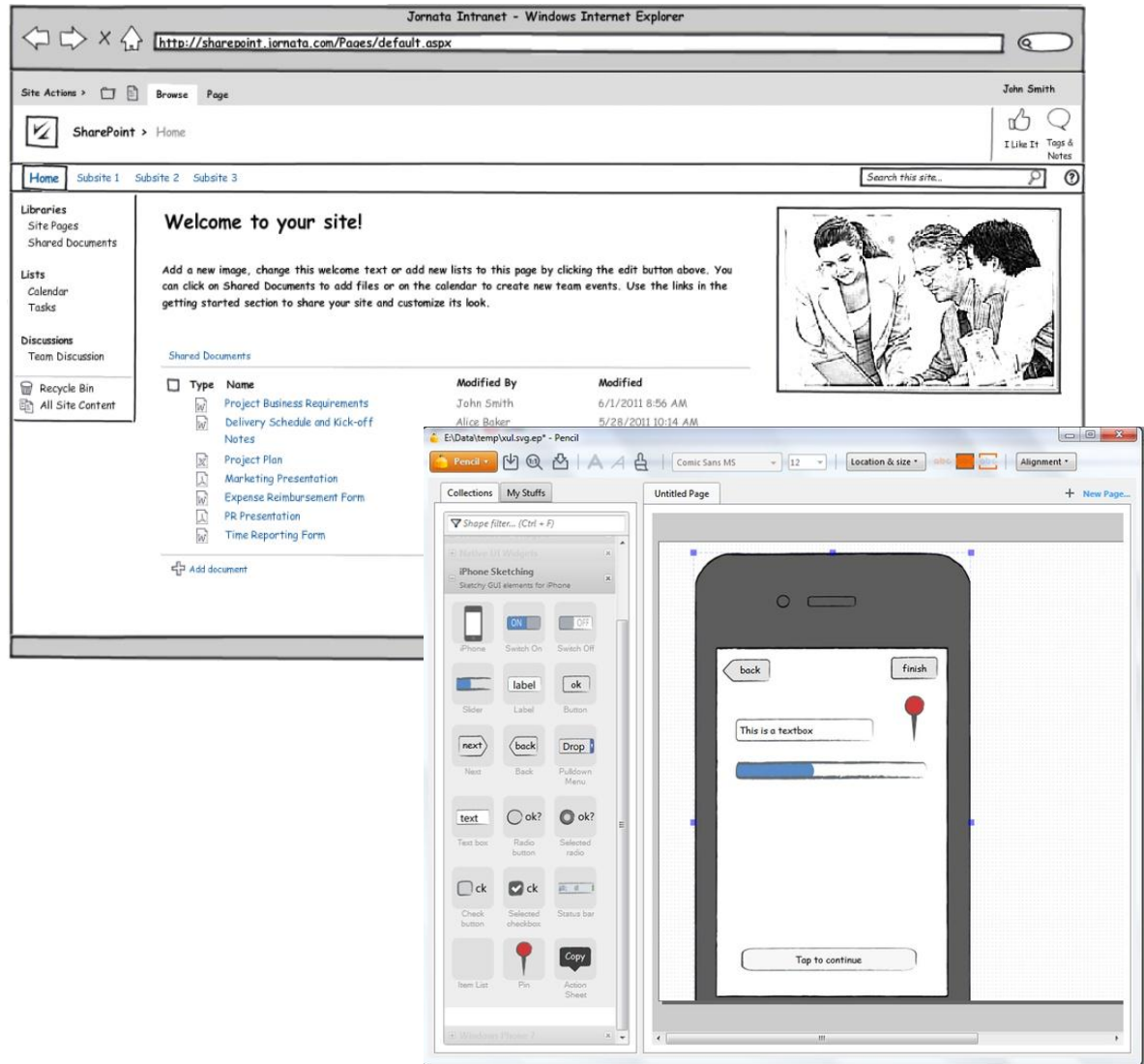
Создание прототипов

Open-source

- Pencil Project
- Inkscape

Proprietary

- Balsamiq
- Lucidchart
- Invision



Документирование требований

- создание документа, имеющего законченное описание программы, которую предстоит разработать (*Software Requirements Specification*, Спецификации требований программного обеспечения).^[1]

[1] Стандарт IEEE 830

Структура документа по стандарту IEEE 830

1. Введение

- Цели, термины, аудитория, источники

2. Общее описание

- Видение, общая функциональность, классы пользователей, среда функционирования, ограничения, зависимости

3. Функциональность системы

- a) Функциональный блок X
 - Описание и приоритет, причинно-следственные связи, workflows
- b) Функциональный блок Y
 - Описание и приоритет, причинно-следственные связи, workflows
- ...

4. Требования к внешним интерфейсам

- UI/UX, программные интерфейсы, интерфейсы оборудования

5. Нефункциональные требования

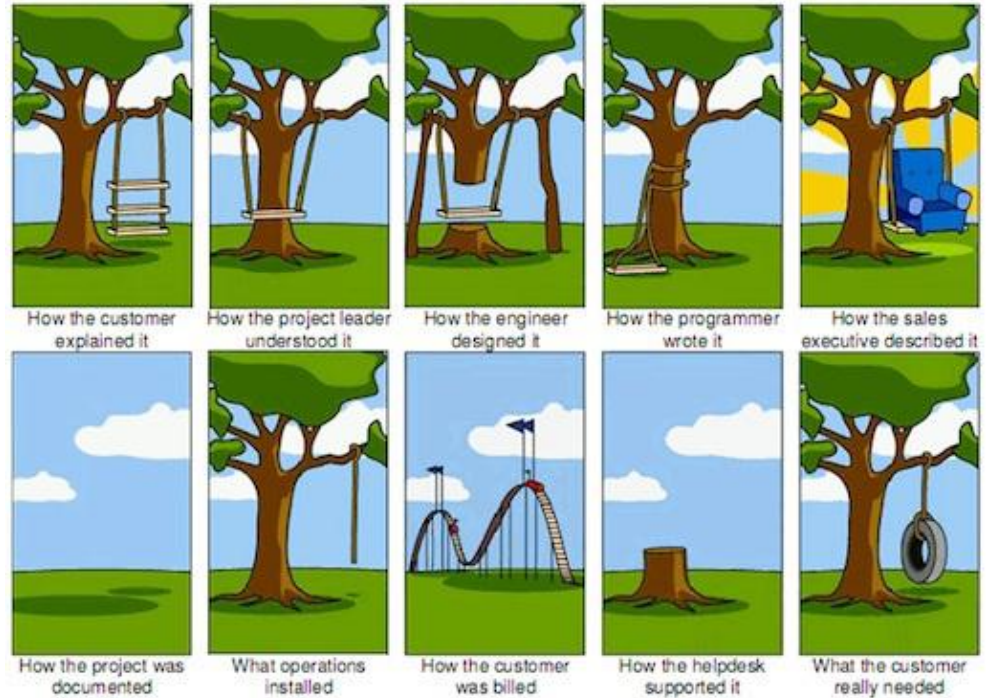
- Требования к производительности, целостности данных, безопасности, критерии качества

6. Другие требования

7. Приложения

Требования к документу

1. Ясность
2. Полнота
3. Краткость
4. Непротиворечивость
5. Достижимость
6. Однозначность
7. Проверяемость
8. Приоритезация
9. Применимость
10. Контроль версий



Проверка требований

- поиск и исключений из требований противоречий, неясностей, пропущенных случаев использования, составление приемочных тестов (Acceptance Tests).

Финансовая сторона:

- Оценка стоимости исполнения
- Возврат инвестиций

Поддержка требований

- формирование требований, как и весь жизненный цикл разработки ПО, это итеративный процесс, поэтому изменение функций и условий функционирования продукта приводят к уточнению и формированию новых требований

Литература

1. «Professional Software Development Ch. 1», Steve McConnell
2. «Code Complete», Steve McConnell
3. «The Mythical Man-Month», Fred Brooks
4. «Software Requirements (Developer Best Practices)», Karl E. Wieggers
5. «Applied Software Project Management», Stellman & Greene

Практическая задача

1. Создание системы удаленного мониторинга производительности компьютеров.
2. Параметры мониторинга: CPU (по процессорам), оперативная память (занято/доступно).
3. Должна иметься возможность вывести список запущенных на компьютере процессов, с именем процесса и потребляемыми ресурсами (CPU, ОЗУ), а также, именем пользователя, от которого процесс запущен.
4. Утилита управления представляет из себя графический или Web-интерфейс, с помощью которого задаются настройки, меняется список компьютеров и просматриваются параметры производительности.
5. Должна быть реализована функция изменения интервала сбора информации с компьютера.
6. Предусмотреть функцию выбора собираемой и выводимой на экран информации о компьютере.
7. Итоговое должно представлять из себя 2 установочных файла (1 – утилита управления, 2 – агент для контролируемого компьютера).
8. Все действия по изменению настроек и состава системы должны записываться в журнал.
9. История собираемой информации должна сохраняться в единую базу данных .
10. В качестве машины, ресурсы которой контролируются могут выступать как Linux, так и Windows машины.

Домашнее задание

1. **Руководителю проекта** необходимо создать репозиторий проекта, настроить права доступа, добавить других участников (*Add Collaborator*) и создать файл README.MD с кратким описанием проекта.
2. **Руководитель проекта** должен назначить задачу **Аналитику** на формирование Спецификации требований программного обеспечения.
3. **Аналитик** по полученным требованиям должен создать, согласно шаблону, спецификацию на создаваемый командой продукт.
4. **Тестировщик** помогает **Аналитику** с составлением требований, чтобы в дальнейшем сформировать на их основе приемочные тесты.

Примечание: Задачи назначаются и оформляются на Github

Полезные ссылки про Github

1. Quick Start Guide - <https://guides.github.com/>
2. Полный справочник - <https://help.github.com/>
3. Приглашение пользователей - <https://help.github.com/articles/inviting-collaborators-to-a-personal-repository/>
4. Создание и назначение задач - <https://help.github.com/articles/managing-your-work-with-issues/>
5. Оформление документации - <https://help.github.com/categories/writing-on-github/>
6. Процесс совместной разработки - <https://help.github.com/articles/github-flow/>