

Основы программной инженерии

Инженерия требований

2012

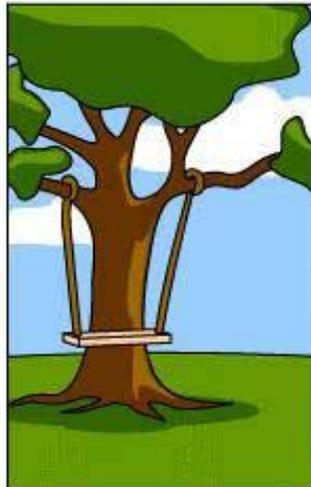
В чем проблема?

«Самой сложной задачей при создании программной системы является точное определение того, что требуется создать... Ни одна задача не приносит такого же вреда конечной системе в случае ошибки. И нет ни одной задачи настолько же сложной в исправлении последствий.»

Фредерик Брукс



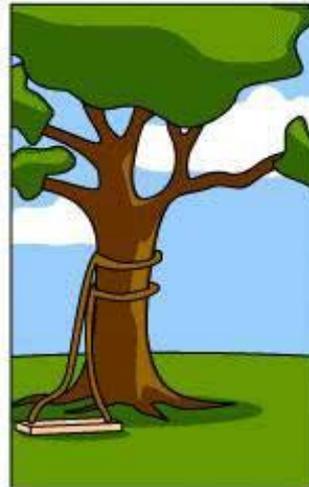
Как объяснил клиент
чего он хочет



Как понял клиента
начальник проекта



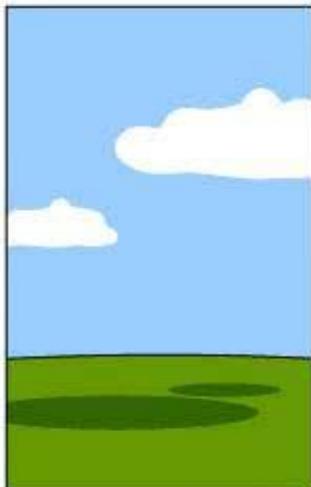
Как описал проект
аналитик



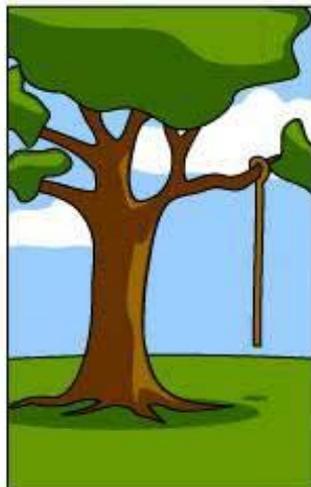
Как написал
программист



Как представил проект
бизнес-консультант



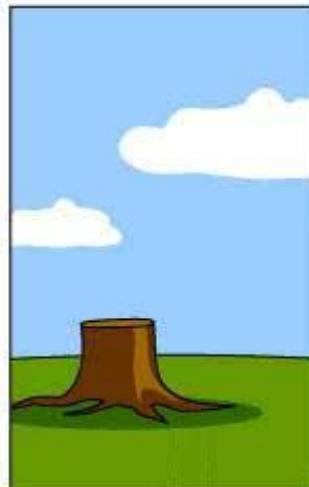
Как
задокументировали
проект



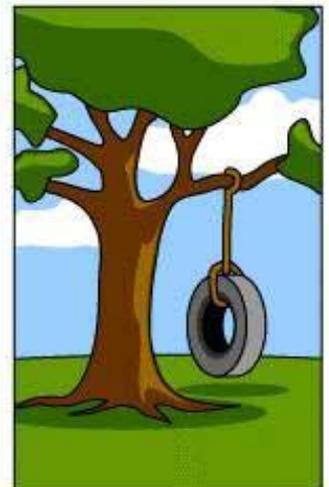
Какие фичи
удалось внедрить



Как заплатил
клиент



Как работала
техническая
поддержка



Что было нужно
клиенту

Проблемы определения требований

- ▶ Разработка требований – самая сложная часть проектирования ПО
- ▶ Требования постоянно меняются
- ▶ Требования могут быть
 - неясны
 - двусмысленны
 - противоречивы
- ▶ Спецификации могут быть неполны
- ▶ Пользователи, излагающие требования, непредставительны

План раздела «Инженерия требований»

- ▶ Определение требований
- ▶ Разработка требований
 - Выявление требований
 - Анализ требований
- ▶ Документирование и организация требований
- ▶ Изменение требований
- ▶ Планирование и управление требованиями

1. Требования

Требование по IEEE 1990:

- ▶ Условие или возможность, необходимые пользователю для решения его задач или достижения цели.
- ▶ Условие или возможность, которым должна отвечать или которыми должна обладать система или ее компонента, чтобы удовлетворить контракт, стандарт, спецификацию или иной формальный документ.
- ▶ Документированное представление условия или возможности, указанное в (1) или (2)

Свойства требований

- ▶ Корректность (correct)
- ▶ Однозначность (unambiguous)
- ▶ Полнота (complete)
- ▶ Непротиворечивость (consistent)
- ▶ Приоритезация (prioritized)
- ▶ Проверяемость (verifiable)
- ▶ Модифицируемость (modifiable)
- ▶ Отслеживаемость (traceable)

Требования

- ▶ Виды требований:
 - Функциональные требования
 - Бизнес-требования
 - Пользовательские требования
 - Нефункциональные требования
 - Ограничения
 - Требования к качеству

Функциональные требования

- ▶ Бизнес-требования
 - Формулируются заказчиками
 - Описывают цели, которые требуется достичь с данной системой
- ▶ Требования пользователей
 - Какие задачи можно решить с помощью системы
- ▶ Собственно функциональные требования
 - Определяются функциональность, которую необходимо реализовать

Нефункциональные требования

- ▶ Требования к характеристикам качества
 - Требования к надежности
 - Требования к совместимости
 - Требования к эффективности
 - Требования к гибкости
 - Требования к эргономике
- ▶ Ограничения
 - Соответствия стандартам и правилам
 - Бюджет
 - Сроки
 - Предопределенные архитектурные решения
 - ...

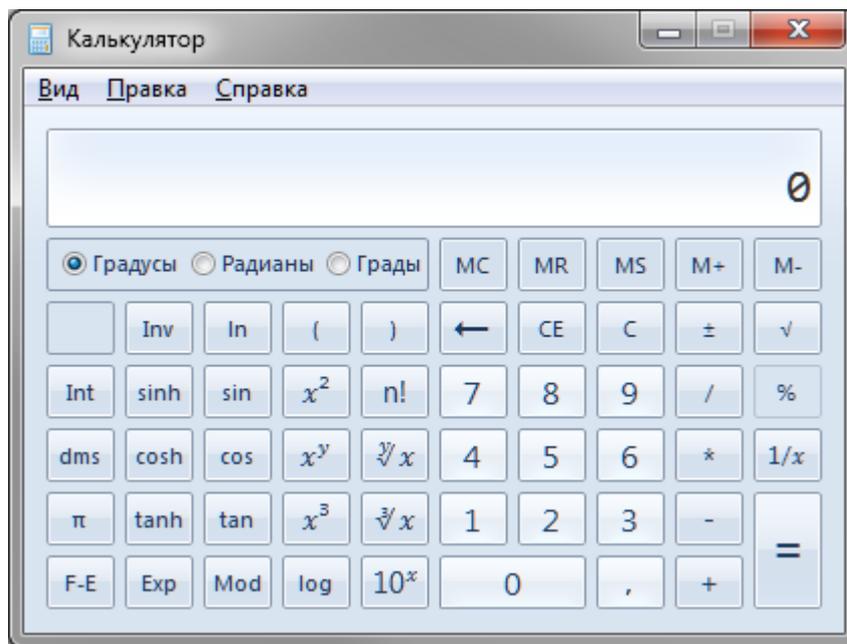
Ограничения

- ▶ Мы сделаем проект:
 - Быстро
 - Качественно
 - Недорого
- ▶ Выберите 2 из 3-х

Что не является требованиями?

- ▶ Детали архитектуры
- ▶ Детали реализации
- ▶ Сведения о планировании
- ▶ Сведения о тестировании
- ▶ Проектная информация:
 - Инфраструктура разработки
 - Процесс разработки
 - Команда разработки

Пример



2. Разработка требований

- ▶ Выявление требований
- ▶ Анализ требований

- ▶ Результат - спецификация требований

Выявления требований

- ▶ Заинтересованные лица
 - Заказчики
 - Менеджеры
 - Пользователи
 - Операторы
 - Менеджеры
 - ...
 - Разработчики
 - Служба поддержки
 - Другие лица
- ▶ ВАЖНО: заказчик ≠ пользователь

Выявления требований

- ▶ Планирование
 - Цели выявления требований
 - Стратегии и процессы выявления требований
 - Результаты усилий по выявлению требований
 - Оценки календарного плана и ресурсов
 - Риски, связанные с выявлением требований

Выявления требований

- ▶ Способы выявления требований
 - Исследования
 - Интервью
 - Семинар
 - Создание прототипов
 - Создание вариантов использования (Use Case)



Выявления требований

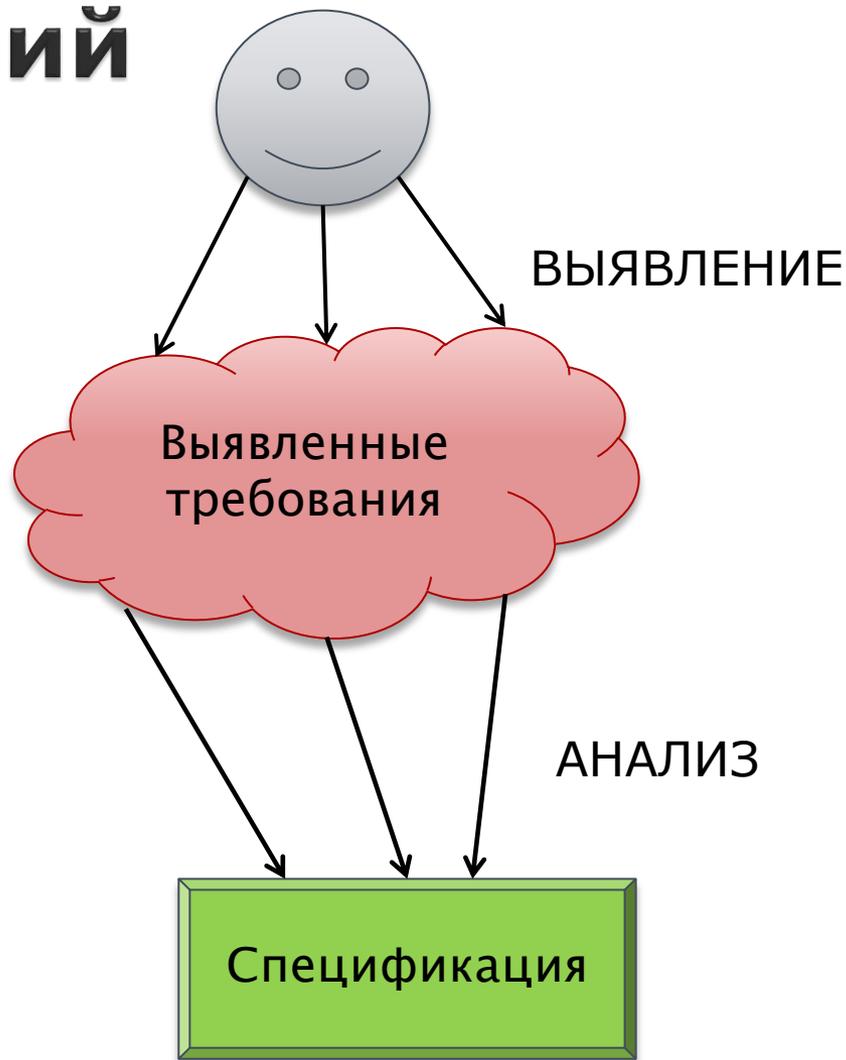
- ▶ Проблемы:
 - Формулирование требований
 - Терминология
 - Неявные допущения
 - Предвзятые решения

Выявления требований

- ▶ Проблемы определения требований:
 - Ожидания пользователей
 - Умение оценить противоречивые требования
 - Недостаточные требования
 - Умение понять требования пользователей

Анализ требований

- ▶ Выявление требований – расходящийся процесс, цель которого собрать как можно больше данных
- ▶ Анализ требований – сходящийся процесс:
 - Уточняет данные
 - Структурирует информацию
 - Устанавливает приоритеты
- ▶ Результат анализа – спецификация требований



Анализ требований

- ▶ Уточнение требований
 - Каждое требование должно быть максимально полным
 - Уточнение достигается путем повторных встреч с заинтересованными лицами
 - Не должно появляться много новых требований – иначе следует вернуться к выявлению
 - На этапе уточнения требования должны быть описаны *количественно*, а не качественно, как на этапе выявления

Анализ требований

▶ Приоритеты

- Необходимо отсортировать требования по важности и срочности
- Должны участвовать все заинтересованные лица проекта:
 - Заказчики
 - Пользователи
 - Разработчики
- Все требования *не могут быть* основными
- Приоритеты могут изменяться по мере развития проекта

Анализ требований

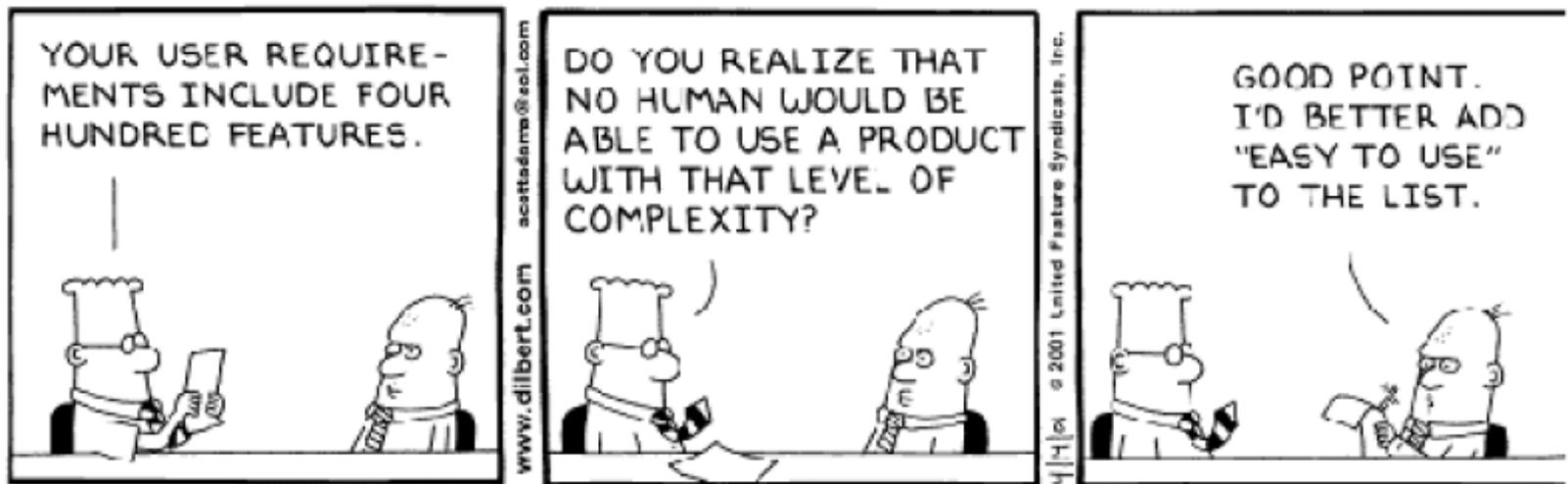
▶ Приоритеты

- Каждое требование относится к какой-либо качественной категории по важности:
 - Высокая, средняя, низкая
 - Обязан, должен бы, мог бы
 - Основной, полезный, желаемый
- Каждое требование относится к какой-либо качественной категории по срочности:
 - Прямо сейчас, чуть позже, когда-нибудь
 - Срочно, чуть позже, потом
- Сортируются по двум измерениям:

	Важно	Не важно
Срочно	Высший приоритет	Минимальный приоритет
Не срочно	Средний приоритет	Не стоит делать

Анализ требований

- ▶ Все ли требования собраны?



3. Документирование и организация требований

- ▶ Как документировать разные требования?
- ▶ Требования пользователей
 - варианты использования
- ▶ Бизнес-требования
 - документ о представлении/границах проекта
- ▶ Функциональные требования
 - спецификация требований к ПО

Организация требований

- ▶ Группирование требований
 - Требования объединяются в родственные группы
- ▶ Иерархическая структуризация требований
 - Подчинение
 - Уточнение



Способы документирования требований

- ▶ Документы на естественном языке
- ▶ Графические модели
 - Диаграммы
 - Графы
 - Схемы
 - Потoki
- ▶ Формальные спецификации

Типы документов

- ▶ Создаются все или некоторые из документов:
 - Состав и распределение работ
 - ***Спецификация требований***
 - Концепция эксплуатации
 - Начальный план разработки ПО
 - Критерии принятия работ

Состав и распределение работ

- ▶ Распределяет ответственности между заинтересованными сторонами проекта – задает правила игры:
 - Кто создает, что и когда
 - Кто тестирует, что, как и когда
 - Кто платит, за что и когда
 - Кто докладывает кому
 - Кто принимает/утверждает завершение работ или этапов
 - Кто, как и когда санкционирует изменения
 - И т.п.

Концепция эксплуатации

- ▶ Описание того, как система должна работать или будет использоваться
 - Какие функции будут использоваться и кем
 - Как эти функции будут использоваться
 - В каких условиях эти функции будут использоваться
 - Как будет происходить ввод/вывод данных
 - Как система взаимодействует с другими системами
- ▶ Этот документ задает основу для разработки вариантов использования

Начальный план разработки ПО

- ▶ Высокоуровневый и приблизительный план разработки
- ▶ Задаёт
 - Основные документы
 - Точки принятия решений
 - Поставляемые артефакты
 - Этапы работ и контрольные точки
 - Графики платежей

Спецификация требований

- ▶ Фундамент всего последующего планирования, проектирования, реализации проекта
- ▶ Основание для тестирования проекта
- ▶ Основание для документирования проекта
- ▶ Должна содержать ограничения проекта
- ▶ **НО:** не должна содержать деталей проектирования, реализации, тестирования и управления проектом
- ▶ Является исходным соглашением между заказчиком и разработчиком

Шаблоны спецификаций требований к ПО

- ▶ Существуют различные государственные, отраслевые и корпоративные стандарты
- ▶ Наиболее распространенные в России:
 - IEEE 830-1998 «Recommended Practice for Software Requirements Specification»
 - ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»
- ▶ Шаблон не должен являться догмой (если это не требование заказчика)
- ▶ Следует при необходимости модифицировать шаблон в соответствии с природой и потребностями проекта
- * Полезный документ: IEEE Guide for Developing System Requirements Specifications

Шаблон спецификации требований (IEEE 830-1998). Общая часть

1. Введение

- 1.1 Назначение
- 1.2 Область действия;
- 1.3 Определения, акронимы и сокращения;
- 1.4 Публикации;
- 1.5 Краткий обзор

2. Общее описание

- 2.1 Перспектива изделия
- 2.2 Функции изделия
- 2.3 Характеристики пользователей
- 2.4 Ограничения
- 2.5 Допущения и зависимости
- 2.6 Разделение требований

3. Специфические требования

- 3.1 Внешние интерфейсы
- 3.2 Функции системы
- 3.3 Требования к рабочим характеристикам
- 3.4. Логические требования к базе данных
- 3.5 Проектные ограничения
- 3.6 Атрибуты системы программного обеспечения (нефункциональные требования)

Шаблон раздела 3 SRS, (по режимам. v1)

3.1 Требования к внешним интерфейсам

3.1.1 Интерфейсы пользователя

3.1.2 Аппаратные интерфейсы

3.1.3 Интерфейсы программного обеспечения

3.1.4 Интерфейсы связи

3.2 Функциональные требования

3.2.1 Режим 1

3.2.1.1 Функц. требование 1.1

...

3.2.1.n. Функц. требование 1.n

...

...

3.2.2 Режим 2

...

3.2.m Режим m

...

3.3. Требования к рабочим характеристикам

3.4. Проектные ограничения

3.5. Атрибуты системы программного обеспечения

3.6. Другие требования

Шаблон раздела 3 SRS (по режимам. v2)

3.1 Функциональные требования

3.1.1 Режим 1

3.1.1.1 Внешние интерфейсы

3.1.1.1.1 Интерфейсы пользователя

3.1.1.1.2 Аппаратные интерфейсы

3.1.1.1.3 Интерфейсы программного обеспечения

3.1.1.1.4 Интерфейсы связи

3.1.1.2 Функциональные требования

3.1.1.2.1 Функциональное требование 1

...

3.1.1.2.n Функциональное требование n

3.1.2 Режим 2

...

3.1.m Режим m

3.2 Проектные ограничения

3.3 Атрибуты системы программного обеспечения

3.4 Другие требования

Шаблон раздела 3 SRS (по классам пользователей)

3.1 Внешние интерфейсы

3.1.1 Интерфейсы пользователя

3.1.2 Аппаратные интерфейсы

3.1.3 Интерфейсы программного обеспечения

3.1.4 Интерфейсы связи

3.2 Функциональные требования

3.2.1 Класс пользователей 1

3.2.1.1 Функция, требование 1.1

...

3.2.1.n Функция, требование 1.n

...

3.2.2 Класс пользователей 2

...

3.2.m Класс пользователей m

3.2.m.1 Функция, требование m.1

...

3.2.m.n Функция, требование m.n

...

3.3 Требования к рабочим характеристикам

3.4 Проектные ограничения

3.5 Атрибуты системы программного обеспечения

3.6 Другие требования

Шаблон раздела 3 SRS (по объектам)

3.1 Внешние интерфейсы

3.1.1 Интерфейсы
пользователя

3.1.2 Аппаратные
интерфейсы

3.1.3 Интерфейсы
программного обеспечения

3.1.4 Интерфейсы связи

3.2 Классы/объекты

3.2.1 Класс/объект 1

3.2.1.1 Атрибуты

3.2.1.1.1 Атрибут 1

...

3.2.1.1.n Атрибут n

...

3.2.1.2 Функции

3.2.1.2.1 Функция требование 1.1

...

3.2.1.2.m Функция требование 1.m

...

3.2.1.3 Сообщения

3.2.2 Класс/объект 1

...

3.3 Требования к рабочим
характеристикам

3.4 Проектные
ограничения

3.5 Атрибуты системы
программного обеспечения

3.6 Другие требования

Шаблон раздела 3 SRS (по свойствам)

3.1 Внешние интерфейсы

3.1.1 Интерфейсы пользователя

3.1.2 Аппаратные интерфейсы

3.1.3 Интерфейсы программного обеспечения

3.1.4 Интерфейсы связи

3.2 Свойства системы

3.2.1 Свойство системы 1

3.2.1.1 Назначение свойства

3.2.1.2 Последовательность стимулов/откликов

3.2.1.3 Ассоциированные ФТ

3.2.1.3.1. ФТ 1

...

3.2.1.3.1. ФТ 2

...

3.2.2 Свойство системы 2

...

3.2.m Свойство системы m

...

3.3 Требования к рабочим характеристикам

3.4 Проектные ограничения

3.5 Атрибуты системы программного обеспечения

3.6 Другие требования

Шаблон раздела 3 SRS (по функциональной иерархии)

3.1 Внешние интерфейсы

3.1.1 Интерфейсы пользователя

3.1.2 Аппаратные интерфейсы

3.1.3 Интерфейсы программного обеспечения

3.1.4 Интерфейсы связи

3.2 Функциональные требования

3.2.1 Информационные потоки

3.2.1.1 Схема потока данных 1

3.2.1.n Схема потока данных n

3.2.2 Описания процессов

3.2.2.1 Процесс 1

3.2.2.m Процесс m

3.2.3 Спецификации структуры данных

3.2.3.1 Структура 1

3.2.3.r Структура r

3.2.4 Словарь данных

3.2.4.1 Элемент данных 1

3.2.4.t Элемент данных t

3.3 Требования к рабочим характеристикам

3.4 Проектные ограничения

3.5 Атрибуты системы программного обеспечения

3.6 Другие требования

Шаблон раздела 3 SRS (множественная организация)

3.1 Внешние интерфейсы

3.1.1 Интерфейсы пользователя

3.1.2 Аппаратные интерфейсы

3.1.3 Интерфейсы программного обеспечения

3.1.4 Интерфейсы связи

3.2 Функциональные требования

3.2.1 Класс пользователей 1

3.2.1.1 Свойство 1.1

3.2.1.1.1 Назначение свойства

3.2.1.1.2 Последовательность стимулов/откликов

3.2.1.1.3 Ассоциированные ФТ

3.2.1.n Свойство 1.n

3.2.2 Класс пользователей 2

3.2.3 Класс пользователей m

3.3 Требования к рабочим характеристикам

3.4 Проектные ограничения

3.5 Атрибуты системы программного обеспечения

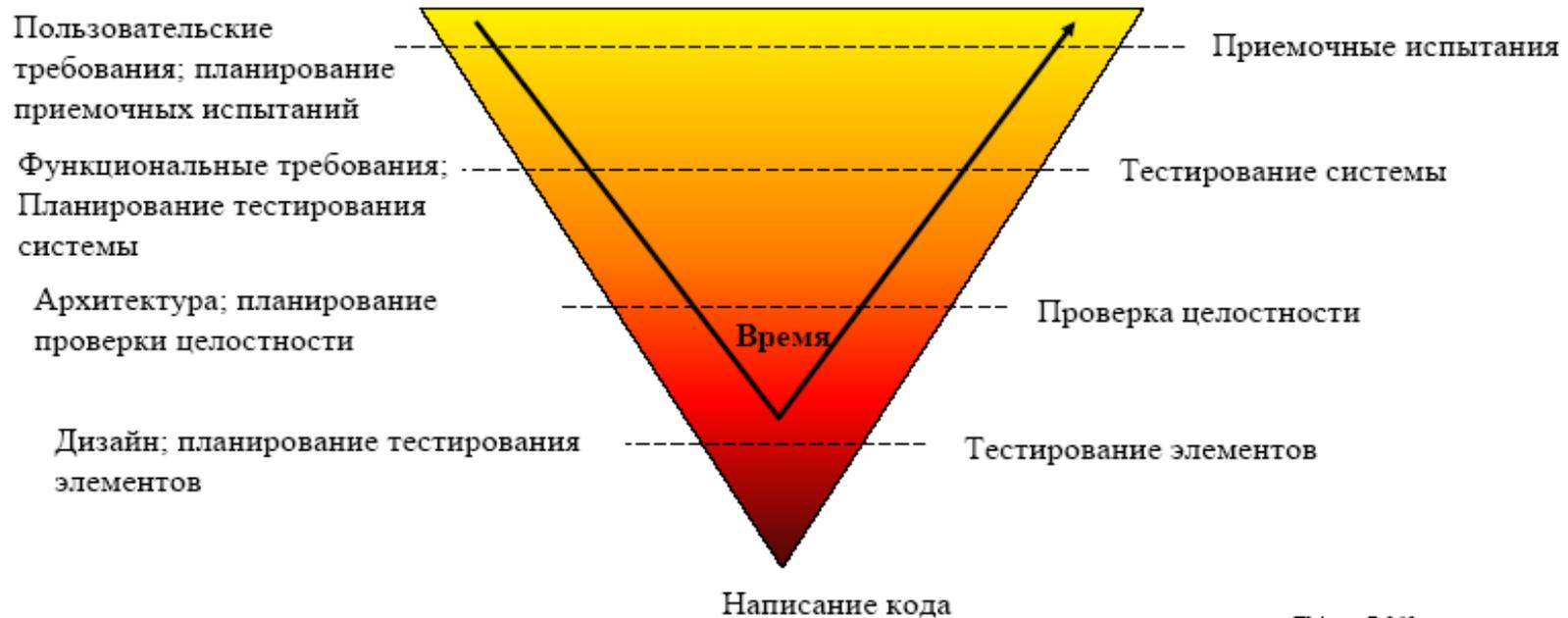
3.6 Другие требования

Критерии принятия работ

- ▶ Должны быть приняты всеми заинтересованными лицами
- ▶ Должны быть четкими и недвусмысленными
- ▶ Разделы методики принятия работы должны определяться количественными параметрами, а не качественными

Проверка правильности требований

► V-модель разработки ПО



Weigers P.260

Тестирование требований!

4. Изменение требований

- ▶ Причины изменения требований
- ▶ Условия возможности изменений
- ▶ Политика управления изменениями
- ▶ Анализ влияния изменения
- ▶ Принятие/непринятие изменений

Проблема изменения требований

Причины изменения требований

- ▶ **Заказчик**
 - Не понравилось после просмотра
 - Передумал
 - Забыл
- ▶ **РЫНОК**
 - Такой продукт уже не продать
 - Нужно выйти на рынок прямо сейчас, иначе этот продукт не продать
- ▶ **Разработка**
 - Неточное определение границ проекта
 - Требования неправильно поняты
 - Требования плохо определены
 - Требования не были поняты
 - Сработали архитектурные риски

Условия возможности изменения требований

- ▶ Водопадные стратегии – не возможно
- ▶ Инкрементные стратегии – возможно с некоторыми ограничениями
- ▶ Эволюционные стратегии - возможно

Политика управления изменениями

- ▶ Должен быть принят **процесс контроля за изменениями**
- ▶ Все изменения должны следовать процессу или не рассматриваться
- ▶ Для неутвержденных требований не выполняется никаких действий, кроме исследования осуществимости
- ▶ Все запросы на изменение должны быть одобрены советом по управлению изменениями
- ▶ Содержание запроса на изменение должно быть доступно всем заинтересованным лица проекта
- ▶ Начальный текст запроса должен быть неизменным
- ▶ Анализ воздействия должен проводиться для каждого изменения
- ▶ Каждое одобренное изменение (добавленное требование) должно прослеживаться до запроса на изменение
- ▶ Обоснование каждого одобрения на изменение должно быть задокументировано

Анализ влияния изменения

- ▶ Выявление последствий внесения изменения
- ▶ Определение всех сущностей (файлы, модели, артефакты, документы), которые нуждаются в модификации, если изменение будет принято
- ▶ Определение задач, необходимых для реализации изменения
- ▶ Оценка усилий для завершения этих задач
- ▶ Оценка нахождения этих задач на критическом пути проекта
- ▶ Оценка влияния на график работ
- ▶ Оценка влияния на стоимость
- ▶ Оценка приоритета изменения, учитывая
 - Достоинства
 - Недостатки
 - Затраты
 - Риски

Варианты решения на запрос об изменении требований

- ▶ Отложить низкоприоритетные требования
- ▶ Привлечь дополнительных сотрудников
- ▶ Организовать краткосрочную сверхурочную работу
- ▶ Изменить график работ
- ▶ Пожертвовать качеством
- ▶ ОТКАЗАТЬ!

5. Планирование и управление требованиями

▶ Цели:

- Контроль версий требований
- Контроль состояния требования
- Прослеживаемость требований
- Управление изменениями требований
- Совершенствование процессов управления

Управление требованиями

- ▶ *Управление изменениями*
 - *Предложение изменений*
 - *Анализ изменений*
 - *Принятие решений*
 - *Обновление требований*
 - *Обновление планов*
- ▶ **Контроль версий**
 - Определение схемы идентификации версий
 - Определение версий спецификаций требований
 - Определение версий отдельных требований
 - Контроль состояния требований
 - Определение состояния требований
 - Регистрация состояния требования
- ▶ **Прослеживание требований**
 - Определение связей с другими требованиями
 - Определение связей с другими элементами системы

Управление версиями требований

- ▶ Требования могут устаревать
- ▶ Требования могут быть противоречивыми
- ▶ Контроль версий документов
 - С помощью любой системы контроля версий
- ▶ Контроль версий требований
 - Создание начальных версий требований
 - Ведение истории изменений
 - Авторизованный доступ к изменениям требований

Управление состояниями требований

Состояние	Определение
Предложено	Требование было выставлено авторизованным источником
Одобрено	Требование было проанализировано и одобрено для определенной версии.
Реализовано	Код, реализующий требование, был написан
Проверено	Корректная функциональность донного требования была подтверждена версией продукта. Требование может быть прослежено до варианта тестирования.
Удалено	Ранее одобренное требование было исключено из базисного списка. Причина удаления – задокументирована
Отклонено	Предложенное требование – отклонено. Причина отклонения – задокументирована

Отслеживание состояний требований

- ▶ Показатель прогресса проекта
- ▶ Используется при анализе изменений
- ▶ Обосновывает некоторые решения, принятые во время разработки
- ▶ Обычно измеряется в процентах завершенности работ
 - Часто может вводить в заблуждение

Прослеживание требований

- ▶ Цели:
 - Получить подтверждение, что цели были реализованы
 - Убедиться, что требования были оттестированы
 - Иметь трассы всех требований от заказчика до тестовых случаев

Прослеживание требований



Матрица прослеживаемости требований

▶ Пример 1

Требование пользователя	Функциональное требование	Элемент дизайна	Часть кода	Тестовый сценарий
UC-16	Catalog.item.sort	Class Catalog	Catalog.sort()	Sort.7 Sort.8
UC-29	2.7.4	Class Order		Order.5
UC-37	SR 2.1	Class Encrypt		Security.31 Security.32 Security.33

Матрица прослеживания требований

▶ Пример 2

Функциональное требование	Сценарии тестирования			
	TC1	TC2	TC3	TC4
FC1	*			
FC2		*		*
FC3		*		
FC4	*			
FC5			*	
FC6		*	*	

Инженерия требований.

Резюме

- ▶ Выявление требований
 - Исследования
 - Интервью
 - Семинар
 - Создание прототипов
 - Use Case
- ▶ Анализ требований
 - Уточнение требований
 - Структуризация
 - Установка приоритетов
- ▶ Документирование требований
 - Состав и распределение работ
 - Спецификация требований
 - Концепция эксплуатации
 - Начальный план разработки ПО
 - Критерии принятия работ
- ▶ Управление требованиями
 - Изменения требований
 - Предложение изменений
 - Анализ изменений
 - Принятие решений
 - Обновление требований
 - Обновление планов
 - Контроль версий требований
 - Прослеживание требований

Программные средства управления требованиями

- ▶ Существует более 40 средств управления требованиями
- ▶ Наиболее функциональные:
 - IBM Rational DOORS (бывшее Telelogic DOORS)
 - <http://www.ibm.com/software/awdtools/doors/>
 - IBM Rational Requisite Pro
 - <http://www.ibm.com/software/awdtools/reqpro/>
 - Borland
 - Caliber DefineIT
 - <http://www.borland.com/us/products/caliber/defineit.html>
 - Caliber RM
 - <http://www.borland.com/us/products/caliber/rm.html>
 - Compuware Optimal Trace

Функции инструментальных средств управления требованиями

1. Захват/идентификация требований
2. Выделение структуры и организация требований
3. Трассировка требований
4. Управление конфигурациями
5. Формирование отчетов
6. Групповая работа
7. Интерфейсы к другим средствам
8. Системное окружение
9. Пользовательский интерфейс
10. Поддержка