

Работа с требованиями

NX Bootcamp

Требования - ISO 12207

Требования к программным средствам – это то что разработчик должен установить и документально оформить:

- функциональные и технические требования;
- требования к внешним интерфейсам программного объекта;
- квалификационные требования;
- требования безопасности;
- и т.д.



Требования - IEEE 830

Требования к программной системе – это:

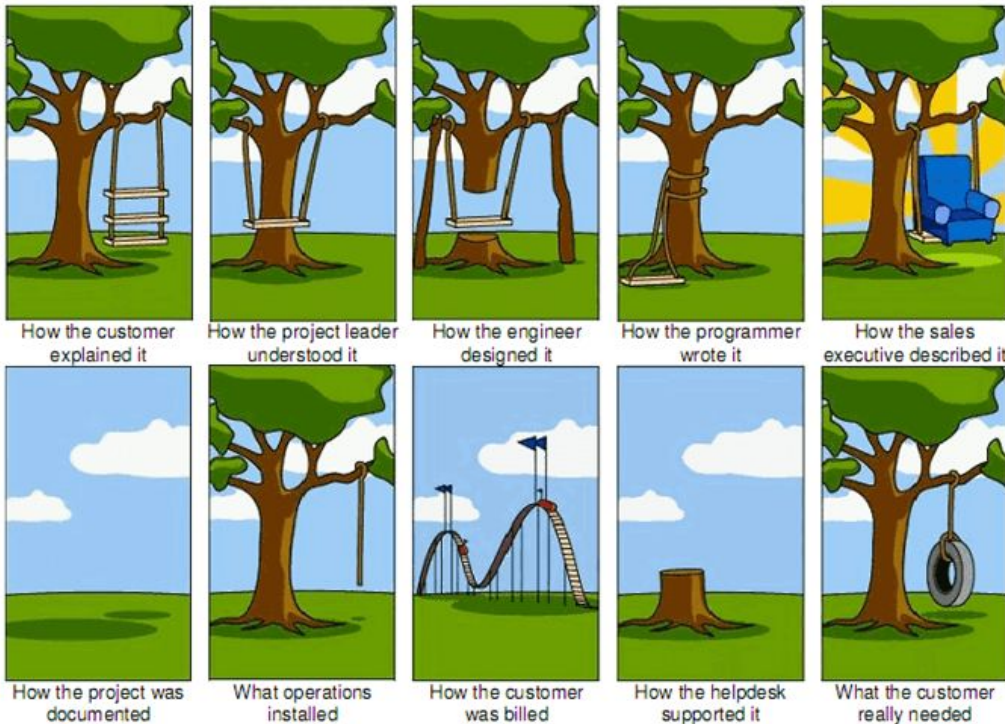
- Функциональность, необходимая пользователю для решения проблемы или достижения цели.
- Функциональность, которая должна быть получена (достигнута) системой или ее компонентами для соответствия контракту, стандарту, спецификации или другим формальным документам.
- Документальное представление пп. 1 – 2.

Стандарты определяют только функциональные требования.



Требования

Требование - это документированное условие или способность продукта, необходимые пользователю, чтобы решить проблему или достичь бизнес-цель.



Группировка требований

1. По фичам (высокоуровневым свойствам)
2. По ролям пользователей и смежным системами
3. По этапам реализации (1, 2, 3, ...)
4. По режимам работы системы
5. По объектам
6. Другие

Кодирование требований

С использованием буквенно-цифровых кодов:

- FR-001
- FR-001/01
- FR_ORD-001.05

С использованием смысловых алиасов:

- Заказ.Создание
- Заказ.Редактирование
- Заказ.СписокЗаказов

Характеристики требований

Характеристика	Описание
Единичность	Требование описывает одно и только одно свойство.
Завершенность	Требование полностью определено в одном месте документации.
Последовательность	Требование не противоречит другим требованиям и ограничениям системы.
Атомарность	Требование атомарно, то есть оно не может быть разбито на ряд более детальных требований без потери завершенности.

Характеристики требований

Характеристика	Описание
Отслеживаемость (Трассируемость)	Требование зафиксировано в документации и можно понять его источник.
Актуальность	Требование не устарело.
Выполнимость	Требование может быть реализовано в рамках проекта.
Недвусмысленность	Требование определено без обращения к техническому жаргону, акронимам и другим скрытым формулировкам. Оно выражает объективные факты, не субъективные мнения. Возможна одна и только одна интерпретация.

Характеристики требований

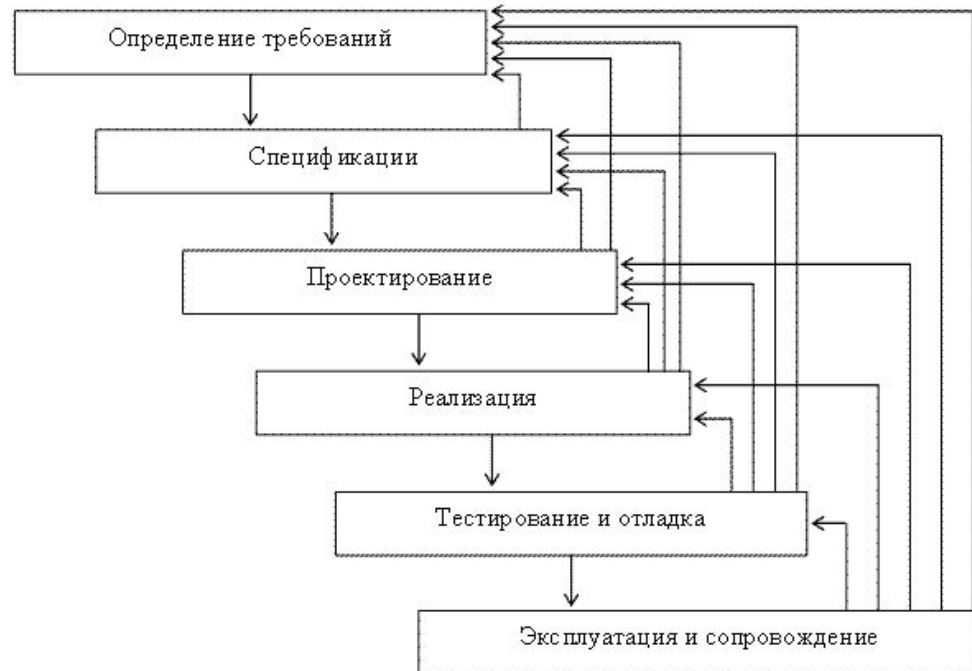
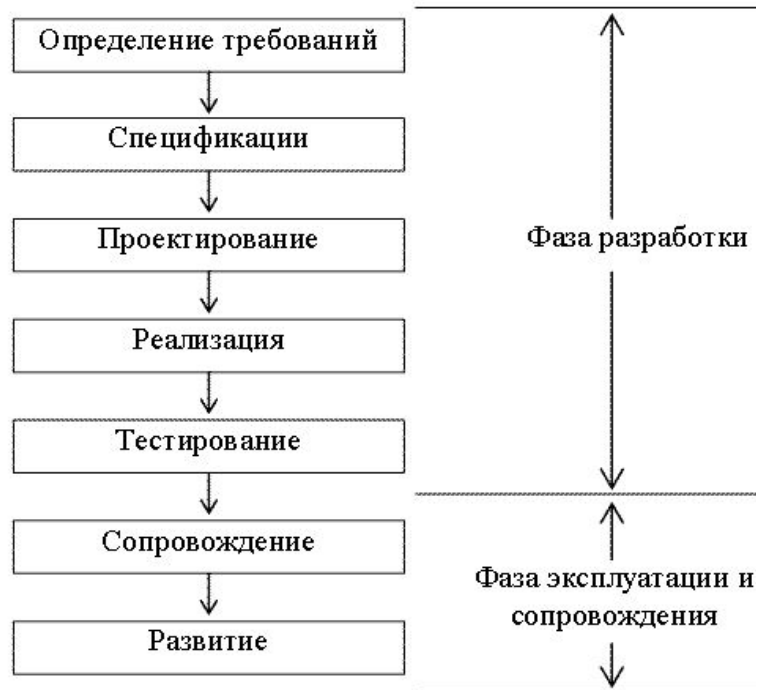
Характеристика	Описание
Обязательность	Требование представляет определенную заинтересованным лицом характеристику, отсутствие которой приводит к неполноценности решения.
Проверяемость (тестируемость)	Реализованность требования может быть проверена с помощью одного из четырех возможных методов: осмотр, демонстрация, тест или анализ.
Полнота	Полнота требований – свойство, означающее, что совокупность артефактов, описывающих требования, исчерпывающим образом описывает все то, что требуется от разрабатываемой системы.

Жизненный цикл требований

Жизненный цикл ПО

Жизненный цикл программного обеспечения (Software Life Cycle Model) — это период времени, который начинается с момента принятия решения о создании программного продукта и заканчивается в момент его полного изъятия из эксплуатации.

Жизненный цикл требований



Состояния требований

Состояние	Определение
Предложено (Proposed)	Требование запрошено уполномоченным лицом
Разработка (In Progress)	Аналитик активно работает над требованием
Подготовлено (Drafted)	Написана начальная версия требования
Одобрено (Approved)	Требование проанализировано, его влияние на проект просчитано, и оно было размещено в базовой версии ПО. Ключевые заинтересованные лица согласились с этим требованием, а разработчики ПО обязались реализовать его

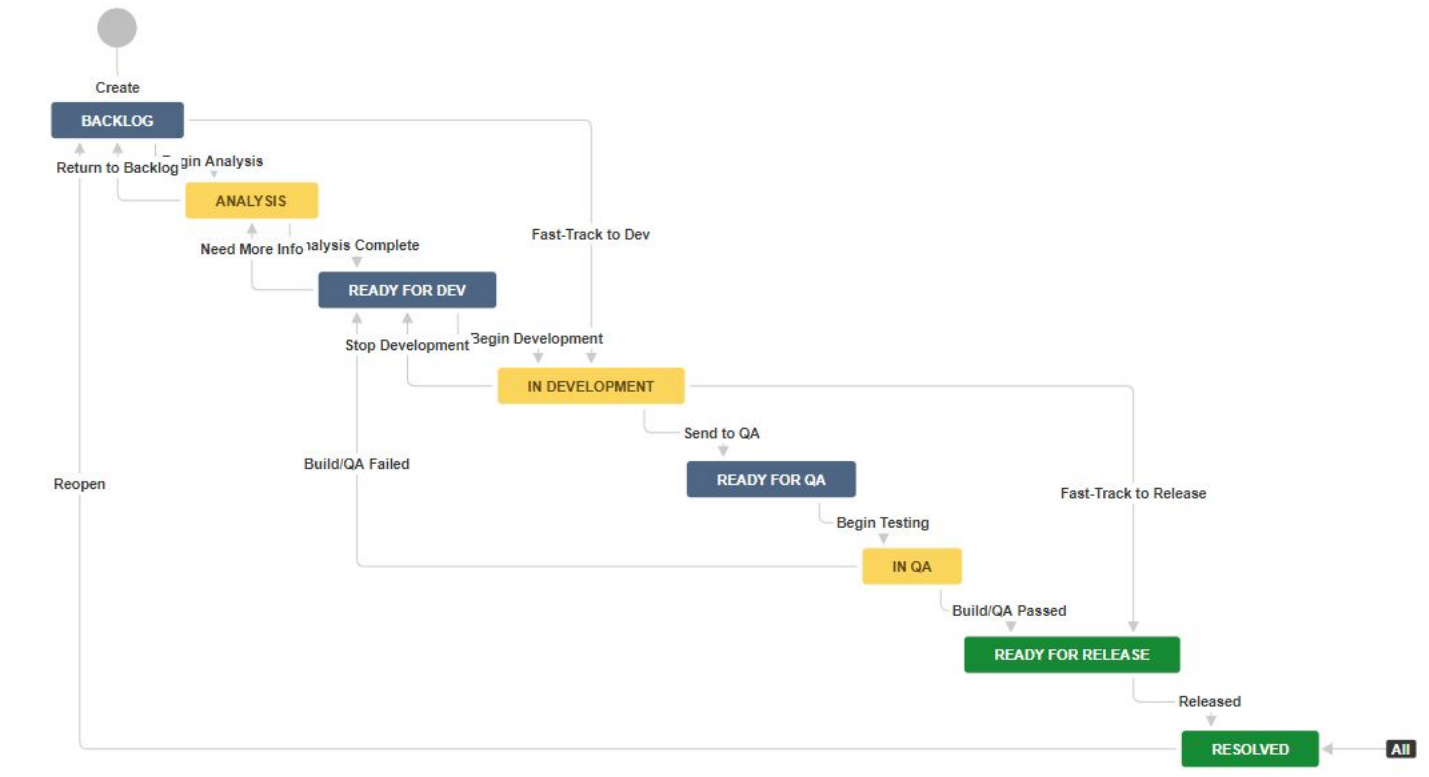
Состояния требований

Состояние	Определение
Реализовано (Implemented)	Код, реализующий требование, разработан, написан и протестирован. Определена связь требования с соответствующими элементами дизайна и кода.
Проверено (Verified)	Требование удовлетворяет критериям приемки, подтверждено корректное функционирование реализованного требования. Определена связь требования с соответствующими вариантами тестирования. Требование считается завершенным

Состояния требований

Состояние	Определение
Отложено (Deferred)	Одобренное требование запланировано для реализации в более позднем выпуске
Удалено (Deleted)	Утвержденное требование удалено из базовой версии. Описаны причины удаления и указан тот, кто принял это решение
Отклонено (Rejected)	Требование предложено, но не запланировано для реализации ни в одном из будущих выпусков. Указаны причины отклонения требования и тот, кто принял это решение

Состояния требований



Методы сбора требований

Методы сбора требований

- интервью
- анкеты
- наблюдение
- погружение
- раскадровка
- прототипирование
- анализ документации
- анализ обращений в техническую поддержку
- анализ ПО аналога

Наблюдение

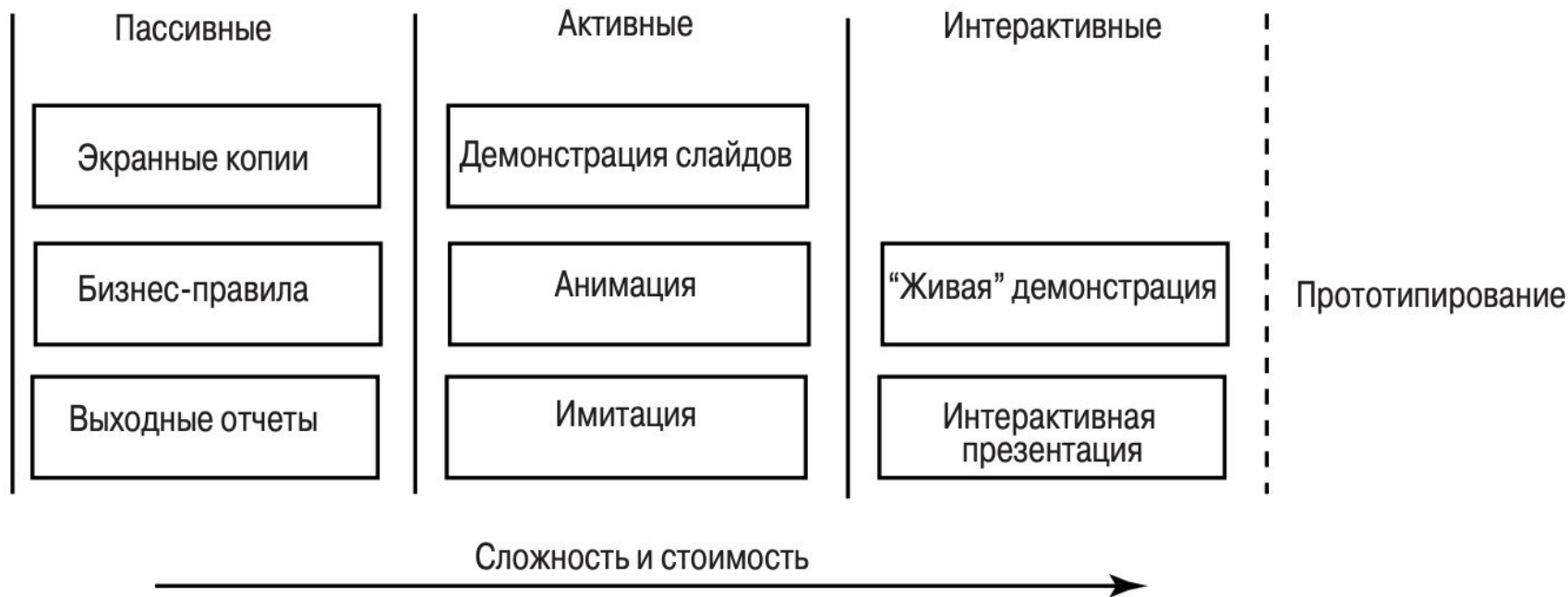
- вспомогательный метод
- применяется для сбора информации
- применяется для верификации информации от экспертов

Проблемы:

- Большие затраты времени
- Искаженная информация
- Присутствие стороннего наблюдателя, как правило, увеличивает производительность



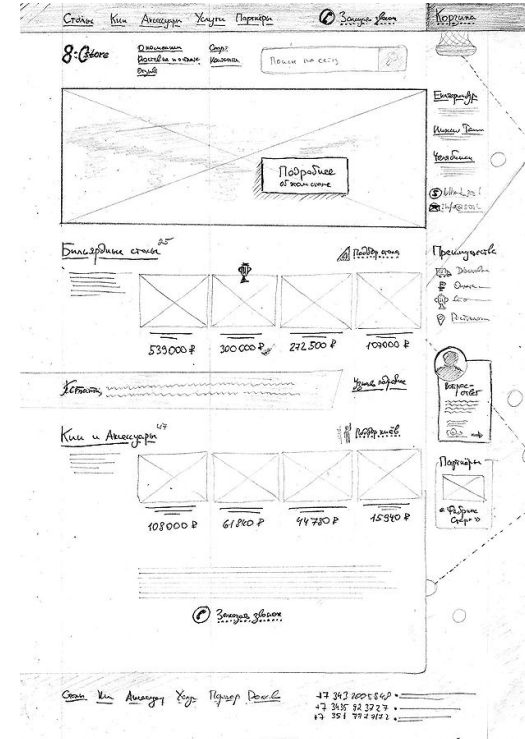
Раскадровка



Прототипирование

Виды прототипов:

- Отбрасываемые – нужен для понимания осуществимости проекта. По окончании прототип выбрасывается;
- Эволюционирующие – прототип реализуется в той же архитектуре, что и рабочий проект, так что можно прототип развивать дальше;

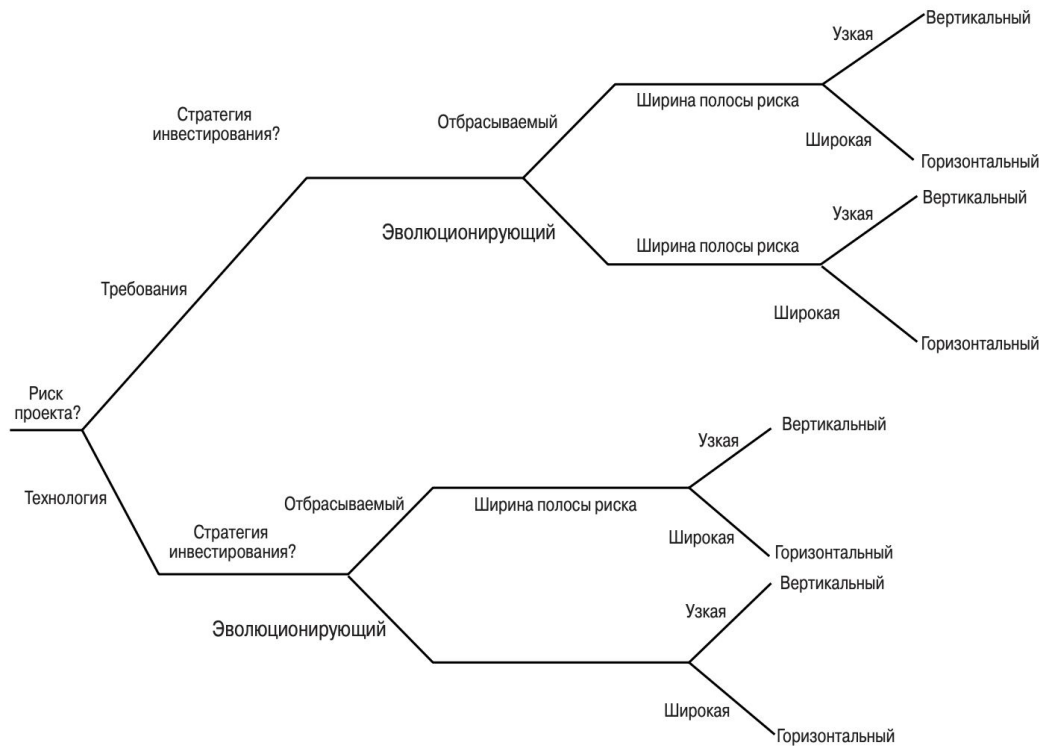


Прототипирование

Операционные:

- Вертикальные – создает довольно мало требований, но делает это качественно;
- Горизонтальные – создаем широкий спектр функциональных возможностей системы;
- Пользовательские интерфейсы – создание интерфейса пользователя, а не бизнес-логики приложения;
- Алгоритмические

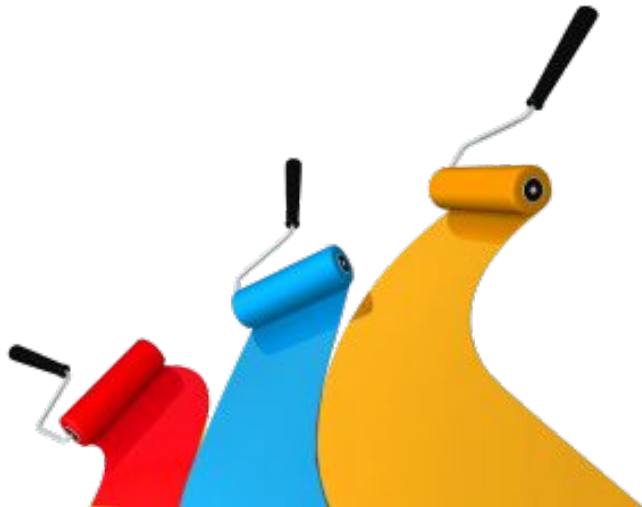
Прототипирование



Виды требований

Виды классификаций требований

- По типу требований
- По уровню требований
- По уровню целей
- FURPS+ подход



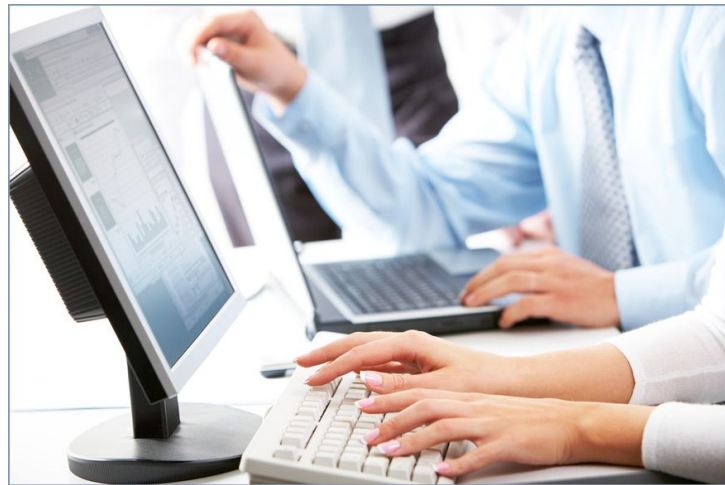
Классификация требований по типу

Функциональные требования:

Описывают непосредственно функционал, реализуемый Системой.

Пример:

«По клику на кнопке <Кнопка А> на форме <Форма Б> должно отображаться модальное диалоговое окно, содержащее <Содержание окна>».



Классификация требований по типу

Нефункциональные требования

Описывают характеристики системы и ее окружения, а так же накладываемые ограничения.

Пример:

ограничения на поддерживаемые разрабатываемой Системой аппаратные платформы и операционные системы, а так же требования к производительности, к безопасности и т.д.



Классификация требований по уровню

Бизнес-требования

Самые высокоуровневые требования, которые определяют цели создания ПО.

Пример:

достижение 20%-го сокращения издержек или
повышение качества управления (за счет
возможности оперативного формирования отчетности).

Данные требования описываются на уровне
концепции проекта.



Классификация требований по уровню

Требования пользователей

Определяют набор пользовательских задач, которые должна решать Система, с описанием сценариев решения данных задач.

Представляются в виде перечисления вариантов использования системы и взаимосвязей между ними.

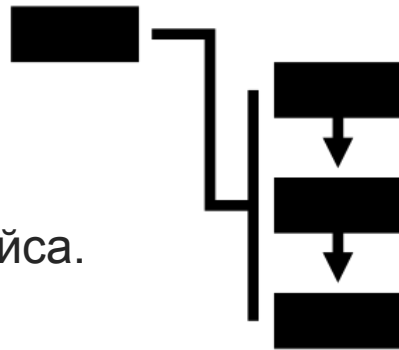


Классификация требований по уровню

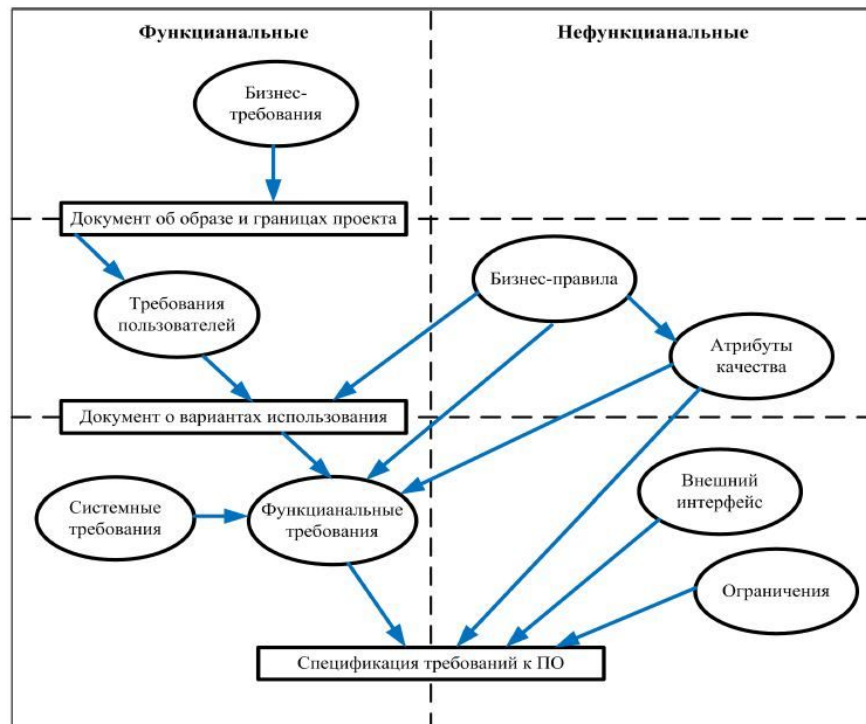
Функциональные требования

Детально описывают все элементы функционала, который должен быть непосредственно реализован в системе, чтобы обеспечить возможность выполнения всех сценариев использования, описанных в Требованиях пользователей.

Функциональные требования являются наиболее детализированными. Они описывают, в том числе, входные/выходные данные и их проверки, алгоритмы обработки данных и элементы пользовательского интерфейса.

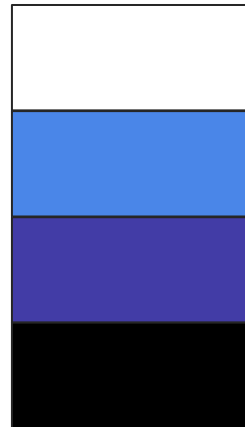


Взаимосвязь различных видов требований

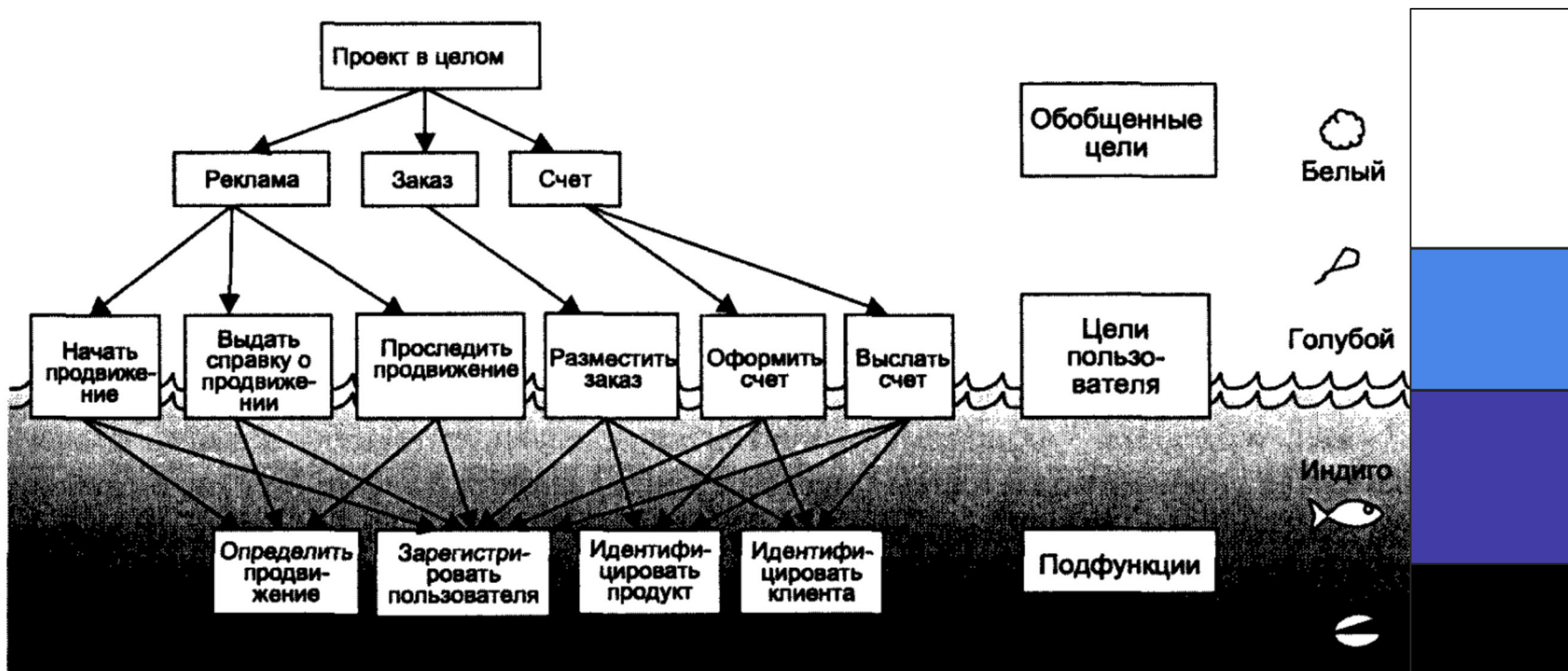


Классификация требований по уровню целей

- Обобщенный уровень
(белый, облако или воздушный змей)
- Цели пользователей
(голубые, уровня моря)
- Подфункции
(индиго или черный, подводный или моллюск)



Классификация требований по уровню целей



Классификация требований FURPS+

1. Functionality, функциональность
2. Usability, удобство использования
3. Reliability, надежность
4. Performance, производительность
5. Supportability, поддерживаемость
6. + возможные ограничения:
 - ограничения проектирования, design
 - ограничения разработки, implementation
 - ограничения на интерфейсы, interface
 - физические ограничения, physical



Роберт Грэйди (Robert Grady)

Реестр функциональных требований

Реестр функциональных требований

Реестр функциональных требований необходим для оценки масштаба проекта, отслеживания полноты их реализации и стадии развития проекта.

В зависимости от способа документирования требований могут использоваться следующие виды реестров:

- Коллекции User Story
- Реестр функциональных требований
- Реестр UseCase
- Реестр пользовательских сценариев

Проверка полноты функциональных требований

Привязка к шагам бизнес-процесса.

Проверка требований, описывающих действия с сущностью:

Стандартные:

- создание;
- просмотр;
- редактирование;
- удаление.

Расширенные:

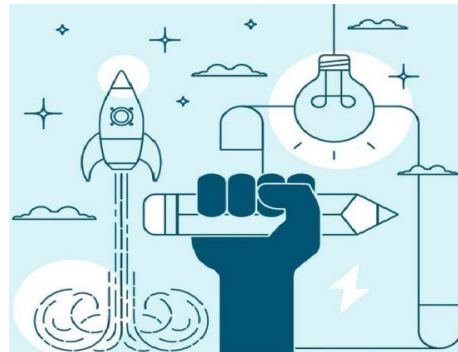
- поиск;
- отображение списка объектов;
- отображение истории.

Специфичные:

- слияние дубликатов;
- откат изменений;
- т.д.

Практическое задание 016

Составить реестр UseCase, на основе диаграммы вариантов использования (практическое задание 006).



Код Use case	Главное действующее лицо	Use case	Приоритет
UC-0.1	Работодатель, Соискатель, Преподаватель	Авторизация в системе	
UC-0.2	Работодатель, Соискатель, Преподаватель	Регистрация в системе	
UC-1	Работодатель	Заккрыть вакансию	
UC-1.1	Работодатель	Просмотреть список резюме	
UC-1.2	Работодатель	Просмотреть рейтинги соискателей	

Thank you

Join us!

VK



WhatsApp



Telegram



nxbootcamp@nexign.com