**Требования**

Что такое требования к П О ? - прежде чем разрабатывать какой-то программный продукт мы должны понимать а что мы вообще хотим разрабатывать, какие должны быть функции у данного продукта, как оно должно внешне выглядеть, как оно будет эксплуатироваться пользователем, что будет если пользователь введёт не корректные данные <- - это всё **требования** к продукту.

**Требования делятся на 3 группы:**

**1)Бизнес треб**. выражают цель, ради которой разрабатывается продукт (зачем вообще он нужен, какая от него ожидается польза, как заказчик с его помощью будет получать прибыль). Результатом выявления требований на этом уровне является общее видение — документ, который, как правило, представлен простым текстом и таблицами. Здесь нет детализации поведения системы и иных технических характеристик, но вполне могут быть определены приоритеты решаемых бизнес-задач, риски и т.п.

Несколько простых, изолированных от контекста и друг от друга примеров бизнес-требований:

• Нужен инструмент, в реальном времени отображающий наиболее выгодный курс покупки и продажи валюты.

• Необходимо в два-три раза повысить количество заявок, обрабатываемых одним оператором за смену.

• Нужно автоматизировать процесс выписки товарно-транспортных накладных на основе договоров.

**2)Пользовательские треб.** описывают задачи, которые пользователь может выполнять с помощью разрабатываемой системы (реакцию системы на действия пользователя, сценарии работы пользователя). Поскольку здесь уже появляется описание поведения системы, требования этого уровня могут быть использованы для оценки объёма работ, стоимости проекта, времени разработки и т.д. Пользовательские требования оформляются в виде вариантов использования (**use cases**), пользовательских историй (**user stories**), пользовательских сценариев (**user scenarios**).

Несколько простых, изолированных от контекста и друг от друга примеров пользовательских требований:

• При первом входе пользователя в систему должно отображаться лицензионное соглашение.

• Администратор должен иметь возможность просматривать список всех пользователей, работающих в данный момент в системе.

• При первом сохранении новой статьи система должна выдавать запрос на сохранение в виде черновика или публикацию.

**3)Требования непосредственно к продукту** их может быть много я выписал только основные

**А)Функциональные требования** (functional requirements) описывают поведение системы, т.е. её действия (вычисления, преобразования, проверки, обработку и т.д.). В контексте проектирования функциональные требования в основном влияют на дизайн системы. Стоит помнить, что к поведению системы относится не только то, что система должна делать, но и то, что она не должна делать (например: «приложение не должно выгружать из оперативной памяти фоновые документы в течение 30 минут с момента выполнения с ними последней операции»).

Несколько простых, изолированных от контекста и друг от друга примеров функциональных требований:

• В процессе инсталляции приложение должно проверять остаток свободного места на целевом носителе.

• Система должна автоматически выполнять резервное копирование данных ежедневно в указанный момент времени.

• Электронный адрес пользователя, вводимый при регистрации, должен быть проверен на соответствие требованиям RFC822.

**Б)Нефункциональные требования** (non-functional requirements) описывают свойства системы (удобство использования, безопасность, надёжность, расширяемость и т.д.), которыми она должна обладать при реализации своего поведения. Здесь приводится более техническое и детальное описание атрибутов качества. В контексте проектирования нефункциональные требования в основном влияют на архитектуру системы. Несколько простых, изолированных от контекста и друг от друга примеров нефункциональных требований:

• При одновременной непрерывной работе с системой 1000 пользователей, минимальное время между возникновением сбоев должно быть более или равно 100 часов.

• Ни при каких условиях общий объём используемой приложением памяти не может превышать 2 ГБ.

• Размер шрифта для любой надписи на экране должен поддерживать настройку в диапазоне от 5 до 15 пунктов.

**Б1) Атрибуты качества** расширяют собой нефункциональные требования и на уровне пользовательских требований могут быть представлены в виде описания ключевых для проекта показателей качества (свойств продукта, не связанных с функциональностью, но являющихся важными для достижения целей создания продукта — производительность, масштабируемость, восстанавливаемость). Атрибутов качества очень много, но для любого проекта реально важными является лишь некоторое их подмножество.

Несколько простых, изолированных от контекста и друг от друга примеров атрибутов качества

• Максимальное время готовности системы к выполнению новой команды после отмены предыдущей не может превышать одну секунду.

• Внесённые в текст статьи изменения не должны быть утеряны при нарушении соединения между клиентом и сервером.

• Приложение должно поддерживать добавление произвольного количества не иероглифических языков интерфейса.

**В)Требования к интерфейсам** (external interfaces requirements) описывают особенности взаимодействия разрабатываемой системы с другими системами и операционной средой.

Несколько простых, изолированных от контекста и друг от друга примеров требований к интерфейсам:

• Обмен данными между клиентской и серверной частями приложения при осуществлении фоновых AJAX-запросов должен быть реализован в формате JSON.

• Протоколирование событий должно вестись в журнале событий операционной системы.

**Г)Требования к данным** (data requirements) описывают структуры данных (и сами данные), являющиеся неотъемлемой частью разрабатываемой системы. Часто сюда относят описание базы данных и особенностей её использования.

Несколько простых, изолированных от контекста и друг от друга примеров требований к данным:

• Все данные системы, за исключением пользовательских документов, должны храниться в БД под управлением СУБД MySQL, пользовательские документы должны храниться в БД под управлением СУБД MongoDB.

• Информация о кассовых транзакциях за текущий месяц должна храниться в операционной таблице, а по завершении месяца переноситься в архивную.

• Для ускорения операций поиска по тексту статей и обзоров должны быть предусмотрены полнотекстовые индексы на соответствующих полях таблиц



**Спецификация требований** (software requirements specification) (спека) объединяет в себе описание всех требований уровня продукта и может представлять собой весьма объёмный документ

Сбором требований занимается аналитик который общается с заказчикам и пытается понять какие функции должны быть у приложения. Его задача понять что хочет заказчик и описать это в требованиях понятных для команды. В соответствии требованиям разработчики создадут приложения. Задача тестировщика - убедится что программа работает в соответствии с требованиями. И если есть несоответствие то составляется баг репорт и отправляется команде разработчиков.

В том числе требования позволяют:

1)Позволяют понять, что и с соблюдением каких условий система должна делать.

2)Предоставляют возможность оценить масштаб изменений и управлять изменениями.

3)Являются основой для формирования плана проекта (в том числе плана тестирования).

4)Помогают предотвращать или разрешать конфликтные ситуации.

5)Упрощают расстановку приоритетов в наборе задач.

5)Позволяют объективно оценить степень прогресса в разработке проекта.

Вне зависимости от того, какая модель разработки ПО используется на проекте, чем позже будет обнаружена проблема, тем сложнее и дороже будет её решение. А в самом начале («водопада», «спуска по букве v», «итерации», «витка спирали») идёт планирование и работа с требованиями. Если проблема в требованиях будет выяснена на этой стадии, её решение может свестись к исправлению пары слов в тексте, в то время как недоработка, вызванная пропущенной проблемой в требованиях и обнаруженная на стадии эксплуатации, может даже полностью уничтожить проект.

