

Петрозаводский государственный университет
Институт Математики и информационных технологий
Кафедра Информатики и математического обеспечения

Формирование и анализ требований.

Лектор:

к.т.н., доцент Богоявленский Ю. А.
ybgv@cs.karelia.ru



Разделы презентации

- Определение, категории и виды требований
- Формирование, анализ, аттестация требований
- Процесс разработки требований
- Шаблон процесса создания требований
- Приемы формулирования требований
- Документ о концепции и границах

Определение, категории и виды требований. Содержание

- Определения и категории представление требований
 - ▶ Определение и примеры
 - ▶ Категории представления
- Спецификация требований
- Виды требований
 - ▶ Функциональные требования, примеры
 - ▶ Нефункциональные требования, примеры
- Формы записи требований

Требования к программному обеспечению: Определение

Требования (requirements) — описание функциональных возможностей и ограничений, накладываемых на создаваемую программную систему — ЧТО система должна делать, но не КАК добиться выполнения этих функций.



Требования к программному обеспечению: Пример

Требование к банковской системе

Требование

Система должна предоставить клиенту возможности выполнения следующих операций над его счетом: просмотр, снятие денег, добавление денег.

НЕ требование

Информация банковского счета должна храниться в виде трех таблиц СУБД MySQL.

Требования к программному обеспечению: Категории представления

Различают две категории представления требований:

- *Требования заказчика (первичные требования)* документируют желания и потребности заказчика и пишутся на языке, понятном заказчику.
- *Требования разработчика (детальные требования)* документируют требования в специальной, структурированной форме, они детализированы по отношению к первичным требованиям.

Сбор (формирование) требований — работа по созданию *первичных требований*, проводящаяся на этапе подготовки жизненного цикла разработки.

Анализ требований — работа по созданию *детальных требований*, проводящаяся на этапе моделирования жизненного цикла разработки.

Требования к программному обеспечению: Категории представления

Требования заказчика

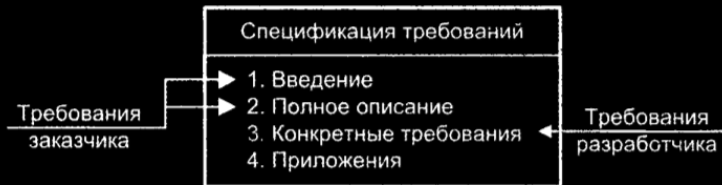
1. ПО должно обеспечить средства для ввода и сохранения разнообразных данных абонента-пользователя.

Требования разработчика

- 1.1. Пользователь должен иметь возможность определять тип вводимых данных.
- 1.2. Для каждого типа данных должно иметься соответствующее средство, обеспечивающее ввод и сохранение элемента данных этого типа.
- 1.3. Каждый тип данных должен представляться соответствующей пиктограммой на дисплее пользователя.

Спецификация требований

- Требования заказчика являются первичным описанием на естественном языке функций, выполняемых системой, и ограничений, накладываемых на нее. Они помещаются в *системную спецификацию*.
- Требования разработчика содержат детализированное описание функций и ограничений системы, оформляемое в виде *спецификации анализа*. Она служит основой для заключения контракта между покупателем и разработчиками.



Виды требований

- *Функциональные требования* описывают поведение системы и сервисы (функции), которые она должна выполнять. При этом исходят из всестороннего анализа проблемной (предметной) области. Рассматриваются разнообразные варианты поведения, определяемые различными данными и состояниями внешней среды.
- *Нефункциональные требования* относятся к характеристикам системы и ее внешнего окружения. Дополнительно могут перечисляться ограничения, накладываемые на действия и функции системы, а также на условия разработки (ограничения по времени, ограничения на организацию проекта, стандарты и т. д.).

Виды требований: Функциональные требования, пример

Система управления летательного аппарата

- 1 Система должна измерять проекции скорости (по трем каналам), вычислять отклонения от программных значений вектора скорости и радиус-вектора, а затем формировать управляющие воздействия, посылая их на исполнительные органы летательного аппарата (ЛА).
- 2 Система должна обеспечивать угловую стабилизацию движения ЛА: измерение углов (по трем каналам), вычисление отклонений углов от программных значений, формирование управляющих воздействий, распределение управляющих воздействий между исполнительными органами.
- 3 Каждая программа автопилота должна снабжаться уникальным идентификатором, который записывается в «черный ящик» вместе с параметрами полета.

Формы записи требований

■ Требования должны быть:

- ▶ ясными (не допускать двоякого толкования, приводящего к искажению смысла пожеланий заказчика);
- ▶ согласованными (не содержать противоречивых и взаимоисключающих утверждений);
- ▶ полными (определять всю требуемую функциональность системы).



Требования Заказчика



Согласовано с разработчиком



Техническое задание



Результат разработки



Сдано Заказчику



Осталось после опытной эксплуатации



Что было нужно на самом деле

Виды требований: Нефункциональные требования

Выделяют три группы нефункциональных требований:

- *Требования к программной системе*, которые описывают свойства и характеристики системы.
- *Организационные требования*, которые отображают вопросы работы и организации взаимодействия заказчика и разработчика.
- *Внешние требования*, которые учитывают факторы внешней среды.

Виды требований: Нефункциональные требования, пример

Нефункциональные требования к производительности

Цикл регулирования скорости летательного аппарата должен укладываться в 64 мс.

Требования к производительности:

- определяют *временные ограничения*, которые должны быть выполнены в системе.
- являются важной частью систем, работающих в *реальном времени*, в которых действия должны уложиться в определенные временные рамки.

Виды требований: Нефункциональные требования, пример

Нефункциональные требования к надежности

Система управления микроклиматом оранжереи должна давать не более двух ошибок в месяц.

Нефункциональные требования к доступности

Система продажи авиабилетов должна быть доступна пользователям 24 часа в сутки. Она может быть недоступна (находиться в состоянии профилактики) в течение 10 минут за 30-дневный период.

Виды требований: Нефункциональные требования, пример

Нефункциональные требования ограничений

- Система управления крылатой ракетой (СУ КР) должна рассчитывать координаты цели с точностью до трех метров.
- СУ КР должна быть разработана на языке Ada 2005.
- Система управления крылатой ракетой должна работать на компьютерах Agat 1415 с расширением емкости оперативной памяти до 4 Гбайт.

Нефункциональные интерфейсные требования

Для передачи сообщений в систему телеметрии используется строковый формат `out_tm <code>`, где `<code>` — двухбайтовый код из таблицы посылок `Tajna_321b`.

Формирование, анализ, аттестация требований.

Содержание

■ Формирование требований

- ▶ Шаги процесса формирования требований
 - Проверка требований

■ Анализ требований

- ▶ Шаг 1. Организация первичных требований
- ▶ Шаг 2. Преобразование первичных требований в детальные требования
- ▶ Шаг 3. Аттестация детальных требований.
 - Методы аттестации

Формирование требований

Требования заказчика представляются в такой форме, что они понятны любому пользователю. Они записываются на естественном языке с использованием простых таблиц, а также наглядных диаграмм, рисунков.

Простота формы естественного языка приводит к проблемам содержания, например:

- *Витиеватость стиля изложения.* Иногда нелегко выразить какую-то мысль на человеческом языке ясно и недвусмысленно, не сделав при этом текст многословным и трудночитаемым.
- *Смешение и объединение требований.* В требованиях могут быть размыты границы между функциональными и нефункциональными требованиями. Несколько различных требований могут описываться как единое требование заказчика, и разработчик может сосредоточиться только на одном из них, «потеряв» другое.

Шаги процесса формирования требований

Шаг 1. Определение представителей заказчика.

- Важно выявить такой круг лиц, который позволит составить комплексное представление о портрете будущей системы. Надо разобраться в предметной области системы, выявить круг пользователей и других заинтересованных лиц.

Шаг 2. Проведение опроса представителей заказчика

- Первый вопрос — решить, в каком порядке опрашивать представителей. Планируется время и длительность опросов, на которых должны присутствовать как минимум два члена команды разработчиков и имеются средства записи разговоров. Во время интервью следует сконцентрироваться на слушании, участвовать в диалоге, делать подробные заметки. В конце опроса следует запланировать следующую встречу.

Шаги процесса формирования требований

Шаг 3. Документирование результатов опроса.

- После каждого опроса создается письменная форма набора требований. Она отсылается заказчиком для комментариев и коррекции. Затем проводится серия повторных опросов. Завершается серия опросов собранием. По итогам собрания готовится документ, содержащий все требования. Этот документ утверждается заказчиком.

Шаг 4. Проверка требований.

- Цель проверки спецификации требований состоит в оценке правильности определений, которые в ней содержатся. Проверка гарантирует, что все положения требований корректны, отражают желаемые характеристики и удовлетворяют потребностям заказчика.

Проверка требований

1. Предметная область проекта описана корректно.
2. Разработчик и заказчик имеют одинаковые представления о целях системы.
3. Анализ внешней среды и риска разработки подтверждает возможность создания систем.
4. Спецификация требований верно описывает желаемую функциональность и характеристики системы, которые соответствуют потребностям заказчика и других заинтересованных лиц.
5. Требования полные и качественные.
6. Все требования согласованы друг с другом, не содержат противоречий.
7. Требования обеспечивают реальную возможность разработки системы.

Анализ требований

- Анализ требований рассматривает требования заказчика как исходные данные, на выходе анализа — требования разработчика, которые справедливо называют детальными требованиями.
- Анализ требований служит мостом между подготовкой, планированием и проектированием ПО.
- Происходит переход из мира заказчика в мир разработчика
- Меняется язык записи требований. Теперь это не естественный язык человека, а язык формализованных моделей.

Шаг 1. Организация первичных требований.

Необходимость этой работы обусловлена большим количеством требований. Стандарты рекомендуют несколько способов организации:

- По режиму. Некоторые системы меняют поведение в зависимости от режима работы.
- По категориям пользователей. Некоторые системы обеспечивают различные наборы функций для разных категорий пользователей.
- По объектам. Объекты — это программные сущности системы, которые могут иметь физические аналоги во внешней среде.
- По свойствам. Свойство — сервис, предоставляемый внешней среде, определяется с помощью пар «входное воздействие — реакция».

Шаг 1. Организация первичных требований.

- По стимулам. Некоторые системы легко организуются при описании их функций на языке стимулов.
- По откликам. Некоторые системы организуются посредством описания всех функций, поддерживающих генерацию различных откликов.
- По иерархии функций. То есть путем разделения ПО на множество высокоуровневых функций и последующего разбиения их на подфункции.

При рассмотрении системы могут оказаться применимыми несколько классификационных признаков организации.

Шаг 2. Преобразование первичных требований в детальные требования.

Рекомендации по работе с детальным требованием:

1. Первичная оценка.
 - 1.1 это функциональное требование — соответствует реализующему методу;
 - 1.2 слишком большое — трудно управлять, следует разделить на части;
 - 1.3 слишком маленькое — нет смысла рассматривать отдельно, надо присоединить к другому требованию;
2. Обеспечение прослеживаемости требования. Анализ возможности прослеживания при проектировании и конструировании.
3. Обеспечение тестируемости требования. Написание тестов для проверки реализации требования. Продумываются варианты как положительного, так и отрицательного исхода тестов.

Шаг 2. Преобразование первичных требований в детальные требования.

4. Анализ **ОДНОЗНАЧНОСТИ** толкования требования.
5. Назначение приоритета требования. Выбираются варианты: существенное, желательное или необязательное.
6. Проверка **ПОЛНОТЫ** требования. Следует убедиться в наличии всех «обеспечивающих» требований.
7. Проверка **СОГЛАСОВАННОСТИ** требования с другими требованиями. Анализируются и устраняются возможные противоречия.
8. Требование заносится в спецификацию анализа (спецификацию требований).

Шаг 3. Аттестация детальных требований

Аттестация должна подтвердить, что требования действительно определяют ту систему, которая нужна заказчику.

В состав аттестации входят проверки:

- **правильности требований.** Последующий анализ может выявить дополнительные функции, и их тоже надо учесть.
- **на непротиворечивость.** В требованиях не должно быть противоречащих друг другу ограничений или различных определений одной и той же функции.
- **на полноту.** Требования должны описывать все необходимые функции и ограничения системы.
- **на выполнимость.** Анализируется реализуемость требований в рамках временных и бюджетных ограничений проекта.

Методы аттестации

В ходе аттестации применяют следующие методы:

1. Совместные проверки требований. Цель — найти неточности и ошибки.
2. Макетирование. Макет создается экспертами и обсуждается заказчиком и разработчиками.
3. Генерация тестов. Создание тестов для требований очень часто выявляет проблемы в описании требований.
4. Автоматизированная проверка непротиворечивости. Если требования представлены как формализованные модели, для проверки непротиворечивости моделей можно применить программные CASE-утилиты.

Процесс разработки требований. Содержание 1

- Введение.
- Итеративный процесс формулирования требований.
- Шаблон процесса создания требований.
- Приемы формулирования требований
 - ▶ Сбор информации.
 - ▶ Анализ.
 - ▶ Спецификации.
 - ▶ Проверка.
 - ▶ Управление требованиями.
 - ▶ Обучение при разработке требований.
 - ▶ Требования при управлении проектами.

Процесс разработки требований. Содержание 2

- Распределение работ с требованиями на протяжении жизненного цикла проекта в разных моделях разработки:
 - ▶ Водопадная модель.
 - ▶ Итеративная модель.
 - ▶ Гибкая модель.
- Рекомендации по созданию требований проектов в разных моделях разработки.

Введение

Разработка требований:

- выявление;
- анализ;
- документирование (разработка спецификации);
- проверка.

Все эти действия невозможно выполнить последовательно, за один проход. На практике они выполняются итеративно.

«Поступательная очистка деталей» — основной принцип процесса разработки требований.

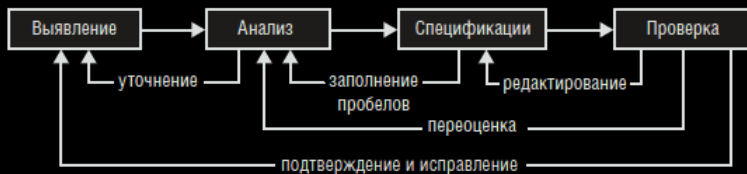


Рис. 1. Итеративный процесс формирования требований

Итеративный процесс формулирования требований

- 1 На этапе **выявления требований** вы задаете клиентам вопросы и наблюдаете, что они делают.
- 2 На этапе **анализа** Вы обработаете эту информацию и связываете потребности клиента с возможными программными требованиями. При этом может потребоваться **уточнение** некоторых требований, для чего нужно вернуться к этапу выявления.
- 3 На этапе **документирования** вы структурируете информацию от пользователей и фиксируете выведенные требования в виде письменных утверждений и диаграмм (**спецификация**). При письменной фиксации может потребоваться вернуться назад и выполнить дополнительный анализ, чтобы **закрыть пробелы** в знаниях.

Итеративный процесс формулирования требований

- 4 На этапе **проверки** вы просите заинтересованных лиц подтвердить что спецификации точно и полно отражает потребности и исправляете ошибки. Это выполняется по отношению к набору требований, которые важнее и своевременнее всего для начала разработки ПО. В процессе проверки может потребоваться **переписать** некоторые неясные требования, **повторить некоторые действия анализа** или даже вернуться и **выполнить дополнительное выявление требований**.

После этого вы переходите к следующему этапу проекта и весь цикл повторяется. Такой итеративный процесс продолжается на всем протяжении разработки требований и, возможно, — как в проектах гибкой разработки — на протяжении всего времени проекта.

Примерный шаблон процесса разработки требований

Определить:

- 1 Бизнес требования.
- 2 Классы пользователей.
- 3 Представления пользователя.
- 4 Лиц ответственных за принятие решений по требованиям.
- 5 План выявления требований.
- 6 Требования пользователей.
- 7 Приоритеты этих требований.

- 8 Сформулировать требования пользователей.
- 9 Вывести функциональные требования.
- 10 Смоделировать требования.
- 11 Определить атрибуты качества.
- 12 Пересмотреть требования.
- 13 Разработать прототипы.
- 14 Разработать (расширить) архитектуру.
- 15 Распределить требования по компонентам.
- 16 Разработать тесты на основе требований.
- 17 Проверить все артефакты требований.

Примерный шаблон процесса разработки требований

- Из-за разнообразия проектов и организационных культур единственный шаблон не возможен, однако данный примерный шаблон создания требований с разумными коррекциями подойдет для большинства проектов.
- Данный процесс порождается является бизнес-потребностью или рыночной возможностью.
- Как правило, действия выполняются по порядку, однако сам процесс не является строго последовательным.
- Первые 7 действий обычно однократно выполняются на ранних стадиях работы над проектом.
- Остальные необходимы для каждого очередного выпуска или этапа работы над проектом. Многие из этих действий могут выполняться итеративно и попеременно. Например, шаги 8, 9 и 10 можно выполнять небольшими порциями, с выполнением пересмотра (шаг 12) после каждой итерации.

Приемы формулирования требований. Введение

Каждому специалисту ПО нужно завести набор приемов, которые он может использовать для решения задач, возникающих в процессе проекта.

Практик, у которого нет набора приемов, вынужден изобретать различные подходы. Другие используют методологии разработки ПО — пакеты методик.

Однако простое следование инструкции или спонтанные методы не всегда дают положительные результаты.

Поэтому, более эффективно применять рекомендуемые в отрасли приемы. Подход на основе рекомендуемых приемов позволяет создать набор различных методик, которые применимы к решению самых разных проблем.

Рекомендуемые приемы

Можно создать группу экспертов и проанализировать проекты различных организаций.

Таким образом экспертам удастся выявить приемы, которые обычно эффективно применялись в успешных проектах, а в неудачных — показали себя плохо или не применялись вообще.

В процессе работы эксперты согласованно определяют, какие рекомендуемые приемы (best practices) дают превосходный результат.

Приемы формулирования требований. Сбор информации 1

- Определите концепцию продукта и границы проекта.
- Определите классы пользователей.
- Выделите из пользователей ярых сторонников продукта.
- Создайте фокус-группы.
- Определите пользовательские требования.
- Определите системные события и реакцию на них.
- Проведите интервью для выявления требований.

Приемы формулирования требований. Сбор информации 2

- Проведите семинары по выявлению требований.
- Наблюдайте за пользователями на рабочих местах.
- Раздайте опросные листы.
- Выполните анализ документов.
- Изучите отчеты о проблемах.
- Повторно задействуйте существующие требования.

Приемы формулирования требований. Анализ

- Смоделируйте среду приложения.
- Создайте прототипы.
- Проанализируйте осуществимость.
- Расставьте приоритеты для требований.
- Создайте словарь данных.
- Проанализируйте интерфейсы.
- Распределите требования по подсистемам.
- Разработайте модели требований.

Анализ требований. Визуальные модели

- диаграммы потоков данных (data flow diagrams, DFD);
- диаграммы рабочих потоков (например swimlane);
- диаграммы переходов состояний (state-transition diagrams, STD) и таблицы состояний;
- карты диалоговых окон;
- таблицы и деревья решений;
- таблицы событий и реакций;
- деревья функций;
- диаграммы вариантов использования;
- диаграммы процессов;
- диаграммы «сущность-связь» (entity-relationship diagrams, ERD).

Приемы формулирования требований. Спецификации и проверка

■ Спецификации:

- ▶ Используйте шаблон спецификации требований.
- ▶ Определите источники требований.
- ▶ Задайте каждому требованию уникальный идентификатор.
- ▶ Задокментируйте бизнес-правила.
- ▶ Определите атрибуты качества.

■ Проверка:

- ▶ Изучите документы с требованиями.
- ▶ Протестируйте требования.
- ▶ Определите критерии приемлемости.
- ▶ Смоделируйте требования.

Управление требованиями

- Определите процесс управления изменениями.
- Проанализируйте, какое влияние оказывают изменения.
- Определите базовую и контрольную версии наборов требований.
- Отслеживайте хронологию изменений.
- Отслеживайте состояние требований.
- Отслеживайте проблемы с требованиями.
- Создайте матрицу связей требований.
- Используйте средство управления требованиями.

Обучение при разработке требований

- Обучите аналитиков требований.
- Ознакомьте представителей пользователей и менеджеров с требованиями.
- Обучите разработчиков основам предметной области.
- Определите процесс разработки требований.
- Создайте словарь терминов.

Требования при управлении проектами

- Выберите соответствующий цикл разработки проекта.
- Планируйте подход к работе с требованиями.
- Оцените объем работ по реализации требований.
- Планируйте на основании требований.
- Определите лиц, ответственных за принятие решений по требованиям.
- Своевременно пересматривайте обязательства.
- Управляйте рисками, касающимися требований.
- Отслеживайте объем работ по реализации требований.
- Делайте выводы из полученного опыта.

Общие рекомендации

Некоторые приемы относятся к нескольким категориям. Большинство этих приемов способствуют более эффективной коммуникации между заинтересованными лицами проекта.

Ни один из этих приемов не будет работать, если вы имеете дело с неблагоприятными людьми. Клиенты, менеджеры и специалисты по ИТ иногда бывают недоговороспособными, но может оказаться, что они просто недостаточно информированы. Они могут не знать, почему вы хотите применить те или иные приемы, или чувствовать себя неуютно с незнакомыми терминами и действиями.

Поэтому важно просветить своих коллег насчет своих приемов и почему вы хотите применить именно их, а также почему для их же собственного блага стоит сотрудничать с остальными членами команды.

Распределение работ с требованиями на протяжении жизненного цикла проекта в разных моделях разработки



Рис. 3. Распределение работ с требованиями на протяжении жизненного цикла проекта в разных моделях разработки

Распределение работ с требованиями на протяжении жизненного цикла проекта в водопадной модели разработки

Общий объем работ по работе с требованиями может быть практически таким же, как и в проектах сравнимого размера с другими жизненными циклами, но распределение работ по времени может сильно отличаться.

- В чистом водопадном цикле планируется только один основной выпуск, поэтому **основной объем работы с требованиями приходится на начало проекта** (сплошная линия на Рис. 3). Такой подход используется в очень многих проектах, но подходит для немногих. Но даже если вы спланируете традиционный этап «выявления требований» в начале проекта, после чего требования используются для проектирования, нужно рассчитывать на выполнение небольшого объема работ с требованиями на протяжении всего проекта.

Распределение работ с требованиями на протяжении жизненного цикла проекта в итеративной модели разработки

- В проектах, в которых используется итеративный процесс разработки, такой как Rational Unified Process (Jacobson, Booch и Rumbaugh, 1999), **работа с требованиями ведется в каждой итерации процесса разработки**, причем на первую итерацию приходится больший объем работы (пунктирная линия на Рис. 3). Это же происходит, если вы запланировали серию поэтапных выпусков, в каждом из которых предоставляется значительная часть конечной функциональности продукта.

Распределение работ с требованиями на протяжении жизненного цикла проекта в модели гибкой разработки

В проектах гибкой разработки (agile) планируется выпускать функциональность каждые несколько недель (Larman, 2004). В таких проектах **работа над требованиями выполняется часто, но небольшими объемами**, как показано на Рис. 3 точечной линией.

Работа начинается с выявления пользовательских требований в форме пользовательских историй, которые описывают основные задачи, которые пользователи хотят решить с помощью системы. В этом подходе из историй нужно извлечь достаточно информации, чтобы можно было оценить объем и определить приоритеты разработки.

Приоритизация пользовательских требований позволяет определить, какие из них назначить на те или иные этапы разработки, которые называются итерациями, или спринтами.

Рекомендации по созданию требований проектов в разных моделях разработки

Независимо от используемой в проекте модели, нужно при каждом выпуске или итерации задавать себе вопрос, какие из действий, описанных на Рис. 2, позволят увеличить ценность и снизить риск. По завершении шага 17, заключающегося в проверке пользовательских, функциональных, нефункциональных требований, моделей анализа и прототипов, для любой части требований можно приступить к построению соответствующей части системы. Повторите шаги с 8 по 17 со следующим набором требований, который ляжет в основу следующего выпуска или итерации.

Заключение

- Формулирование требований представляет собой итеративный процесс, состоящий из выявления, анализа, документирования и проверки.
- Шаблон процесса создания требований является единым для большинства проектов.
- В проектах с разными моделями разработки распределение работ, связанных с требованиями, существенно отличается: в водопадной модели основной объем работы с требованиями приходится на начало проекта; в итеративной – работа с требованиями ведется в каждой итерации процесса разработки, причем на первую итерацию приходится больший объем работы; в гибкой – работа над требованиями выполняется часто, но небольшими объемами.

Документ о концепции и границах. Содержание

- Документ о концепции и границах
- Использование шаблонов
- Бизнес-требования
 - 1 Исходные данные
 - 2 Возможности бизнеса
 - 3 Бизнес-цели
 - 4 Критерии успеха
 - 5 Положение о концепции
 - 6 Бизнес-риски
 - 7 Предположения и зависимости
- Рамки и ограничения проекта
 - 1 Основные функции
 - 2 Объем первоначально запланированной версии
 - 3 Объем последующих версий
 - 4 Ограничения и исключения
- Бизнес-контекст
 - 1 Профили заинтересованных лиц
 - 2 Приоритеты проекта
 - 3 Особенности развертывания

Документ о концепции и границах

- Содержит все важные для разработки бизнес-требования.
- Не нужно дублировать информацию.
- Определяются границы на высоком уровне (подробности границ представлены базовыми требованиями отдельных выпусков, определенных командой).
- Должен быть завершенным для крупных новых проектов
- Каждый этап улучшения продукта может включать собственное положение о границах в документации требований к проекту (отдельный документ создавать необязательно).
- Владелец документа - куратор проекта.
- Информация поступает от лиц, четко понимающих, почему они взялись за проект.

Использование шаблонов

Рекомендуемый шаблон документа о концепции и границах:

1. Бизнес-требования
 - 1.1 Исходные данные
 - 1.2 Возможности бизнеса
 - 1.3 Бизнес-цели
 - 1.4 Критерии успеха
 - 1.5 Положение о концепции проекта
 - 1.6 Бизнес-риски
 - 1.7 Предположения и зависимости
2. Рамки и ограничения проекта
 - 2.1 Основные функции
 - 2.2 Объем первоначально запланированной версии
 - 2.3 Объем последующих версий
 - 2.4 Ограничения и исключения
3. Бизнес-контекст
 - 3.1 Профили заинтересованных лиц
 - 3.2 Приоритеты проекта
 - 3.3 Особенности развертывания

1. Бизнес-требования

Для чего нужны:

- описывают основные преимущества системы;
- определяют какие пользовательские требования будут реализованы и в какой последовательности.

1.1 Исходные данные

- 1 Суммируют обоснование и содержание нового продукта или изменения, которые нужно внести в существующий продукт
- 2 Описывают те ситуации, в которых было решено создать продукт

1.2 Возможности бизнеса

Что следует описать:

- Задачи, которые поможет разрешить предлагаемое решение.
- Насколько решение соответствует тенденциям рынка, развитию технологий или корпоративной стратегии.
- Другие технологии, процессы или ресурсы, необходимые для удовлетворения клиента.
- Потребности типичных клиентов или целевого рынка.
- Задачи клиента, которые будет решать новый продукт.
- Примеры того, как клиенты будут использовать продукт.
- Указать все известные критичные требования к качеству или интерфейсам

1.3 Бизнес-цели

Суммируют важные преимущества бизнеса, предоставляемые продуктом, в количественном и измеряемом виде.

Модель бизнес-целей отображает иерархию связанных бизнес-проблем и измеряемых бизнес-целей. Проблемы описывают, что не позволяет организации достичь требуемых ориентиров в настоящее время, а бизнес-цели определяют способы измерения достижения этих ориентиров.

Финансовые цели	Нефинансовые цели
Освоить X% рынка за Y месяцев	Достигнуть показателя удовлетворения покупателей, равного по крайней мере X, в течение Y месяцев со времени выпуска продукта
Увеличить долю рынка в стране W с X% до Y% за Z месяцев	Увеличить производительность обработки транзакций на X% и снизить уровень ошибок данных до величины не более Y%
Достигнуть объема продаж X единиц или дохода в Y долларов за Z месяцев	Разработать надежную платформу для семейства связанных продуктов
Получить X% прибыли по инвестициям в течение Y месяцев	

1.3 Бизнес-цели: определение целей и задач

Для определения бизнес-задач и целей следует задавать ряд уточняющих вопросов.

Процесс выполняется итеративно путем передвижения по иерархии задач и целей, пока не получится список функций, необходимых для решения задачи.

Пример разговора между аналитиком и куратором проекта

Вопросы аналитика	Ответы топ-менеджера
Какими мотивами вы руководствуетесь, заказывая систему контроля химикатов?	Управление наличными запасами химикатов неэффективно и обходится слишком <u>дорог</u>
Насколько вы бы хотели сократить свои расходы на химикаты?	На 25% в течение года
Что сейчас мешает сократить расходы на 25%? Что вызывает высокие затраты и неэффективность?	Мы покупаем лишние химикаты, потому что не знаем, какие у нас имеются запасы. Мы утилизируем слишком много неиспользованных материалов с истекшим сроком годности
Есть еще какие-то вещи, о которых я должен знать?	Процедура размещения заказа сложна и отнимает у пользователей много времени. Отчеты для регулирующих органов создаются в ручную, на что тратится неоправданно много времени

1.4 Критерии успеха

Необходимо установить факторы, которые максимально влияют на успех проекта — те, которые организация может контролировать, и те, которые находятся вне сферы ее влияния.

Иногда бизнес-цели невозможно измерить, пока не завершится проект, а в других случаях достижение бизнес-целей может зависеть от других проектов. Однако успех проекта важно оценить.

Критерии успеха:

- указывают, находится ли проект на пути достижения бизнес-целей;
- могут определяться во время тестирования или сразу после выпуска продукта;
- должны оценивать то, что важно для бизнеса, а не просто то, что легко оценить.

1.5 Положение о концепции

- 1 Обобщает долгосрочные цели и назначение нового продукта.
- 2 Удовлетворяет различным заинтересованным лицам.
- 3 Основывается на существующих или предполагаемых рыночных факторах.

Ключевые слова:

- Для [целевая аудитория покупателей];
- Который [положение о потребностях или возможностях];
- Эта (этот) [имя продукта];
- Является [категория продукта];
- Который(ая) [основные функции, ключевое преимущество, основная причина для покупки или использования];
- В отличие от [основной конкурирующий продукт];
- Наш продукт [положение об основном отличии и преимуществе].

Бизнес-риски. Предположения и зависимости

1.6 Бизнес-риски

- Риски: рыночная конкуренция, временные факторы, приемлемость для пользователей, проблемы, связанные с реализацией, возможные негативные факторы, влияющие на бизнес.
- Нужно оценить: возможные потери от каждого фактора риска, вероятность его возникновения, способность контролировать его, определить все возможные действия по смягчению ситуации

1.7 Предположения и зависимости

Что следует задокументировать:

- все предположения, сделанные заинтересованными лицами;
- важнейшие зависимости проекта от внешних факторов;
- возможные последствия того, что предположения окажутся ошибочными или зависимости нарушены.

2. Рамки и ограничения проекта

«Распухание границ» — резкий рост из-за активного расширения функциональности продукта.

Обуздать распухание границ можно определив рамки проекта, которые помогают установить реалистичные ожидания заинтересованных лиц.

Границы проекта определяют концепцию и круг действия предложенного решения.

Границы определяются через набор функциональных требований, которые планируется реализовать в определенном выпуске или итерации. То есть границы определяются:

- На самом высоком уровне - когда клиент решает, какие бизнес-цели преследовать.
- На низком - функции, пользовательских историй, вариантов использования или событий и реакции на них.

Основные функции. Объем первоначально запланированной версии

2.1 Основные функции. Необходимо:

- описать основные функции продукта или возможности пользователей. Важно показать отличия продукта от предыдущей версии или конкурентов.
- понять как пользователи будут работать с этими функциями, убедиться, что список функций полон и не содержит ненужных функций.

2.2 Объем первоначально запланированной версии

Если задача — сосредоточиться на разработке и уложиться в график, то в версии 1.0 должны содержаться только наиболее ценные функции (с максимально приемлемой стоимостью), исключая функции, которые когда-нибудь в будущем могут пригодиться. Распухание кода и сдвиг графика — типичные исходы такого набивания объема.

Объем последующих версий. Ограничения и исключения

2.3 Объем последующих версий

В случае поэтапной эволюции продукта или использовании итеративной модели разработки, нужно создать план выпуска - указывает функции, которые будут отложены и желательные сроки последующих выпусков.

2.4 Ограничения и исключения

В ограничениях и исключениях следует перечислить:

- все возможности или характеристики, которые могут ожидать заинтересованные в проекте лица, но включение которых в продукт или в определенную версию не запланировано.
- изъятые элементы, чтобы не забыть решения по границам проекта.

3 Бизнес-контекст: Профили заинтересованных лиц

Заинтересованные в проекте лица - это отдельные лица, группы или организации, которые активно вовлечены в проект, на которых влияет результат проекта и которые сами могут влиять на этот результат.

В профиль каждого заинтересованного лица включается:

- Ценность для заинтересованных лиц:
 - ▶ повышенная производительность;
 - ▶ меньшее количество переделок;
 - ▶ снижение себестоимости;
 - ▶ ускорение бизнес-процессов;
 - ▶ автоматизация задач, ранее выполнявшихся вручную;
 - ▶ возможность выполнять совершенно новые задачи;
 - ▶ соответствие соответствующим стандартам и правилам;
- их вероятное отношение к продукту;
- самые важные для них функции и характеристики;
- все известные ограничения, которые должны быть соблюдены.

3.2 Приоритеты проекта

Каждое из измерений относится к определённой категории:

- ограничение — сдерживающий фактор, в рамках которого должен оперировать менеджер проекта;
- ведущий фактор — важный фактор успеха, ограниченно гибкий при изменениях;
- степень свободы — возможность менять измерение и балансировать относительно других измерений.

При неожиданном ускорении срока выпуска проекта можно:

- отложить реализацию определенных требований;
- сократить запланированный цикл тестирования системы;
- оплатить сверхурочную работу или пригласить специалистов по контракту;
- привлечь ресурсы других проектов для разрешения ситуации.

3.3 Особенности развертывания

- Перечислить информацию и действия, необходимые для обеспечения эффективного развертывания решения в рабочую среду.
- Описать доступ, который потребуется пользователями для работы с системой (находятся ли они далеко в разных часовых поясах или недалеко друг от друга).
- Указать, когда пользователям в разных местах нужен доступ к системе. Если требуются, изменения инфраструктуры, чтобы обеспечить потребности ПО в мощностях, доступе к сети, хранилищу данных и миграции данных, следует описать их.
- Зафиксировать информацию, которая потребуется тем, кто будет готовить бизнес-процессы обучения и модификации в связи с развертыванием нового решения.

Спасибо за внимание.