

приемы создания требований

План

1. Анализ требований
2. Спецификации требований
3. Проверка требований
4. Управление требованиями
5. Управление проектом

Анализ требований подразумевает их детализацию, гарантирующую, что требования понимают все заинтересованные лица, а также тщательное исследование требований на предмет ошибок, пробелов и других недостатков. Анализ включает создание прототипов, анализ осуществимости и согласование приоритетов.

Цель анализа — достаточно качественно и подробно описать требования, позволяющие менеджерам реалистично оценить все затраты на проект, а техническому персоналу — начать проектирование, сборку и тестирование.

Отдельные требования стоит представить несколькими способами, например в текстовой и графической форме.

Создание контекстной диаграммы. Контекстная диаграмма — простая модель анализа, отображающая место новой системы в соответствующей среде. Она определяет границы и интерфейсы между разрабатываемой системой и сущностями, внешними для этой системы, например пользователями, устройствами и прочими информационными системами.

Создание пользовательского интерфейса и технических прототипов. Если разработчики или пользователи не совсем уверены насчет требований, создайте прототип — частичную, возможную или предварительную версию продукта, которая сделает концепции и возможности более осязаемыми. Оценка прототипа поможет всем заинтересованным лицам достичь взаимопонимания по решаемой проблеме.

Анализ осуществимости требований. Проанализируйте, насколько реально реализовать каждое требование при разумных затратах и с приемлемой производительностью в предполагаемой среде. Рассмотрите риски, связанные с реализацией каждого требования, включая конфликты с другими требованиями, зависимость от внешних факторов и препятствия технического характера

Определение приоритетов требований. Определите относительные приоритеты реализации функций продукта, решаемых задач или отдельных требований. На основании приоритетов установите, в какой версии будет реализована та или иная функция или набор требований. Подтверждая изменения, распределите все их по конкретным версиям и включите в план выпуска этих версий затраты, необходимые на внесение изменений. В ходе работы над проектом периодически корректируйте приоритеты в соответствии с потребностями клиента, условиями рынка и бизнес-целями.

Моделирование требований. В отличие от подробной информации, представленной в спецификации требований к ПО или пользовательского интерфейса прототипа, графическая модель анализа отображает требования на высоком уровне абстракции. Модели позволяют выявить некорректные, несогласованные, отсутствующие и избыточные требования. К таким моделям относятся диаграммы потоков данных, диаграммы «сущность — связь», диаграммы перехода состояний, называемые также *автоматами*, карты диалогов, диаграммы классов, диаграммы последовательностей, диаграммы взаимодействий, таблицы решений и деревья решений.

Создание словаря терминов. В нем соберите определения всех элементов и структур данных, связанных с системой, что позволяет всем участникам проекта использовать согласованные определения данных. На стадии работы над требованиями словарь должен содержать определения элементов данных, относящихся к предметной области, чтобы клиентам и разработчикам было проще общаться.

Распределение требований по подсистемам. Требования к сложному продукту, включающему несколько подсистем, следует соразмерно распределять между программными, аппаратными и операторскими подсистемами и компонентами. Как правило, это осуществляет системный инженер или разработчик.

Применение технологий развертывания функций качества. Технология развертывания функций качества — точная методика, соотносящая возможности и атрибуты продукта с их значимостью для клиента. Она позволяет аналитически выявить функции, которые максимально удовлетворят потребности клиента.

Технология развертывания функций качества рассчитана на три класса требований: **ожидаемые**, о которых клиент может не упомянуть, но будет расстроен, если их не окажется в продукте, **обычные** требования и отдельные, **специальные** требования, которые обеспечивают удобство работы клиентам, но отсутствие которых не влечет санкций со стороны клиента.

Контекстная диаграмма

Уточнение рамок определяет границу и связи системы, которую мы разрабатываем, со всем остальным миром.

Контекстная диаграмма (contextdiagram) графически иллюстрирует эту границу. Она определяет **оконечные элементы**, расположенные вне системы, которые определенным образом взаимодействуют с ней, а также данные, элементы управления и материальные потоки, протекающие между оконечными элементами и системой. Контекстная диаграмма представляет собой высший уровень абстракции в диаграмме потока данных, разработанной по принципам структурного анализа, но эта модель полезна и в случае применения какой-либо другой методики разработки.

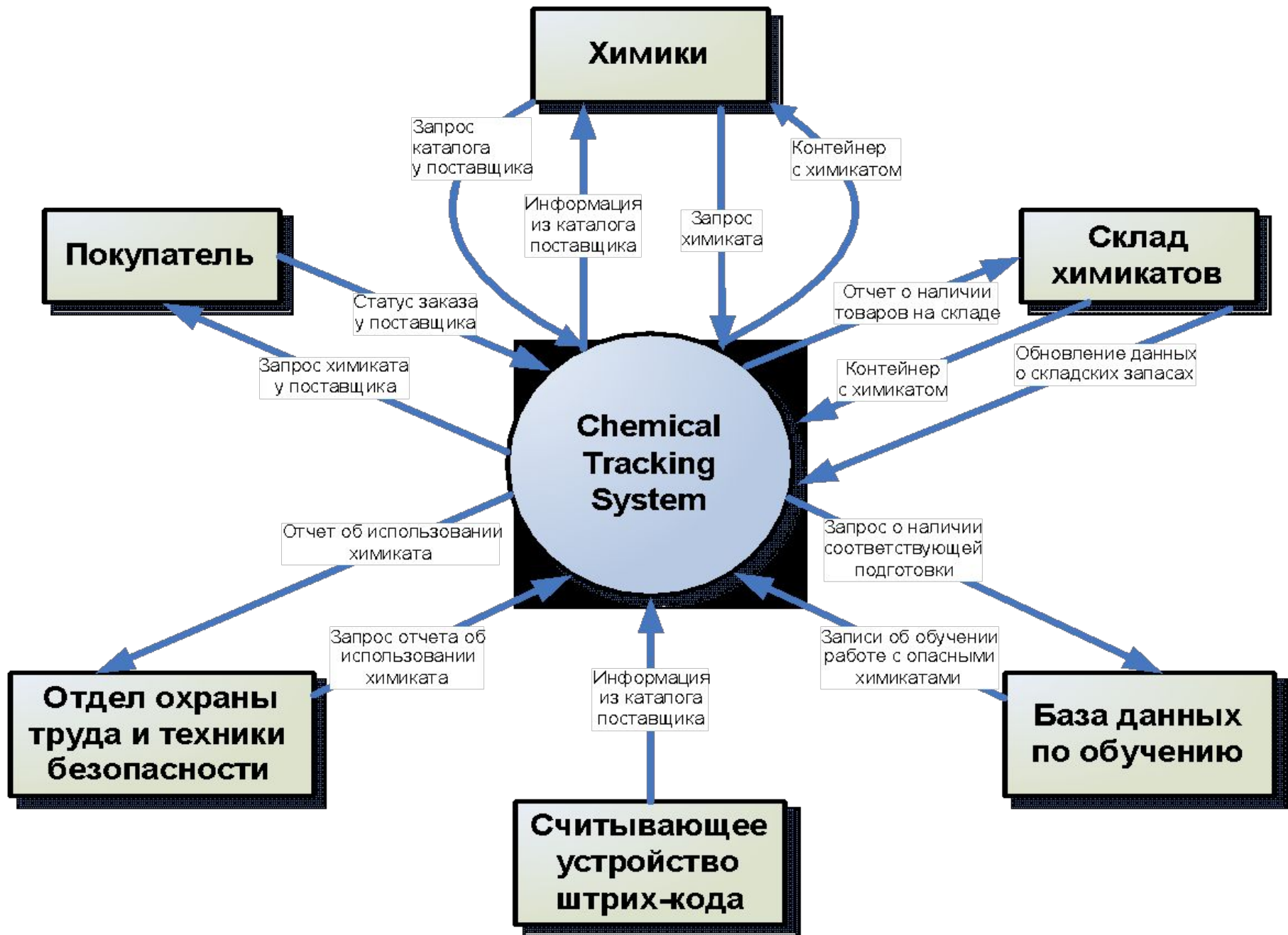
Можно включить контекстную диаграмму в документ об образе и границах, или определить ее как приложение к спецификации требований, или как часть модели потоков данных системы.

На рис. показана часть контекстной диаграммы для Chemical Tracking System.

Вся система изображена кружком; на контекстной диаграмме намеренно не показывают внутренние объекты системы, процессы и данные. «Система» внутри кружка может иметь любую комбинацию ПО, оборудования или людских ресурсов. Оконечные элементы в прямоугольниках представляют классы пользователей («Химик» или «Покупатель»), отделы («Отдел охраны труда и техники безопасности»), другие системы («База данных по обучению») или аппаратные устройства («Считывающее устройство штрих-кода»). Стрелками показаны потоки данных («запрос химиката») или физические элементы («контейнер с химикатом») между системой и окончательными элементами.

Вы можете ожидать, что поставщики химикатов должны быть показаны на диаграмме в виде окончательных элементов. Ведь компания направляет заказы для выполнения поставщикам, а те отправляют контейнеры с химикатами и счета в ContosoPharmaceuticals, отдел же закупок пересылает чеки продавцам. Однако эти процессы происходят вне ChemicalTrackingSystem, как часть операций отделов закупок и приобретений. Глядя на контекстную диаграмму становится совершенно ясно, что система не участвует напрямую в размещении заказов у поставщиков, в получении продуктов или оплате счетов.

Назначение таких средств, как контекстная диаграмма, заключается в стимулировании ясного и точного взаимодействия между заинтересованными в проекте лицами. Рекомендовано использовать схему, в качестве стандарта для изображения контекстной



Спецификации требований

Независимо от способа выявления требований, документировать их нужно так, чтоб это обеспечивало удобный доступ и просмотр. Зафиксировать **бизнес-требования** можно в положении об образе и границах проекта. **Пользовательские** требования обычно представляют в виде вариантов применения или таблиц «событие — реакция». **Спецификация** требований содержит подробные функциональные и нефункциональные требования к ПО.

Шаблон для спецификации требований к ПО

на основе стандарта IEEE 830

- **1. Введение**
- 1.1 Назначение
- 1.2 Соглашения, принятые документах
- 1.3 Предполагаемая аудитория и рекомендации по чтению
- 1.4 Границы проекта
- 1.5 Ссылки

- **2. Общее описание**
- 2.1 Общий взгляд на продукт
- 2.2 Особенности продукта
- 2.3 Классы и характеристики пользователей
- 2.4 Операционная среда
- 2.5 Ограничения проектирования и реализации
- 2.6 Документация для пользователей
- 2.7 Предположения и зависимости
- **3. Функции системы**
- 3.x Функция системы X
- 3.x.1 Описание и приоритеты
- 3.x.2 Последовательности «воздействие - реакция»
- 3.x.3 Функциональные требования
- **4. Требования к внешнему интерфейсу**
- 4.1 Интерфейсы пользователя
- 4.2 Интерфейсы оборудования
- 4.3 Интерфейсы ПО
- 4.4 Интерфейсы передачи информации

5. Другие нефункциональные требования

5.1 Требования к производительности

5.2 Требования к охране труда

5.3 Требования к безопасности

5.4 Атрибуты качества

6. Остальные требования

Приложение А. Словарь терминов

Приложение Б. Модели анализа

Приложение Г. Список вопросов

Определение источников требований.

Чтобы гарантировать, что все заинтересованные лица понимают, почему то или иное требование зафиксировано в спецификации требований к ПО, и упростить последующее прояснение требований, выявите источники всех требований. Это может быть вариант использования или другая информация от пользователей, системное требование высокого уровня, бизнес-правило или иной внешний фактор. Указав всех лиц, заинтересованных в каждом требовании, вы будете знать, к кому обратиться при поступлении запроса на изменение.

Присвоение уникальных идентификаторов всем требованиям,

Выработайте соглашение о присвоении уникальных идентификаторов требованиям, зафиксированным в спецификации требований к ПО. Соглашение должно быть устойчивым к дополнению, удалению элементов и изменениям, вносимым в требования. Присвоение идентификаторов позволяет отслеживать требования и фиксировать вносимые изменения.

- **Указание атрибутов качества.** Выявляя качественные характеристики, удовлетворяющие потребности клиента, не ограничивайтесь только обсуждением функциональности. Выясните ожидаемые производительность, эффективность, надежность, удобство использования и др. Информация от клиентов об относительной важности тех или иных качественных характеристиках позволит разработчику принять правильные решения, касающиеся архитектуры приложения.
- **Документирование бизнес-правил.** К бизнес-правилам относятся корпоративные политики, правительственные распоряжения и алгоритмы вычислений. Ведите список бизнес-правил отдельно от спецификации требований к ПО, поскольку правила обычно существуют вне рамок конкретного проекта. Для выполнения некоторых приходится создавать реализующие их функциональные требования, и поэтому необходимо определить связь между этими требованиями и соответствующими правилами.

Проверка требований

Проверка гарантирует, что все положения требований корректны, отражают желаемые качественные характеристики и удовлетворяют потребностям клиента. В большинстве случаев удастся выявить двусмысленности и неопределенности, написав для требований сценарии тестирования.

Изучение документов с требованиями. Официальная проверка документирования требований — один из наиболее ценных способов, проверки качества ПО. Соберите небольшую команду, члены которой представляют различные направления (например, аналитик, клиент, разработчик и специалист по тестированию), и тщательно изучите спецификацию требований к ПО, модель анализа и соответствующую информацию на предмет недостатков. Данный прием — один из самых полезных.

Тестирование требований. На основе пользовательских требований создают сценарии функционального тестирования и документируют ожидаемое поведение продукта в конкретных условиях. Проследите связь сценариев тестирования с функциональными требованиями и удостоверьтесь, что ни одно требование не пропущено и что для всех требований есть соответствующие сценарии тестирования.

Определение критериев приемлемости. Предложите пользователям описать, как они собираются определять соответствие продукта их потребностям и его пригодность к работе. Тесты на приемлемость следует основывать на сценариях использования

Управление требованиями

Начальные требования неизбежно корректируются в процессе работы клиентами, менеджерами, специалистами по маркетингу, разработчиками и другими лицами. Для **эффективного управления требованиями** необходим процесс, позволяющий предлагать изменения и оценивать их возможную стоимость и влияние на проект.

Определение процесса управления изменениями.

Определить процесс представления, анализа и утверждения или отклонения изменений. Применяйте его для управления всеми предлагаемыми изменениями. В контексте процесса управления изменениями полезно использовать коммерческие средства отслеживания недостатков.

Создание совета по управлению изменениями.

Из представителей заинтересованных в проекте лиц организуйте совет по управлению изменениями, который будет получать информацию о предполагаемых изменениях требований, оценивать ее, решать, какие изменения принять, а какие отклонить, и определять, в какой версии продукта будет внедрена та или иная функция.

Анализ влияния изменений требований.

Анализ влияния изменений помогает совету по управлению изменениями принимать обоснованные решения.

Создание базовой версии и управление версиями требований.

Базовая версия содержит требования, утвержденные для реализации в конкретной версии продукта. После определения базовых требований изменения можно вносить только в соответствии с процессом управления изменениями. Присвойте всем версиям спецификации требование уникальные идентификаторы, чтобы избежать путаницы между черновыми вариантами и базовыми версиями, а также между предыдущей и текущей версиями требований. Более надежное решение — управлять версиями документов с требованиями при помощи соответствующих средств управления конфигурацией.

Ведение журнала изменений требований.

Фиксируйте даты изменения спецификаций требований, сами коррективы, их причины, а также лиц, вносивших изменения. Автоматизировать эти задачи позволяет утилита управления версиями или коммерческая утилита управления требованиями.

Контроль за состоянием всех требований. Создайте БД, включающую по одной записи для каждого дискретного функционального требования. Занесите в БД ключевые атрибуты каждого требования, включая его состояние (например «предложено», «одобрено», «реализовано» или «проверено»), чтобы в любой момент вы могли узнать количество требований в каждом состоянии,

Оценка изменяемости требований. Еженедельно фиксируйте количество требований, внесенных в базовую версию, а также число предложенных и одобренных изменений (добавлений, модификаций и удалений).

Использование средств управления требованиями. Коммерческие утилиты управления требованиями позволяют хранить различные типы требований в БД. Для каждого требования можно определить атрибуты, отслеживать его состояние, а также выявить связи между требованиями и другими рабочими продуктами. Данный прием поможет вам автоматизировать прочие задачи по управлению требованиями, описанные ниже.

Создание матрицы связей требований. Создайте таблицу, сопоставляющую все функциональные требования с элементами архитектуры и кода, которые реализуют данное требование, и с тестами, проверяющими его. Матрица связей требований позволяет также сопоставить функциональные требования с требованиями более высоких уровней на основе которых они созданы и с другими родственными

Управление проектом

Способы управления проектом ПО тесно связаны с работой над требованиями к нему. Планируйте ресурсы, графики и обязательства по проекту на основании требований, которые собираетесь реализовать. Изменения требований влияют на планы реализации, поэтому в планах следует предусмотреть возможность частичного изменений требований и расширения границ проекта.

Выбор цикла разработки ПО. Следует определить, несколько жизненных циклов разработки для проектов различного типа и различных степеней неопределенности требований. Каждый менеджер проекта должен выбрать и использовать цикл, оптимальным образом подходящий для его проекта. Включите в цикл операции по созданию требований. Если на ранних этапах работы над проектом требования или границы проекта определены нечетко, разрабатывайте продукт постепенно (небольшими этапами), начиная с наиболее понятных требований и устойчивых элементов архитектуры. По возможности реализуйте наборы функций, чтобы периодически выпускать промежуточные версии продукта и как можно раньше предоставлять клиенту работоспособные образцы приложения

Планы реализации проекта должны быть основаны на требованиях. Разрабатывайте планы и графики работы над проектом постепенно, по мере прояснения границ и подробных требований. Начните с оценки затрат, необходимых на реализацию функциональных требований, определенных на основе первоначального образа и границ продукта.

Пересмотр обязательств по проекту при изменении требований.

Добавляя в проект новые требования, оцените, удастся ли соблюдать обязательства, касающиеся графика и требований к качеству, при доступном объеме ресурсов. Если нет, обсудите реалии проекта с менеджерами и согласуйте новые, достижимые обязательства.

Документирование и управление рисками, связанными с требованиями. Одна из составляющих управления рисками проекта — выявление и документирование рисков, связанных с требованиями. Уменьшайте или предотвращайте их посредством мозговых штурмов, реализуйте корректирующие действия и отслеживайте их эффективность.

Контроль объема работ по созданию требований. Фиксируйте усилия, прилагаемые вашей командой на разработку требований и управление проектом. Эти данные позволят оценить соответствие планам и эффективнее спланировать необходимые ресурсы для будущих проектов.

Извлечение уроков из полученного опыта. Для этого в организации следует провести *ретроспективу* проектов, называемую также *изучением законченных проектов*. Ознакомление с опытом в области проблем и способов создания требований, накопленным в ходе работы над предыдущими проектами, помогает менеджерам и аналитикам требований более эффективно работать в будущем.