

Основные положения

Требование - это некоторое свойство программного обеспечения, которым должна обладать система или её компонент, чтобы удовлетворять требованиям контракта, стандарта, спецификации либо иной формальной документации.

Требование - 1) условия или возможности, необходимые пользователю для решения проблем или достижения целей; 2) условия или возможности, которыми должна обладать система или системные компоненты, чтобы выполнить контракт или удовлетворять стандартам, спецификациям или другим формальным документам; 3) документирование представления условий или возможностей для пунктов 1) и 2).

Классификация требований

Для удобства работы и управления требованиями их классифицируют. Классификация требований по определённым типам позволяет разделить требования по уровням абстракции, природе, назначению и другим признакам.

Требования:

к продукту: в своей основе - то, что формулирует заказчик, цель, которую он преследует (получить хороший конечный продукт, т. е. функциональный и удобный в эксплуатации), поэтому требования к продукту являются основополагающим классом требований; в дальнейшем мы будем более подробно рассматривать и классифицировать этот тип требований; к проекту (процессу): требования к тому, как разработчик будет выполнять работы по созданию целевой системы.

Бизнес-требования: описывают высокоуровневые цели организации или заказчиков системы. Как правило, их формулируют те, кто финансирует проект, либо покупатели системы, менеджеры пользователей, отдел маркетинга и т. д. Бизнес-требования относятся к наивысшему уровню абстракции требований и обычно характеризуют цели организации, её миссию в решении проблем бизнеса. На практике иногда аналитики пренебрегают бизнес-требованиями, что в итоге может привести к созданию системы, которая будет удобна в эксплуатации, но не решает задачи бизнеса.

Ключевые возможности (характеристики продукта)

Ключевая возможность - это набор логически связанных функциональных и нефункциональных требований, которые обеспечивают возможности пользователя и удовлетворяют бизнес-требованиям. В области коммерческого программного обеспечения ключевая возможность представляет узлаваемую всеми заинтересованными лицами группу требований, которые важны при принятии решения о покупке, т. е. это фактически элемент маркетингового списка в описании продукта. В отличие от бизнес-требований, которые описывают цели организации, ключевые возможности на высоком уровне абстракции описывают функциональность и характеристики качества разрабатываемой информационной системы.

Пользовательские требования: описывают цели и задачи, которые пользователи смогут решать при помощи системы, т. е. они описывают систему с точки зрения конечного пользователя, т. е. человека, непосредственно работающего с системой. На практике часто пользовательские требования идут в разрез с бизнес-требованиями, например, заказчик хочет получить дешёвую систему, а пользователь желает работать с удобным пользовательским интерфейсом, реализация которого может дорого обойтись заказчику. Подобные противоречия должен идентифицировать и решать системный аналитик.

Функциональные требования: определяют функциональность программного обеспечения, которую разработчик должен обеспечить, чтобы пользователи смогли выполнять свои задачи в рамках пользовательских и бизнес-требований. Иногда эти требования называют требованиями поведения, поскольку они описывают детальное поведение системы, выполняемые ею действия и отклики, а также информацию, которой система будет управлять. Поскольку функциональные требования являются низкоуровневым описанием действий системы, они часто реализуются на уровне методов класса и данных, которые обрабатываются в конкретном методе.

Характеристики качества: описывают цели и атрибуты качества разрабатываемой системы. Атрибуты качества системы представляют собой дополнительное описание функций продукта, выраженное через описание его характеристик, важных для пользователей или разработчиков. К таким характеристикам относятся: лёгкость и простота использования, производительность, удобство эксплуатации и технического обслуживания, надёжность и устойчивость к сбоям, взаимодействие системы с внешним миром, расширяемость, требования к пользовательским и программным интерфейсам.

Ограничения: к требованиям данного типа относятся технические ограничения и бизнес-правила. Технические ограничения касаются выбора способа разработки, внешнего вида и структуры продукта, используемых языков программирования, технологий и платформ. Бизнес-правила включают корпоративные политики, правительственные постановления, промышленные стандарты, вычислительные алгоритмы и т. д. Иногда бизнес-правила

становятся источником атрибутов качества, которые реализуются в функциональности.

Иерархия требований  
[Бизнес-требования]

[Ключевые возможности (характеристики продукта)]

[Требования пользователя] < [Характеристики качества]

[Функциональные требования] [Ограничения]

Системные требования и требования к ПО

Существует две основные трактовки понятия "системные требования". Виргес формулирует этот термин следующим образом: высокоуровневые требования к продукту, который содержит множество подсистем, т. е. это требования к системе. При этом под системой понимается программная, программно-аппаратная либо человеко-машинная система. В практике компьютерной инженерии бытует другой, более узкий контекст: под системными требованиями понимаются требования, выдвигаемые прикладной программной системой, в частности, информационной, к среде своего функционирования (системной или аппаратной).

Управление требованиями

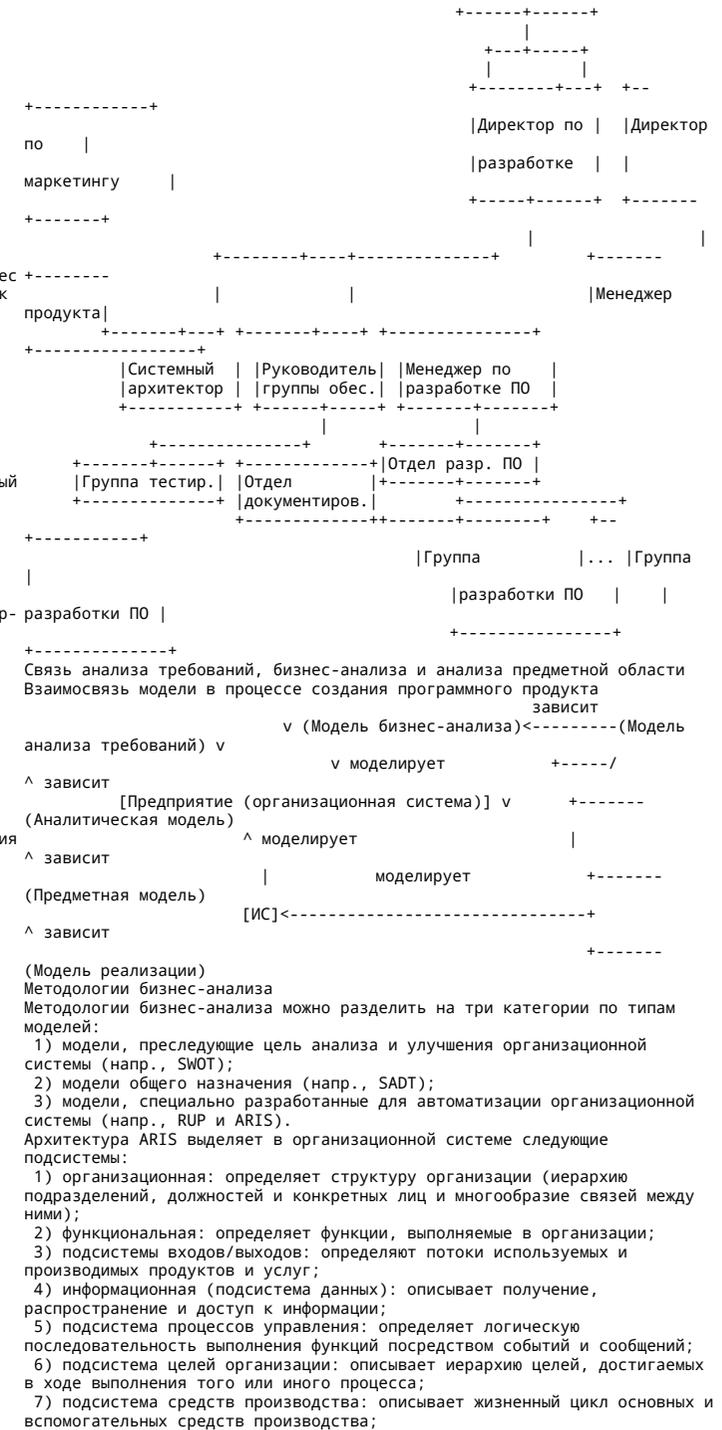
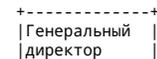
В качестве основы управления качеством требований к ПО предлагается использовать 6 уровней зрелости процесса управления требованиями. Каждый последующий уровень процесса управления качеством требований. Каждый последующий уровень полностью включает в себя предшествующий.



0: команда разработчиков уверена в том, какой продукт следует реализовать; члены команды разработчиков считают, что они экономят время, пропуская этапы задачи выявления и документирования требований 1: цель уровня заключается в получении документов, описывающих требования. На основе документов в дальнейшем разрабатываются сценарии тестирования, архитектура ПО, руководства пользователей и другая проектная документация

Технический описатель разрабатывает документацию на разработку информационной системы согласно государственному и международным стандартам. Коммуникатор отвечает за организацию, проведение, моделирование и протоколирование совещаний, фокус-групп и мозговых штурмов.

Примерный состав команды разработчиков:



8) подсистема человеческих ресурсов: описывает приём на работу, обучение и продвижение по службе персонала организации;

9) подсистема расположения организационных структур: описывает территориальное расположение организационных единиц.

Требования и архитектура информационной системы

Говоря об архитектуре информационной системы, обычно подчёркивают деление её на аппаратные, программные, информационные, организационные компоненты, их связность и детализацию. Как только собран минимальный набор ключевых требований, дающих понимание того, что нужно делать, должна быть найдена архитектура, объясняющая, как это можно реализовать. В крупных и ответственных проектах обычно создаётся несколько альтернативных архитектур. В процессе работы с требованиями они детализируются и, соответственно, детализируется архитектура. В случае множественных альтернативных архитектур на определённом этапе анализа требований выбирается одна. Природа требований такова, что они не только выявляются, но ещё и изменяются. Изменение требований вносит изменения в детали архитектуры. Устойчивость архитектуры проявляется в незначительных её изменениях при добавлении, детализации или изменении требований. Если наступил момент, при котором появление новой информации о требованиях перестало оказывать влияние на архитектуру, значит, архитектура стабилизировалась.

Анализ проблемы

Анализ проблемы - это процесс осознания реальных проблем и потребностей пользователей, а также предложения решений, позволяющих удовлетворить эти потребности.

Проблема - это разница между желаемым и воспринимаемым.

Цель анализа проблемы состоит в том, чтобы добиться лучшего понимания решаемой проблемы до начала разработки. Для этого нужно пройти пять этапов: 1) достигнуть соглашения об определении проблемы; 2) выделить основные причины, т. е. подпроблемы, стоящие за проблемой; 3) выявить заинтересованных лиц и пользователей; 4) определить границу системы; 5) выявить ограничения, которые необходимо наложить на решение.

Элемент	Описание
Проблема	Описание проблемы
Воздействует на ...	Выявление лиц, на которых оказывает влияние данная проблема
Результатом чего является	Описание воздействия данной проблемы на заинтересованных лиц и бизнес-процесс
Выигрыш от ...	Указать предлагаемое решение
Может состоять в ...	Список основных преимуществ предлагаемого решения

2) Пример: компания занимается торговлей по каталогу через Интернет. Проблема: невысокая прибыль. Применив методику "качество во всём", было обращено внимание на "ущерб от несоответствия", который представляет собой стоимость всего, что идёт не так, как надо, и приводит к бесполезным затратам. Этот ущерб включает в себя: переделки, остатки, неудовлетворённость клиента, текучесть кадров и пр. Проанализировав ущерб от несоответствия, компания решила, что наибольший вклад в него вносят "остатки". Далее следует определить какие факторы (причины) влияют на наличие "остатков".

Остатки	
1. Устаревшие товары.	8%
2. Неправильные заказы на покупку.	50%
3. Повреждения при доставке.	20%
4. Производственные дефекты.	4%
5. Возвраты клиентами.	10%
6. Прочее.	3%

Иногда для выявления факторов (корневых причин) достаточно побеседовать с сотрудниками фирмы. Если проблема более серьёзная, то нужно провести специальное расследование.

Устранение корневых причин

Нет надобности устранять все корневые причины: влияние некоторых столь незначительное, что затраты на их устранение превысят результирующий эффект; следовательно, необходимо определить влияние каждой корневой причины.

После проведенного анализа предложение команды разработчиков заменить существующую систему ввода заказов на покупку можно считать аргументированной. После этого следует представить обоснование стоимости новой системы, сопоставить её с выгодой, полученной от её внедрения. Продолжая анализ, можно определить, какие типы ошибок вносят наибольший

вклад в проблему слишком больших "остатков".

3) Заинтересованные лица - это все, на кого реализация новой системы может оказать материальное воздействие. Для определения требований к проекту следует определить множество лиц, заинтересованных в его реализации. В это множество может войти не только заказчик и персонал, который будет эксплуатировать программный продукт, но также представители других организаций, например, налоговая служба, банк, система связи и т. д., которые будут взаимодействовать с организацией-заказчиком в рамках новой разработки.

Исполнитель (актор) - это сущность, обладающая поведением. К числу исполнителей может относиться и сама рассматриваемая система. Исполнителями являются не только люди, но и организации, машины и программы. Существует три типа внешних по отношению к разрабатываемой системе исполнителей: 1) основной исполнитель: его задача выполняется с использованием системы; 2) вспомогательный исполнитель: обслуживает систему, например, предоставляет информацию: служба авторизации платежей: этого исполнителя нужно идентифицировать для определения внешних интерфейсов и протоколов; 3) закулисный исполнитель: заинтересован в реализации функций системы, но не является ни основным, ни вспомогательным исполнителем, напр., налоговая служба. В зависимости от предметной области, выявление заинтересованных лиц может оказаться как тривиальной, так и нетривиальной задачей. Часто достаточно провести опрос среди тех, кто принимает решения в организации: 1) кто будет пользователем системы?; 2) кто является заказчиком (экономическим покупателем) системы?; 3) на кого ещё окажут влияние результаты работы системы?; 4) кто будет оценивать и принимать систему, когда она будет представлена и развёрнута?; 5) существуют ли другие, внешние или внутренние пользователи системы, чьи потребности необходимо учесть?; 6) кто будет заниматься сопровождением новой системы?; 7) не забыли ли Вы кого-нибудь?

Для интернет-магазина можно составить следующую табличку, из которой можно определить заинтересованных лиц:

Пользователи	Другие заинтересованные лица
Служащие, занимающиеся вводом заказов	Администратор ИС Главный финансист Управляющий производством
Руководитель отдела приёма заказов	
Контролёр производства	
Служащий, выписывающий счета	

4) Для определения границ системы необходимо поделить наш мир на две части: 1) наша система; 2) всё, что взаимодействует с нашей системой. То, что взаимодействует с нашей системой, относится к общей группе "акторы". Актор - это нечто, находящееся вне системы и взаимодействующее с системой. Во многих случаях границы системы очевидны, однако если система (подобно рассматриваемому примеру) встраивается в уже существующую систему, то выявление границ требует определенных усилий.

Клерк, ввод заказов на покупку	Новая система обработки заказов	Существующая ИС
Руководитель отдела приёма заказов	Клерк, выписка счетов	Контролёр производства
		Границы системы

5) Ограничения уменьшают степень свободы, который мы располагаем при принятии решений. Существуют различные источники ограничений.

Источник	Образцы вопросов
Экономический	Какие финансовые или бюджетные ограничения следует учесть? Существуют ли соображения, касающиеся себестоимости и ценообразования? Существуют ли вопросы лицензирования?
Политический	Существуют ли внешние или внутренние политические вопросы, влияющие на принятие решения?

Технический	Системный	Эксплуатационный	График и ресурсы
Существуют ли проблемы в отношениях между подразделениями? Существуют ли ограничения на выбор технологий? Нужно ли работать в рамках существующих платформ или технологий? Запрещено ли использование любых новых технологий? Должны ли мы использовать какие-либо закупаемые пакеты ПО?	Будет ли решение приниматься для наших ранее разработанных систем? Должны ли мы обеспечивать совместимость с существующими решениями? Какие операционные системы и среды должны поддерживаться?	Существует ли ограничение информационной среды или правовые ограничения? Существуют ли юридические ограничения? Существуют ли требования безопасности? Какими другими стандартами мы ограничены?	Определённый график? Ограничены ли мы существующими ресурсами? Можем ли мы привлечь работников со стороны? Можем ли мы увеличить штат (временно или постоянно)?

После того, как ограничения выявлены, некоторые из них станут требованиями к новой системе. Например, для компании торговли через интернет выявлены следующие ограничения:

Источник	Ограничения	Пояснения
Эксплуатационный	Копия данных заказа на покупку должна оставаться в унаследованной БД в течение одного года	Риск потери данных слишком высок. Необходима беспере-
Системы и ОС	Приложение должно занимать на сервере не более 200 МБ	Количество доступной памяти ограничено
Средства, выделенные на оборудование	Система должна быть разработана на существующем сервере. Для пользователей можно предложить новое клиентское АО	
Средства, выделенные на оплату труда	Фиксированный штат, не привлекать новых работников со стороны	Фиксированные расходы на зарплату по отношению к текущему бюджету
Технические ограничения	Должна использоваться объектно-ориентированная методология	Надеемся

Функции программного продукта и потребности заинтересованных лиц и пользователей

Команда разработчиков может создать хорошую систему только в том случае, если она понимает реальные потребности заинтересованных лиц. Потребность - это отражение некой лично, рабочей или бизнес-проблемы, решение которой оправдывает замысел, покупку или использование нового программного продукта. Говоря о потребностях, пользователи обычно представляют их в виде функций. Недостатком подхода, основанного только на функциях, является то, что часто непонятно, какая потребность стоит за функцией, что может привести

к ошибкам при создании программного продукта.  
 Функция - это обслуживание, предоставляемое системой для выполнения одной или нескольких потребностей заказчика. Примеры функций: прикладная область: система управления элеватором, функция: осуществлять ручную управление дверью при угрозе пожара; прикладная область: система управления запасами, пример функции: предоставлять свежую информацию о состоянии всех инвентарных единиц; система обработки платёжных ведомостей: сообщать текущие начисления по категориям.  
 Количество функций рекомендуется от 25 до 99.  
 Атрибуты функций программного продукта  
 Атрибуты функций обеспечивают дополнительную информацию о каждой функции. Присваивая функциям различные атрибуты, можно успешно управлять сложностью предоставления информации.

Атрибут	Описание
Статус	Отслеживает ход процесса определения базового уровня проекта
Приоритет/полезность	Функции различаются своей важностью, нуно определять их масштаб и очерёдность: критическое, важное, полезное
Трудоёмкость	Желательно записать этот атрибут в виде человекочасов или командочасов (командонедель)
Риск	
Стабильность	Вероятность того, что данная функция будет меняться или будет меняться её понимание командой; обычно приводятся качественные характеристики: высокая/средняя/низкая.
Целевая версия	Версия продукта, в которой впервые появится данная функция. Комбинирование этого атрибута с атрибутом статуса позволяет команде предлагать, записывать или обсуждать различные функции, не прибегая к их разработке
Назначение	Во многих проектах функции будут приписываться "функциональным командам", ответственным за их дальнейшую доработку, написание программных требований, а также, возможно, и реализацию
Обоснование появления	Используется для отслеживания источника функции, например, ссылка может указывать на номер строки спецификации продукта, или видеомаркер на важном интервью с клиентом

**Выявление требований**  
 Проблемы, связанные с выявлением требований:  
 1) проблема непонимания пользователем того, что будет представлять собой разрабатываемая система: пользователь никогда раньше не видел новую систему, не понимает её описание, структуру и т. д.: в других областях техники веками сложилась технология моделей: чертежи, физические макеты, действующие прототипы, пробные промышленные изделия; все они осязаемы и действуют аналогично разрабатываемому устройству;  
 2) проблема новых требований: чем больше найдено требований, тем лучше известно, что это ещё не всё;  
 3) проблема общения пользователя и разработчика. Синдром пользователя и разработчика заключается в том, что пользователь и разработчик обычно принадлежат к разным мирам, говорят на разных языках, имеют различный опыт, мотивацию и цели.

Проблема	Решение
Пользователи не знают, чего они хотят, а если знают, то не могут сформулировать	Назначить пользователя экспертом предметной области, ценить его в этом положении, использовать альтернативные методы общения и выявления требований
Пользователи думают, что они знают, чего	Как можно раньше предлагать альтернативные методы выявления:

хотят, до тех пор, пока разработчики не предоставят им то, что они якобы хотели  
 Аналитики думают, что они понимают проблему пользователя лучше него самого  
 Все считают, что другие руководствуются политическими мотивами

Методы выявления требований  
 Существует множество методов выявления требований, основные: интервьюирование и анкетирование, совещание, посвящённое требованиям, мозговой штурм и отбор идей, описание прецедентов, раскадровки, обыгрывание ролей, создание прототипов. Выбор конкретного метода (методов) зависит от типа приложения, опыта и уровня подготовки команды разработчиков, заказчика, масштаба проблемы, критичности приложения, его уникальности и используемой технологии.  
 Интервьюирование - это форма очного проведения опроса, при котором исследователь находится в непосредственном контакте с респондентом. Этот метод предпочтительнее анкетирования исходя из следующих соображений: 1) при таком методе вопросов без ответов практически не бывает; 2) неопределённые или противоречивые ответы могут быть уточнены; 3) имеется возможность наблюдения за респондентом и фиксации не только его вербальных ответов, но и невербальных реакций. Недостатки: 1) малая оперативность; 2) существенные затраты времени; 3) необходимость большого числа интервьюеров; 4) высокие требования к подготовке аналитика: для начинающих аналитиков этот метод представляет большие трудности, поскольку требует специальной подготовки и длительного тренинга.  
 Контекстно-свободные вопросы  
 Нужно избежать предубеждения пользователя при ответах на вопросы. Для этого нужно задавать вопросы о природе проблемы пользователя, никак не связывая их с возможным решением. Для этого Гаусс и Вайнберг ввели понятие "контекстно-свободный вопрос". Примеры таких вопросов: "Кто является пользователем?", "Кто является клиентом?", "Отличаются ли их потребности?", "Где ещё можно найти решение данной проблемы?". Эти вопросы позволяют выслушать заказчика прежде, чем пытаться предложить потенциальное решение.  
 Добавление контекста  
 После ответов на контекстно-свободные вопросы полезно сместить вопросы в область исследования решений, обсуждение предлагаемых решений поможет пользователю углубить или даже изменить взгляд на проблему, найти ещё не обнаруженные требования.  
 Образец интервью  
 Часть I: определение профиля заказчика или пользователя  
 Компания  
 Отрасль  
 Должность  
 Фамилия  
 Каковы Ваши основные обязанности?  
 Что Вы в основном производите?  
 Для кого?  
 Как измеряется успех Вашей деятельности?  
 Какие проблемы влияют на успешность Вашей деятельности?  
 Какие тенденции (если такие существуют) делают Вашу работу проще или сложнее?  
 Часть II: оценка проблемы  
 Для каких проблем (прикладного типа) Вы ощущаете нехватку хороших решений? Назовите их (ещё вопросы).  
 Для каждой проблемы выясните следующее: почему существует эта проблема, как она решается в настоящее время, как заказчик (пользователь) хотел бы её решить.  
 Часть III: понимание пользовательской среды  
 Кто такие пользователи?  
 Какое у них образование?  
 Каковы их навыки в компьютерной области?  
 Имеют ли пользователи опыт работы с приложениями подобного типа?  
 Какая платформа используется?  
 Каковы планы относительно будущих платформ?  
 Используются ли дополнительные приложения, которые имеют отношение к данному?  
 Если используются, то пусть о них немного расскажут.

раскадровку, прототипы, ролевые игры и т. д.  
 Поставить аналитика на место пользователя, провести игру в течение часа или целого дня  
 Такова человеческая натура, тут ничего не изменишь

Каковы ожидания заказчика относительно практичности продукта? Сколько времени необходимо на обучение?  
 В каком виде должна быть представлена справочная информация для пользователя?  
 Часть IV: резюме (перечисляются основные пункты, чтобы проверить, правильно ли всё понято)  
 Итак, Вы сказали мне, что...  
 Адекватно ли этот список передаёт проблемы, которые имеют в настоящее время?  
 Какие ещё проблемы Вы испытываете?  
 Часть V: предложения аналитика относительно проблемы заказчика  
 Выяснить, какие проблемы, если они есть, связаны с (перечислите все потребности или дополнительные проблемы, которые по Вашему мнению может испытывать заказчик или пользователь; для каждой из этих проблем выясните следующее: является ли она реальной, каковы её причины, как она решается в настоящее время, как бы заказчик или пользователь хотел её решить, насколько важно для заказчика или пользователя решение этой проблемы в сравнении с другими упомянутыми им).  
 Часть VI: оценка предлагаемого Вами решения (если это уместно)  
 Охарактеризовать основные возможности предлагаемого Вами решения, а потом задавать пользователю следующие вопросы:  
 Что, если Вы сможете...  
 Как Вы расцениваете важность...  
 Часть VII: оценка возможностей  
 Кто в организации нуждается в данном приложении?  
 Сколько пользователей указанных типов будут использовать его?  
 Насколько значимо для Вас успешное решение?  
 Часть VIII: оценка необходимого уровня надёжности и производительности, а также потребности в сопровождении  
 Каковы Ваши ожидания относительно надёжности?  
 Какой, по-Вашему, должна быть производительность?  
 Будете ли Вы заниматься поддержкой продукта или этим будут заниматься другие?  
 Испытываете ли Вы потребность в поддержке?  
 Что Вы думаете о доступе для сопровождения и обслуживания?  
 Каковы требования относительно безопасности?  
 Каковы требования относительно установки и конфигурации?  
 Существует ли специальное требование по лицензированию?  
 Как будет распределено ПО?  
 Есть ли требования по маркетингу и упаковке?  
 Часть IX: другие требования  
 Существуют ли законодательные требования, требования информационной среды, инструкции или другие стандарты, которых необходимо придерживаться?  
 Нет ли других требований, о которых нам следовало бы знать?  
 Часть X: окончание интервью  
 Существуют ли другие вопросы, которые мне следовало бы задать?  
 Если мне понадобится задать Вам несколько вопросов, могу ли я Вам позвонить?  
 Будете ли Вы принимать участие в обсуждении требований?  
 Часть XI: заключение аналитика  
 После интервью, пока его данные свежи в Вашей памяти, зафиксируйте наиболее важные проблемы с наиболее высшими приоритетами, выявленные в беседе.  
 Подготовка к проведению интервью  
 Подготовьте интервью и запишите его в записную книжку, просмотрите вопросы непосредственно перед беседой. Перед интервью соберите информации о клиенте, о его компании. Не задавайте вопросы, ответы на которые можно узнать заранее, однако полезно кратко сверить эти ответы. Кратко записывайте ответы в записную книжку, не пытайтесь в это время сохранять данные в электронном виде.  
 Интервьюер должен стараться, чтобы сценарий не был слишком увлечён. Если клиент увлечён, не перебивайте его следующим вопросом.  
 Уже после нескольких интервью исследователь получает некоторые знания о предметной области, понимание решения проблемы, а также представления пользователя об её успешном решении. Если проблемы начинают повторяться, это значит, что исследователь нашёл общие потребности и дальше можно приступать к уточнению функций.  
 Анкетирование - это вербально-коммуникативный метод, в котором в качестве средств сбора сведений от респондента используется специально оформленный список вопросов - анкета. В отличие от интервьюирования, контакт аналитика с респондентом здесь сведен к минимуму. Анкетирование позволяет наиболее жестко следовать намеченному плану исследования, поскольку процедура "вопрос - ответ" строго регламентирована. При помощи анкетирования можно с наименьшими затратами получить высокий уровень массовые исследования. Особенностью метода является его анонимность,

то есть личность респондента не фиксируется. Анкетирование в основном проводится в тех случаях, когда необходимо выяснить мнение массы людей по некоторым вопросам за короткий срок. Достоинства анкетирования:

- 1) высокая оперативность получения информации;
- 2) жёсткая регламентация процедуры позволяет получить хорошо структурированные и сопоставимые результаты;
- 3) анонимность респондентов повышает объективность и искренность получаемых ответов;
- 4) возможность организации массовых обследований и сбора большого объёма данных;
- 5) сравнительно малая трудоёмкость процедур подготовки, проведения исследований и обработки результатов;
- 6) отсутствие влияния личности и поведения опрашиваемого на работу респондентов.

Недостатки анкетирования:

- 1) отсутствие личного контакта не позволяет изменить порядок и формулировки вопросов в зависимости от ответов и поведения респондентов;
- 2) невозможность корректности понимания респондентом формулировок вопросов и ответов, а также получения респондентом пояснений в случае непонимания;
- 3) невозможность гарантировать точное выполнение инструкций респондентом, например, соблюдение порядка ответов на вопросы;
- 4) возможное влияние формулировок вопросов и ответов на выбор респондента;
- 5) возможная пристрастность респондента;
- 6) при заочном анкетировании невозможно гарантировать, что респондент заполняет анкету самостоятельно, без влияния других лиц.

Способы формирования вопросов:

- 1) многоальтернативные вопросы - при такой форме вопроса допускаются комментарии респондента в свободной форме;
- 2) рейтинговые вопросы - представляют predetermined набор ответов на сформулированные вопросы, например, могут использоваться такие значения, как "Абсолютно согласен", "Согласен", "Отношусь нейтрально", "Не согласен", "Абсолютно не согласен", "Не знаю";
- 3) вопросы с ранжированием - предусматривают упорядочивание ответов путём присваивания им порядковых номеров, процентов, значений и т. п. Анкетирование не может заменить интервьюирование, и хотя анкетирование проводится значительно быстрее и его результаты поддаются автоматизированной обработке, интервьюирование способно дать значительно более ценные результаты.

Совещание, посвящённое требованиям

Совещание является самым мощным методом выявления требований. Совещания созываются для достижения консенсуса в вопросах определения требований к приложению и быстрого принятия решения о том, в каком направлении действовать. Хорошо проведенное совещание по вопросам требований имеет следующие преимущества: 1) оно позволяет создать команду, подчинённую одной общей цели - успеху данного проекта; 2) все заинтересованные лица получают возможность высказать своё мнение; 3) оно формирует соглашение между заинтересованными лицами и командой по поводу того, что должно делать приложение; 4) оно может высветить и разрешить политические вопросы, которые влияют на успех проекта; 5) результат в виде предварительного определения системы на уровне функций немедленно становится известными.

Подготовка к совещанию

Подготовка к совещанию - это залог успеха мероприятия. Прежде всего следует убедить членов команды в полезности совещания, затем нужно выявить всех важных заинтересованных лиц.

Логистика

Логистика, или организация мероприятия, включает в себя всё - от создания приглашения участникам до организации приезда участников и освещения помещения. Если подойти к логистике с высокой степенью профессионализма, то для участников будет очевидно, что совещание - это действительно важное событие и они будут вести себя соответственно, в результате совещание будет успешным.

Подготовительные материалы

Необходимо заранее разослать подготовительные материалы для того, чтобы подготовить участников, а также производительность проводимого совещания. Подготовительные материалы должны стимулировать как конкретное, так и свободное мышление. Рекомендуется представить два вида подготовительных материалов: 1) информация, относящаяся к данному конкретному проекту: планы документов, которые будут содержать требования, списки предлагаемых функций, доклады аналитиков о наблюдаемых в отрасли тенденциях, доклады об ошибках существующей системы, новые данные маркетинга и т. д. Нужно не перегружать участников совещания данными и вместе с тем представить им необходимую информацию; 2) информация для подготовки свободного мышления: предлагаются статьи,

стимулирующие процесс творчества, правила проведения "мозгового штурма", разработки требований, управления масштабом и т. д. Нельзя посылать данные слишком рано (а также слишком поздно), рекомендуется за 3-7 дней. Образец уведомления:

Уведомление

Кому:

Предмет: совещание, посвящённое требованиям (каким-то)

От: я являюсь администратором проекта <Название>, проект инициирован <Дата> и будет завершён <Дата>. Как и в большинстве проектов, оказалось трудно достигнуть консенсус по новым функциям (каким?) данного приложения, которые удовлетворяют потребностям разнородной группы участников. Чтобы упростить данный процесс, мы проведём совещание, посвящённое (вот этим) требованиям, которое состоится <Дата>.

Цель: целью совещания является окончательное определение новых функций и т. д. Совещание будет проводить: <ФИО>, который обладает значительным опытом в сфере разработки требований.

Результаты: результаты совещания будут известны немедленно и будут доведены до групп разработчиков и маркетинга на следующий день. Мы приглашаем Вас принять участие в совещании и внести свои предложения, которые будут представлять потребности вашего (чего?) клиента, отдела, команды и т. д. Если вы не можете приехать, мы настоятельно рекомендуем Вам прислать члена Вашей команды, который уполномочен принимать решения, представляющие Ваши потребности. Вместе с этим уведомление мы высылаем Вам краткое описание функций продукта, а также некоторые справочные материалы касательно мозгового штурма. Совещание начнётся в 08:30 и будет продолжаться до 17:30. Надеемся [...], искренне Ваш, [...].

Роль ведущего

Рекомендуется, чтобы совещание проводилось сторонним человеком, имеющим опыт в решении задач управления требованиями. Если такого человека нет, то можно это поручить члену команды, обладающему следующими качествами:

- 1) обучен данному процессу;
- 2) продемонстрировал способности в достижении консенсуса;
- 3) его авторитет признаётся как внутренними, так и внешними членами команды;
- 4) достаточно энергичен для того, чтобы руководить совещанием. Ведущий не должен вносить свои идеи и участвовать в обсуждении. Ведущий играет ключевую роль, потому что возможно, Вам не представится повторно собрать совещание.

Некоторые обязанности ведущего:

- 1) задать профессиональный и объективный тон встречи;
- 2) начать и закончить совещание вовремя;
- 3) предложить цели встречи и повестку дня;
- 4) управлять течением дискуссии и удерживать команду "на правильном пути";
- 5) способствовать процессу принятия решения и достижения консенсуса, но избегать участия в содержательной части дискуссии;
- 6) заниматься всеми организационными вопросами и вопросами логистики, чтобы основное внимание было сконцентрировано на повестке дня;
- 7) удостовериться, что все заинтересованные лица участвуют и их пожелания учтены;
- 8) контролировать поведение, которое ведёт к расколу или мешает продуктивной работе.

Составление повестки дня совещания

Повестка дня зависит от потребностей конкретного проекта

Время	Пункт повестки дня	Проводимые мероприятия
8:00-8:30	Открытие	Повестка дня, организация, правила
8:30-10:00	Обсуждение контекста	Состояние проекта, потребности рынка, результаты интервью, ...
10:00-12:00	Мозговой штурм	Интенсивное обсуждение функций приложения
12:00-13:00	Обеденный перерыв	
13:00-14:00	Мозговой штурм	Продолжение
14:00-15:00	Определение функций	Запись определений функций в 2-3 вариантах
15:00-16:00	Отбор идей и расстановка приоритетов	Расстановка приоритетов функций
16:00-17:00	Завершение	Подведение итогов и др. действия

Проведение совещания

Подобные совещания часто характеризуются весьма напряжённой атмосферой. Аудитория может быть политически предубеждённой и враждебной (позтому нужен внешний ведущий). Ведущий должен внести некоторые правила

проведения совещания: каждый участник изначально получает один билет на опоздание; истратив его, он вносит 20 грн. в штрафную коробку; каждый участник изначально получает один билет на право "пнуть" одного участника или отдел ("бесплатный выстрел"), истратив его, он вносит 20 грн. в штрафную коробку (предостережение от политических выступлений); каждый имеет право на пятиминутное выступление: ведущий предоставляет ему слово и устанавливает таймер, все слушают и никто не перебивает (создание структурированного процесса представления слова).

Проблема: трудно восстановить работу после перерыва, основные участники опаздывают. Решение: у ведущего должен быть таймер, ввести правило штрафов

Проблема: доминирующие позиции отдельных участников. Решение: ведущий предлагает билеты "пятиминутное выступление".

Проблема: недостаток предложений от участников. Решение: ведущий предлагает билеты "пятиминутное выступление". Он объявляет, что ни один участник не должен покинуть совещание, не использовав свои билеты и не получив свой билет "замечательная идея" от других.

Проблема: негативные комментарии, мелочное поведение и скрытые враждебные действия. Решение: использование билетов "бесплатный выстрел".

Проблема: ослабление энергии после обеда. Решение: лёгкий ланч, лёгкие закуски во время перерывов, изменение оформления помещения, изменение освещения, возможно, температуры и т. д.

Результат и продолжение

После завершения совещания ведущий раздаёт его протоколы и записывает другие итоги. На этом его работа заканчивается и ответственность за успех вновь передаётся команде разработчиков. Руководитель проекта несёт ответственность за то, чтобы завершить всё начатое на встрече и организовать рассылку информации участникам.

Мозговой штурм и отбор идей

Мозговой штурм очень полезен, когда нужны новые идеи или творческие решения. Применяется, когда участники собираются вместе. Преимущества мозгового штурма:

- 1) поддерживает участие всех присутствующих;
- 2) позволяет участникам развивать идеи друг друга;
- 3) ведущий ведёт запись всего хода обсуждения;
- 4) его можно применить при различных обстоятельствах;
- 5) как правило, в результате получаем множество решений для любой поставленной проблемы;
- 6) метод способствует свободному мышлению, не ограниченному обычными рамками.

Мозговой штурм состоит из двух основных фаз: генерация идей и их отбор. Цель первой фазы: получить как можно больше идей, не обязательно глубоких. Вторая фаза предусматривает анализ и отбор идей.

Организация "живого" мозгового штурма

Все основные участники собираются в одном помещении, каждый готовит материал - листки заданного формата, и чёрный толстый маркер. Могут пригодиться кнопки и большая мягкая поверхность (например, доска). Затем объясняется правило проведения мозгового штурма: 1) не допускаются критика или дебаты; 2) надо дать свободу фантазии; 3) нужно генерировать как можно больше идей; 4) можно переделывать и комбинировать идеи.

Ведущий объясняет цель процесса, затем предлагает участникам:

"Высказывайте свои идеи и записывайте их на листках". Идеи оглашаются громко, чтобы все могли развивать их или предлагать связанные с ними идеи и. Каждый участник записывает свои идеи на этих листочках, чтобы гарантировать их сохранность; чтобы позволить их развитие, чтобы уменьшить задержки процесса. Никакой критики; поощряются замечания типа "отличная идея". Обычно процесс длится 1 час, редко - 2-3 часа. Ведущий не должен прекращать процесс, если он идёт активно. Обычно число идей - 100-200.

Отбор идей

Отбор идей наступает после фазы генерации с состоит из нескольких этапов: ведущий спрашивает у участников, заслуживает ли конкретная идея дальнейшего рассмотрения. Если участники единодушны, то идея удаляется. Две одинаковые идеи объединяются (лучше, чем удалять одну идею).

Группировка идей

Полезно по ходу процесса группировать аналогичные идеи. Удобно, когда участники совещания подходят к доске и переставляют идеи. Группам можно давать названия по принципу группировки: новые функции, улучшение производительности, интерфейс пользователя и т. д.

Определение функций

Пример: область приложения: автоматическое управление освещением; идея: автоматическая установка освещения. Функция: домовладелец может предварительно задавать последовательности возникновения определённых осветительных событий в зависимости от времени дня.

Расстановка приоритетов

**Способы:**

- 1) накопительное голосование: каждому участнику совещания условно выдаётся некоторая сумма. Участник совещания раздаёт части этой суммы определённым идеям по своему усмотрению;
  - 2) разбиение идей на категории: критический, важные, полезные
- Наблюдение**  
 Наблюдение - это полезная стратегия получения информации о предметной области, основанная на детальном изучении работы моделируемой организации. Различают пассивное наблюдение и активное наблюдение. При пассивном наблюдении исследователь не участвует непосредственно в работе изучаемой системы, при активном - участвует, поэтому активное наблюдение скорее можно отнести к категории обыгрывания ролей.

**Достоинства наблюдения**  
 Через наблюдение аналитик получает информацию о происходящих ситуациях день за днём из первых рук. Во время наблюдения часто выявляются проблемы, которые невозможно выявить, изучая документы или проводя интервью.

**Недостаток наблюдения**  
 Наблюдатель, как и всякий "измерительный инструмент", вносит помехи в результат измерения, то есть сотрудники организации, находясь "под колпаком", могут вести себя не так, как они это делают обычно.

**Обыгрывание ролей**  
 С помощью обыгрывания ролей команда разработчиков может почувствовать мир пользователя, поставив себя на его место. Основания для применения это метода:

- 1) многие пользователи не могут описать процедуры, которые они выполняют, или потребности, которые необходимо учесть. Это не входит в их задачу и иногда оказывается значительно сложнее, чем кажется;
- 2) некоторые пользователи применяют нестандартные и иногда уникальные способы выполнения рабочих действий, которые могут скрывать реальные проблемы от наблюдателя;
- 3) ни один разработчик не в состоянии предвидеть все вопросы, которые необходимо задать, и ни один пользователь не знает, какие вопросы должен задать разработчик.

Есть методы, похожие на обыгрывание ролей:

- 1) сценарный просмотр: При сценарном просмотре каждый участник следует сценарию и ему соответствует конкретная роль "в пьесе". При просмотре будут продемонстрированы любые неточности в понимании ролей, недостаток информации, доступной актёру или подсистеме, недостаток описания поведения необходимого актёра для успешного выполнения ролей.

**Достоинства метода**  
 Сценарий можно модифицировать и снова проигрывать, пока актёры не сочтут его правильным.

CRC			
Объект	Инспектор	Территория	
Список безработных			
Обязанности	Обязанности:	Обязанности:	
Добавить безработного [Взаимодействие]	1. Проверить адрес	1. Проверить адрес	1.
Исключить безработного	2. Проверить ФИО	2. Включить улицу	2.
Найти безработного	3. Установить специальность	3. Включить дом	3.
		4. Исключить улицу	...
	...		5. Исключить дом
Инспектор	Территория		
	Список безработных	Инспектор	

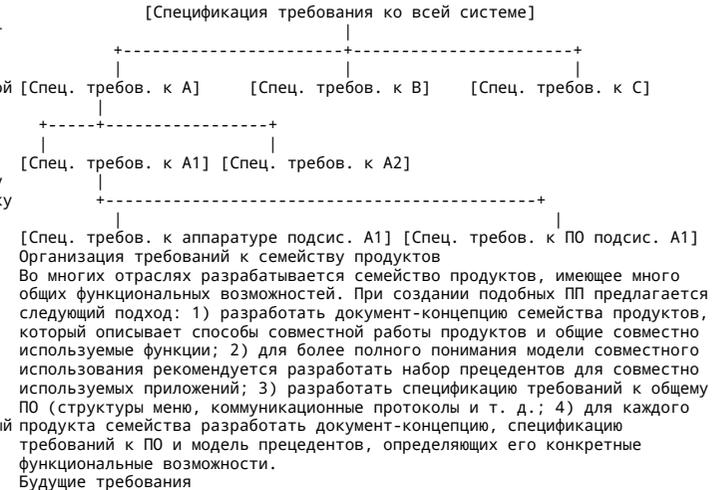
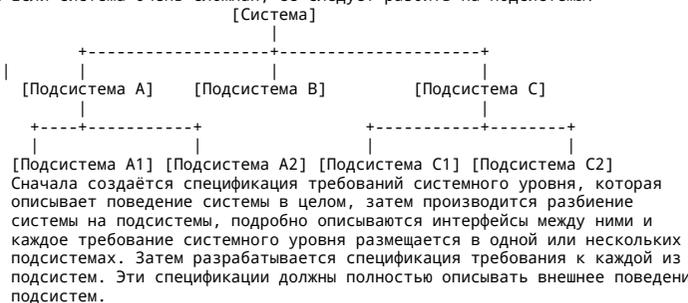
**Раскадровка**  
 Цель раскадровки - получение ранней реакции пользователей на предложенные концепции приложения. Раскадровки делятся на три типа в зависимости от способа взаимодействия с пользователем: 1) пассивные: представляют собой историю, которая рассказывается пользователю; это могут быть схемы, картинка, копии экрана, презентации PowerPoint; аналитик играет роль системы и проводит пользователя по раскадровке; 2)

активные раскадровки: создаются с помощью анимации, автоматического показа слайдов или фильма; 3) интерактивные: они дают пользователю опыт общения с системой, подобный реальной работе; они могут быть имитационными (в виде макета) либо использовать код. Выбор типа раскадровки зависит от сложности системы: для совершенно новой и сложной системы может оказаться полезным проводить разные виды раскадровок несколько раз. В простейшем случае раскадровка - это ряд сделанных от руки рисунков или схем. Советы по раскадровке: 1) не вкладывайте много средств в раскадровку: если она будет выглядеть как реальный продукт, клиенты будут опасаться вносить в неё изменения; 2) делайте раскадровку такой, чтобы её можно было менять; 3) если возможно, делайте раскадровку интерактивной; 4) создавайте раскадровки как можно раньше и делайте их чаще; 5) применяйте раскадровки во всех проектах, имеющих новое или новаторское содержание.

**Создание прототипов**  
 Программные прототипы демонстрируют ряд функциональных возможностей системы. Пользователи могут видеть и взаимодействовать с прототипом, чего не может обеспечить ни один другой метод. Прототип требований к программному обеспечению - это частичная реализация программного продукта, созданная с целью помочь разработчикам, пользователям или клиентам лучше понять требования к системе. Прототипы принято подразделять на горизонтальные и вертикальные. Горизонтальный прототип создаёт широкий спектр функциональных возможностей системы. Вертикальный прототип реализует довольно мало требований, но делает это качественно. Назначение прототипа: 1) используется для получения подтверждения от заказчика о том, что разработчик правильно понимает требования; 2) играет роль катализатора, заставляющего заказчика подумать о дополнительных требованиях; 3) помогает донести требования до разработчика.

**Организация информации о требованиях**  
 Проблемы организации информации о требованиях  
 В процесс создания системы вовлечены многие люди: разработчики, пользователи, заинтересованные лица, аналитики, специалисты по архитектуре и т. д. Все они должны прийти к одному соглашению о том, какая должна быть система. Документ, с которым могут свериться и на который могут ссылаться участники, называется спецификацией требований. Редко удаётся определить требования с помощью только одного документа. Причины: система может быть очень сложной; потребности заказчика могут быть документированы до того, как производится документирование подробных требований; система может быть членом семейства родственных продуктов; система может реализовывать только некоторое подмножество выявленных требований; цели маркетинга и бизнеса следует отделить от подробных требований к продукту. Несколько специальных случаев, когда необходимы несколько документов: 1) некоторый "родительский" документ определяет требования ко всей системе (ПО, аппаратура, персонал и т. д.), а другой - только к ПО; тогда первый называется спецификацией требований к системе, а второй - спецификацией требований к ПО; 2) один документ содержит общее определение функций системы, а второй - более конкретное; тогда первый называется документом-концепцией, а второй - спецификацией требований к программному обеспечению; 3) один документ содержит полный набор требований к семейству продуктов, в другом представлены требования к конкретному приложению (версии); тогда первый называется "требования к семейству продуктов", а второй - "спецификация требований к ПО"; 4) один документ описывает общие бизнес-требования и бизнес-окружение, в котором будет существовать система, а другой определяет её внешнее поведение; тогда первый называется документом бизнес-требований или документом требований маркетинга, а второй - спецификацией требований к ПО.

**Организация требования к сложным аппаратным и программным системам**  
 Если система очень сложная, её следует разбить на подсистемы:



**Будущие требования**  
 Среди требований есть такие, которые не включают в данную реализацию системы, однако не следует их полностью выбрасывать. Желательно их сохранить для будущих версий.

**Отличие бизнес-требований и требований маркетинга от требований к продукту**  
 Планирование нового продукта не производится в чисто технической среде. Надо подумать о рынках, <...> продукта, каналах распределения, функциональных возможностях, затратах на маркетинг, доступности ресурсов, возможности возмещения затрат путём продажи большого количества копий и т. д. Эти соображения следует задокументировать, однако они не входят в спецификацию требований. Иногда для этого используют документ требований маркетинга. Для составления этого документа необходимо ответить на следующие вопросы: кто является заказчиком, кто является пользователем, на каких рынках предполагается продавать продукт, как поделены эти рынки, различаются ли требования пользователей в различных сегментах рынка, какие классы пользователей существуют, какие потребности удовлетворяет продукт, в чём заключаются основные преимущества продукта, почему его будут покупать, что собой представляют конкуренты, в какой среде будет использоваться система, какими будут затраты на разработку, по какой цене предполагается продавать продукт, как будет выполняться установка, дистрибуция и сопровождение продукта.

**Диспетчеризация полиграфического производства**  
 Некоторая типография "ВМВ" выполняет заказы от частных лиц или организаций. Приём заказов осуществляют 10 менеджеров продаж. У каждого заказа имеется плановый срок выполнения. В каждом заказе может быть одно или несколько изданий, напр., плакат, визитка, рекламный буклет, книжка, журнал, этикетка, коробка. Длительность выполнения заказ составляет от 1 до 10 дней (в среднем - 3 дня). Для производства каждого издания требуется выполнить ряд работ, таких, как: редактирование, дизайн, вёрстка, изготовление плёнок, печать, нарезка, склеивание, сортировка листов, упаковка. В момент передачи заказа производству для каждой из работ известна её плановая продолжительность. Работы должны выполняться в определённом порядке. Часть работ выполняется на полиграфическом оборудовании, часть - вручную. Некоторые единицы оборудования являются взаимозаменяемыми. На предприятии имеется 5 производственных участков, по которым группируется оборудование. Требуется разработать информационную систему, позволяющую автоматизировать работу диспетчера типографии. В функции диспетчера входит: назначение точного срока начала и окончания выполнения работ над заказом; составление графиков работы оборудования и сменного персонала; планирование работ в производственных участках; информирование менеджеров продаж о текущем состоянии их заказов в производстве; контроль исполнения и оперативная корректировка планов; учёт брака. Основная цель разработки: добиться бесперебойной работы типографии, исключив ситуацию срыва плановых сроков исполнения заказов.

Документ - концепция (видение) работы по формированию видения продукта и границ проекта, обычно начинается на самой ранней фазе проекта, до начала широкомащтабных консультаций по выявлению подробных требований. Понятие видения широко употребляется в бизнес-анализе. Если компания имеет представление о том, какие ключевые цели, сегменты рынка, товарные

позиции, прибыль должны быть достигнуты, допустим, через 5 лет, значит, компания имеет долгосрочное видение себя на рынке. Такой опыт формирования видения во многом переносится на процесс разработки программного продукта. Нужно увидеть "как программный продукт впишется в организационные процессы предприятия, какие ключевые выгоды он даст, какие проблемы позволит разрешить". Документ-концепция - это важнейший документ программного проекта, который фиксирует потребности пользователя, функции системы и другие общие требования к проекту. Он описывает на высоком уровне абстракции проблему и её решение. Концепция также служит основой для достижения согласия между основными внутренними сообществами заинтересованных лиц, а именно: 1) отделом маркетинга, который выступает в качестве доверенного лица заказчика и пользователя и отвечает за успех продукта после его реализации; 2) командой проекта, разрабатывающей приложение; 3) руководством, которое несёт ответственность за бизнес-результат. Есть несколько способов формирования документа-концепции, от которой зависит его содержание и последовательное создание.

Концепция в ГОСТ: после выявления требований к системе выполняется этап разработки требований к системе. Основные работы этого этапа: 1) изучение объекта; 2) проведение научно-исследовательских работ; 3) разработка вариантов концепции автоматизированной системы; 4) оформление отчёта о выполненной работе.

Видение в рамках MSF (методология разработки программных продуктов фирмы Microsoft): согласно "Белой книге" MSF, на фазе разработке концепции закладывается одна из фундаментальных основ успеха проекта, создания и сплочения проектной группы на основе выработки единого видения. Проектная группа должна чётко представить себе, что она хочет сделать для заказчика, и сформулировать свою цель таким образом, чтобы максимально мотивировать как заказчика, так и саму проектную команду. Выработка высокоуровневого взгляда на цели и условия проекта может рассматриваться как ранняя форма планирования. Она подготавливает почву для процессов создания детальных планов, которые будут выполняться непосредственно во время фазы планирования.

Видение RUP:  
Построение документооборота для типографии.

Видение (концепция)

1. Введение.

1.1. Цель.

Цель создания этого документа состоит в том, чтобы собрать, проанализировать и определить высокоуровневые потребности и возможности системы диспетчеризации типографии. Документ акцентирует внимание на возможностях, необходимых совладельцу и целевым пользователям, и на том, почему эти потребности существуют. Подробности того, как система диспетчеризации типографии выполняет эти потребности, будут детализированы в прецедентах и дополнительных спецификациях.

1.2. Контекст.

Настоящий документ разрабатывается в рамках проекта автоматизации деятельности типографии BMB.

1.3. Определение и сокращения.

Основные определения приведены в документе "Приложение 1. Глоссарий".

1.4. Ссылки.

1.5. Краткое содержание.

Документ описывает высокоуровневые требования к системе диспетчеризации полиграфического предприятия. Указаны основные деловые преимущества рассматриваемого решения, сформулированы ключевые проблемы и способы их решения, приведены характеристики пользователей системы, возможности системы, ограничения, показатели качества и другие требования к продукту.

2. Позиционирование.

2.1. Деловые преимущества.

В настоящее время диспетчеризация в типографии BMB осуществляется на основе использования электронных таблиц. В сравнении с существующим, новое решение позволит обеспечить более удобный режим доступа заинтересованных лиц к информации, повысить быстродействие, обеспечить надёжное хранение данных и более полный охват функций, подлежащих автоматизации.

2.2. Определение проблемы.

Проблема: недостаточно оперативный обмен информацией между диспетчером и всеми взаимодействующими с ними лицами.

Затрагивает: диспетчера, менеджеров, мастера цеха, работников цеха.

Её следствием является: задержка выполнения заказов.

Успешное решение: максимальная оптимизация работы цехов, экономия времени всех участников процесса.

Проблема: высокая трудоёмкость процесса диспетчеризации.

Затрагивает: диспетчера.

Её следствием является: затынутость процесса диспетчеризации, ошибки планирования.

Успешное решение: исключение ошибок, повышение эргономичности работы диспетчера.

Проблема: сложность назначения реального срока выполнения заказа, согласованного с производством.

Затрагивает: диспетчера, менеджера.

Её следствием является: возможная потеря заказа и, соответственно, выгодных клиентов, неисполнение заказов в срок.

Успешное решение: при внесении в систему запроса желательно получить варианты возможного времени исполнения данного заказа, исключить возможность накладок.

Проблема: трудность получения информации о текущем состоянии заказов в производстве.

Затрагивает: диспетчера, менеджеров, мастера цеха.

Её следствием является: ситуация, когда заказчик длительное время не имеет информации о реальном состоянии дел.

Успешное решение: оптимальное распределение времени выполнения заказов, получение текущей информации о ходе выполнения заказа.

2.3. Определение позиции изделия.

Для: типография BMB.

Которой: требуется оптимизировать процесс диспетчеризации заказов.

Название продукта: АИС "Диспетчер".

Который: основан на промышленной СУБД и высоконадёжен.

В отличие от: существующего механизма на основе электронных таблиц.

3. Описание пользователя.

3.1. Сведения о пользователях.

У системы существуют три основных пользователя: диспетчер, менеджер, мастер цеха.

Менеджер: вводит данные о вновь поступивших заказах, контролирует их исполнение.

Диспетчер: планирует работы цеха, следит за их исполнением.

Мастер цеха: вводит данные о реальном состоянии работ (заказов).

3.2. Пользовательская среда.

В настоящее время на предприятии имеется 5 производственных участков (цехов), 5 мастеров, 1 диспетчер и 10 менеджеров продаж. Увеличение количества мастеров в ближайших 8 лет максимально до 10-и, менеджеров - максимумом до 20-и, диспетчеров - максимумом до 2-х. Разумное время планирования одного заказа - 10 минут, для сложных заказов - не более 25-и минут. Платформа: . . . , операционная система: . . .

3.3. Профили пользователей.

Типичный представитель: менеджер.

Описание: пользователь системы, наделённый правами на чтение информации и занесение данных о заказах.

Тип: пользователь.

Ответственность: вводит данные о вновь поступивших заказах, назначает срок исполнения заказа на основе представленной в системе информации, осуществляет контроль заказов.

Критерий успеха: возможность определения реального срока исполнения заказов, получение данных о реальном состоянии заказа в производстве.

Типичный представитель: диспетчер.

Описание: пользователь системы, наделённый правами на изменение плановой информации в системе.

Тип: пользователь.

Ответственность: выполнение заказов в срок, ритмичная загрузка оборудования.

Критерий успеха: выполнение заказов в срок, минимизация простоя оборудования.

Типичный представитель: мастер цеха.

Описание: пользователь системы, наделённый правами на внесение фактической информации о выполнении заказов в производстве.

Тип: пользователь.

Ответственности: ввод данных о назначении исполнителей работ и результатах их деятельности.

Критерий успеха: наличие в системе оперативной информации.

3.4. Ключевые потребности пользователей.

Диспетчер: затрачивает большое количество времени на составление очереди заказов и внесение необходимых изменений.

Менеджеры: затрачивают много времени на урегулирование накладок с диспетчером.

Предприятие: нуждается в системе, которая позволит ускорить и оптимизировать указанные процессы.

4. Краткий обзор изделия.

4.1. Контекст использования системы.

Система является законченной независимой разработкой. В перспективе возможно использование системы в комплексе с системами автоматизации других подразделений. Коммуникации: на уровне доступа к общей базе данных.

4.2. Сводка возможностей.

Возможности заказчика|Поддерживаемые функции

Упрощение работы диспетчера|Автоматическое планирование управления срочными заказами, обратная связь с цехом, оперативная корректировка планов.

Ускорение обращения информации.|Система позволит ускорить процесс получения необходимой информации о времени выполнения заказов диспетчером и менеджерами, оптимизирует взаимодействие диспетчера и мастера цеха.

Формирование единой базы для планирования и последующего анализа|Все заинтересованные пользователи со своих рабочих мест имеют доступ к оперативной информации о загрузке производственных ресурсов и состоянии заказов. Накопленные в базе данные позволяют анализировать статистику.

Возможность индивидуального подхода к каждому заказу.|Система позволяет классифицировать заказы на простые, срочные и т. п. и поддерживать разные режимы планирования в зависимости от типа заказа. Отказ от излишних коммуникаций|Система позволит пользователям получать нужную им информацию самостоятельно, не отвлекая от работы других участников процесса.

4.3. Предложения и зависимости.

Система будет использоваться на территориально сосредоточенном предприятии; в случае изменений в формах документов система должна претерпеть малосущественные изменения. В случае приобретения или разработки информационных систем, автоматизирующих сложные участки (маркетинг, склад готовой продукции) будет необходимо разработать соответствующие средства импорта-экспорта информации.

5. Возможности продукта.

5.1. Структурированное описание заказа.

Возможность описания заказа через упорядоченную во времени совокупность работ, а также параметров.

5.2. Расчёт нормативного времени выполнения работ по заказу. Возможность для каждой из работ заказа автоматически определить на основании введенных параметров время исполнения работы с точностью до минуты.

5.3. Передача заказов в производство.

Возможность направить заказ, в котором указаны все необходимые параметры в производство.

5.4. Диспетчеризация работ заказа.

Возможность для каждой из работ заказа выбрать требуемый ресурс из возможных ресурсов и установить время начала и окончания выполнения с точностью до минуты.

5.5. Планирование работы цехов.

Возможность сверстать планы работы в разрезе цехов, суток, смен, оборудования.

5.6. Назначение исполнителей.

Возможность для каждой из цеховых работ назначать исполнителей.

5.7. Контроль исполнения и оперативная корректировка планов

Возможность контроля исполнения работ над заказами и цеховых планов.

Возможность оперативной корректировки планов при возникновении критических ситуаций.

6. Ограничения.

Внедрение системы не должно занимать более 3 месяцев. В ядре системы должна быть заложена промышленная СУБД relationalного типа. Все обращения к информации должны осуществляться через драйвер ODBC.

7. Показатели качества.

7.1. Применимость.

7.1.1. Время, необходимое для обучения обычных пользователей (три рабочих дня (24ч); для обучения продвинутых пользователей - 1 рабочий день).

7.1.2. Время отклика для типичных задач - не более 5с, для сложных задач - не более 20с.

7.2. Надёжность.

7.2.1. Доступность (время, затрачиваемое на обслуживание системы не должно превышать 3% от общего времени работы).

7.2.2. Среднее время безотказной работы - 10 рабочих дней.

7.2.3. Максимальная норма ошибок или дефектов - одна ошибка на 10 000 строк кода.

8. Другие требования к продукту.

8.1. Применяемые стандарты.

Система должна соответствовать всем стандартам интерфейса пользователя

Microsoft.

8.2. Системные требования.

Процессор: не менее 2 ГГц (4 ядра); оперативная память: не менее 4 Гб; жёсткий диск: 2-4 диска (резервные); сетевой адаптер: 100 Мб/с; операционная система: Microsoft Windows Server 2003; СУБД: SQL Server.

8.3. Эксплуатационные требования.

Система должна быть способна поддерживать как минимум 15 одновременно работающих пользователей, связанных с общей базой данных, и иметь возможность увеличить их количество на случай увеличения штата сотрудников предприятия (см. пункт 3.2).

9. Требования к документации.

9.1. Руководство пользователя.

В системе должны быть представлены руководства пользователей (по типам пользователей). Они должны содержать расшифровку всех используемых терминов, описание основных вариантов использования, включая альтернативные сценарии, а также подробный обзор интерфейса программы.

9.2. Интерактивная справка.

Интерактивная справка необходима для разрешения возникших во время работы вопросов. В справке должна быть реализована возможность поиска информации по ключевым словам, а также вариант представления информации по отдельным позициям меню программы. Справка должна содержать максимально полную и подробную информацию о работе системы.

9.3. Руководство по установке и конфигурированию. Файл ReadMe.

Система должна иметь руководство по установке в файле README.txt, который должен прилагаться к системе. Файл ReadMe должен содержать подробную инструкцию по установке данной системы, чтобы в случае необходимости пользователь смог произвести установку самостоятельно, без помощи администратора.

10. Маркировка и пакетирование.

Система будет распространяться на компакт-диске, на котором будет находиться сама система, а также интерактивная справка, руководство по установке и руководство пользователей. Инсталляционная программа должна включать общее лицензионное соглашение и информацию об авторских правах. Словарь терминов (глоссарий)

В простейшем варианте словарь терминов (глоссарий) представляет собой список важных понятий и их определение. Часто наблюдается ситуация, когда технический или другой термин используется заинтересованными лицами в нескольких возможных значениях. Такие противоречия следует устранить для облегчения общения и формулировки требований. В рамках унифицированного процесса словарь терминов также играет роль словаря данных, то есть документа, содержащего информацию о данных. На начальной стадии проекта словарь терминов включает только основные термины и их описание, а на стадии развития желательно превратить его в словарь данных. В этом случае к атрибутам терминов следует отнести: список синонимов, описание термина, формат термина (тип, длина, единица измерения), взаимосвязи с другими элементами, диапазон значений, правила проверки корректности. В словарь терминов записываются не только простейшие понятия, такие как "цена товара", но и более сложные элементы, например, "продажа", которые, в свою очередь, могут включать в себя другие элементы, такие как "дата", "место", "дата авторизации платежа" и т. д.

Документ Delta Vision

Документ-концепция требует сопровождения. По мере работы над программным продуктом появляются новые функции, новый взгляд на старые функции и т. д. Не рекомендуется каждый раз создавать новую версию документа концепции и переписывать в неё всё то, что есть в старой версии. Предлагается ввести документ изменений, в котором основное внимание уделено тому, что новое включается в данную версию и чем она отличается от предыдущей.

Управление масштабом

Проблема масштаба проекта

Приступая к разработке приложения, необходимо произвести оценку ресурсов, выделенного времени и поставленных целей. При разработке ПО эти факторы задают масштаб проекта, который определяется следующими переменными: 1) набором функций программного продукта; 2) ресурсами, которыми располагает проект; 3) временем, выделенным на реализацию.



Площадь прямоугольника представляет достижимый масштаб проекта.

Ресурсы в основном состоят из труда разработчиков, тестологов, составителей руководств, сотрудников отдела управления качеством и т. д.

Добавление ресурсов в отстающий по времени проект часто приносит отрицательный результат. Объясняется это тем, что обученные ранее сотрудники отвлекаются на консультации для новых работников. Время иногда имеет мягкое ограничение, однако существуют проекты, где оно задано очень строго. Если время и ресурсы определены, то управление масштабом связано только с функциями.

Задание масштаба проекта

Рекомендуется задать базовый уровень требований. Базовый уровень - это разбитое на элементы множество функций и требований, которые намечено реализовать в данной версии приложения. Он должен быть по крайней мере приемлем для заказчика и иметь приемлемую вероятность успеха с т. з. команды разработчиков. Первым шагом в определении базового уровня является перечисление функций, которые были определены для приложения. Разрабатывается некоторый программный продукт. Допустим, были выделены следующие функции:

№ функции	Функция	Приоритет	Трудоёмкость	Риск
Ф1	Поддержка внешней реляционной БД	Критический	средняя	
Ф2	Многопользовательская безопасность	Важный	высокая	
Ф3	Возможность клонирования проекта	Важный	высокая	
Ф4	Порт для новой версии операц. системы	Критический	высокая	
Ф5	Новый "Мастер" проекта	Важный	низкая	
Ф6	Импортирование внешних данных по типам	Критический	низкая	
Ф7	Реализация средств предупреждения	Полезный	низкая	
Ф8	Интеграция с системой управления версиями	Полезный	низкая	

1) Установка приоритетов

Первичное определение приоритетов должно производиться заказчиками и пользователями. Технические моменты учитываются позже. Рекомендуется использовать шкалу "критический", "важный", "полезный".

Оценка трудозатрат

Анализ трудоёмкости

Для анализа трудоёмкости используются в основном два метода: 1) метод анализа трудоёмкости проекта (задачи) на основе трудоёмкости известного образца; для применения этого метода в качестве значения трудоёмкости основной работы выбирают данные, характеризующие трудоёмкость изделия-аналога, относительно которого вводят коэффициент сложности новой разработки или его части. сложность программы-аналога или её части принимается за единицу, затем определяют коэффициент квалификации работника (программиста) nv(сл.), который отражает степень его подготовленности к выполнению порученной ему работы; коэффициент квалификации исполнителя nv(кв.) рекомендуется определять в зависимости от его стажа работы: до 2-х лет: 0,8, от 2 до 3 лет: 1, от 3 до 5 лет - 1.1-1.2, от 5 до 7 лет - 1.3-1.4, более 7 лет - 1.5-1.7;

q^(new)vi=(q^(o1d)vj\*nv(сл.))/nv(кв.); 2) анализ трудоёмкости на основе экспертных оценок: суть метода заключается в том, что опрашивается несколько экспертов с целью определения длительности работы (этапа работы проекта). Полученные результаты заносятся в таблицу перечня работ:

Работа	Экспертная оценка			
	1	2	...	k
rv1	tv(11)	tv(12)	...	tv(1n)
rv2				
.				
.				
rvn				

Добавление элементов риска

Далее необходимо оценить риск, связанный с каждой функцией. Под риском понимаем вероятность того, что разработка функций может оказать негативное воздействие на график и бюджет. Здесь обычно используют классификацию "низкий", "средний", "высокий".

Сокращение масштаба

Если функция критическая и имеет высокий риск, то нужна эффективная стратегия уменьшения риска. Если функция с высоким риском только полезна, следует рассмотреть вариант её удаления. Если функция важная, но имеет высокий риск, то её можно удалить или разрабатывать в случае, если останется время. Если команда чувствует, что масштаб продукта превышает 100%, следует сокращать список функций. Если масштаб достигает 200%, то следует в базовый уровень включать только критические функции.

	Приоритет	Трудоёмкость
Ф1	Поддержка вложений PER	критич.
Ф4	Порт для новой версии ОС	критич.
Ф6	Импортирование внешних ...	критич.
Ф3	Возможность клонирования проекта	важный
Базовый уровень (отсутс. Важность)		
Ф2	Многопользоват. безопас.	Важный
Ф5	Новый мастер проекта	Важный
Ф7	Реализация средств предупреждения	Полезная
Ф8	Интеграция с подсист. управ. верс.	Полезная

Умение общаться с заказчиком  
Сокращение масштаба проекта до размеров, сопоставимых с имеющимся временем и ресурсами, может привести к враждебным отношениям между командой проекта и его заказчиками. Для того, чтобы избежать подобной ситуации, следует активно привлекать заказчиков к управлению их требованиями и масштабам проекта. Нужно дать понять заказчику, что сокращение функций позволит создать качественный продукт вовремя. Руководящий принцип при управлении масштабом - меньше обещать и больше делать.

Лидер продукта

В каждом успешном проекте должен быть лидер. Лидер ставит концепцию продукта во главу угла и постоянно о ней думает. Лидером может быть: менеджер продукта, менеджер информационных технологий, менеджер проектирования, руководитель проекта. В любом случае он должен заниматься следующим: 1) руководить процессом выявления требований; 2) рассматривать конфликтующие пожелания, поступающие от различных участников; 3) находить компромиссы, необходимые для определения набора функций, предоставляющего наибольшую ценность для максимального числа участников; 4) владеть концепцией продукта; 5) защищать интересы продукта; 6) вести переговоры с руководством, пользователями и разработчиками; 7) противодействовать "просачиванию" функций; 8) поддерживать здоровое равновесие между тем, что хочет заказчик, и тем, что может сделать команда разработчиков за время, отведенное на реализацию проекта; 9) вызывать в роли представителя, официального канала общения между заказчиком и командой разработчиков; 10) управлять ожиданиями клиентов, дирекции, отдела маркетинга и проектирования; 11) сообщать о реализуемых функциях всем заинтересованным лицам; 12) осуществлять проверку спецификаций ПО, чтобы удостовериться, что они соответствуют функциям концепций; 13) осуществлять управление изменением приоритетов: добавление и удаление функций.