

---

---

# Анализ и управление требованиями

— Часть 4 —  
Документирование. Контроль.

---

---

# Спецификация требований

особый итоговый набор  
расположенных по  
приоритетам требований,  
который является  
формальным  
соглашением заказчика с  
разработчиком системы.

---

# Спецификация требований

требования:

- должна быть простой, ясной и понятной пользователю неспециалисту
- должна быть точной, подробной, формальной и понятной для разработчика

# Спецификация требований

Пользовательские  
требования

Системные  
требования

# Спецификация требований (основные разделы)

1. Введение
2. Общее описание
3. Функции системы
4. Внешний интерфейс
5. Нефункциональные требования

# Введение

**обзор** спецификации, позволяющий разобраться в структуре и принципах ее построения.

**определяется** специфицируемая далее **система** или подсистема, указывается ее редакция и версия;

перечисляются **стандарты**, используемые при написании спецификации, а также все документы, на которые есть ссылки в спецификации.

краткое описание **концепции системы**, перечисляют различные группы пользователей спецификации и приводится рекомендуемая последовательность чтения разделов для них.

# Общее описание

*Продукт* - особенности продукта или его основные функции, основные группы требований и их взаимосвязь ( диаграмма потоков данных высокого уровня, диаграмма варианта использования и т.п.)

*Пользователи.* перечисление классов предполагаемых пользователей и их характеристики.

*Окружение* - рабочая среда ПС: аппаратные средства, операционные системы, другие программные компоненты, с которыми система должна быть совместима.

*Ограничения реализации.* технологии, языки программирования, базы данных, стандарты разработки, которые следует (или не следует) использовать; ограничения операционной среды, аппаратуры; ограничения, связанные с продуктами, разработанными ранее; соглашения, связанные с пользовательским интерфейсом, форматом обмена данными и т.п.

# Функции системы

- название и приоритет;
- системные входные и выходные данные, или последовательности «воздействие – реакция»;
- детальные функциональные требования;
- реакция на ошибки ввода данных или неверные действия пользователя.

# Внешний интерфейс

- ссылки на стандарты графического интерфейса, шрифтов, элементов управления и т.п.;
- конфигурация экрана, стандартные кнопки и функции навигации;
- стандарты отображения сообщений и т.д.
- характеристики интерфейсов программных и аппаратных компонентов системы, протоколы их взаимодействия.
- интерфейсы между системой и другими программными компонентами и интерфейсы передачи информации.

# Нефункциональные требования

требования к производительности,

атрибуты качества продукта,

требования к безопасности, охране труда и т.д.

словарь терминов (глоссарий),

список нерешенных проблем, связанных с требованиями и т.д

# Рекомендации по документированию требований

# Пользовательские требования

1. Если требование независимое и простое, то оно может быть записано в виде нескольких простых предложений.
2. Если требование представляет собой взаимодействие (пользователя и системы), то его можно описать с помощью варианта использования.
3. Если требование определяет обработку данных, т.е. получение входных данных их обработку и генерацию выходных данных, то можно использовать диаграммы потоков данных.
4. Если требование определяет состояния системы, то для его фиксации можно применить конечный автомат.

# Системные требования

1. Перед написанием спецификации системных требований необходимо выбрать стиль описания.
2. Для каждого требования:
  - определить требование как функциональное или нефункциональное;
  - оценить сложность функционального требования (требование большое – трудно управлять, очень маленькое – нет смысла рассматривать отдельно);
  - сделать требование прослеживаемым и проверяемым (написать проверочный тест),
  - доопределить требование для случая нештатных ситуаций;
  - проверить недвусмысленность требования, назначить ему приоритет,
  - проверить полноту и согласованность требования.

1. Требование должно быть **понятным**
2. Требование должно быть **конкретным**
3. Требование должно быть **тестируемым**

# Стандарты на документирование требований

# IEEE 830-1993 (спецификация ФТ)

## 1. Введение

- a. цели документа
- b. назначение программного продукта
- c. определения, термины, сокращения
- d. список литературы
- e. обзор спецификации

## 2. Общее описание

- a. описание ПП
- b. функции ПП

c. пользовательские характеристики

# Способы структурирования требований

1. *По основным свойствам.* Предоставляемые программой сервисы определяются с помощью пар «стимул – реакция».
2. *По режиму работы,* например, демонстрационный, нормальный или аварийный режимы;
3. *По вариантам использования,* или сценариям. Особенность такой организации требований в том, что наиболее подробные требования являются частью варианта использования;
4. *По классам.* В этом объектно-ориентированном способе требования классифицируются по классам.

5. *По иерархии функций.* Это традиционный способ упорядочивания детальных требований. Программа разбивается на множество высокоуровневых функций, каждая из которых, в свою очередь, представляется в виде подфункций, и т. д.

6. *По состояниям,* например, состояния ожидания, выполнения и завершения. Внутри классификации каждого состояния перечисляются события, влияющие на программу, находящуюся в конкретном состоянии. Классификация по состояниям может быть уместной, если требования для каждого состояния сильно отличаются.

# ГОСТ 34.602-89 (Техническое задание)

1. Общие сведения
2. назначение и цели создания (развития) системы;
3. характеристика объектов автоматизации;
4. требования к системе;
5. состав и содержание работ по созданию системы;
6. порядок контроля и приемки системы;
7. требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;

# Шаблон SRS, предложенный в RUP

## 1. Введение.

1.1. Цель.

1.2. Краткая сводка возможностей.

1.3. Определения, акронимы и сокращения.

1.4. Ссылки.

1.5. Краткое содержание.

## 2. Обзор системы

2.1. Обзор прецедентов. Содержит список имен и кратких описаний вариантов использования и акторов с иллюстрациями в виде диаграмм прецедентов.

2.2. Предположения и зависимости. Данная секция описывает ключевые технические возможности, компоненты, подсистемы, связанные проекты, которые могут влиять на жизнеспособность разрабатываемой системы. Предположением (assumption) называется положение, которое считается истинным при отсутствии доказательства или определяющей информации. [1].

При определении зависимостей (dependencies) проекта от внешних факторов, необходимо проанализировать, какие новые операционные системы, регламенты бизнес-процессов, стандарты качества, информационные системы могут появиться на предприятии внедрения и как это может повлиять на функционирование изготавливаемой АИС.

### 3. Описание требований

3.1. Описание вариантов использования. Параграф содержит описание вариантов использования и связанных с ними нефункциональных требований, либо ссылки на соответствующие артефакты.

3.2. Специальные требования. Параграф содержит описание функциональных требований (не описанных в качестве вариантов использования), а также описание нефункциональных требований общего характера (не сопоставленных ни одному прецеденту в предыдущем разделе), либо ссылки на соответствующие артефакты.

4. Вспомогательная информация. Сюда включается информация, облегчающая понимание документа. Это может быть оглавление и приложения, например, описывающие прототипы пользовательского интерфейса.

# MSF (Microsoft Solution Framework)

## 1. Бизнес-преимущества

- a. описание преимуществ
- b. формулировка видения
- c. анализ выгод

## 2. Концепция решения

- a. цели, задачи, предложения и ограничения
- b. анализ применимости
- c. требования

# MSF (Microsoft Solution Framework)

## 3. Рамки

- a. список характеристик/функций
- b. вне рамок
- c. стратегия подготовки релизов
- d. критерии применимости
- e. эксплуатационные критерии

## 4. Стратегии проектирования решения

- f. стратегии проектирования архитектуры

# Оценка качества спецификации требований

# Характеристики качества

- полнота, согласованность,
- способность к модификации
- трассируемость.

# Аттестация требований

экспертиза спецификации,

неофициальная (во время разработки)

официальная (по окончании разработки)

прототипирование,

автоматизированный анализ

тестирование

# Проблемы

1. большой объем документации
2. большая команда экспертов

# Управление

# Принципы управления требованиями

1. Определение основной (базовой) версии спецификации требований для конкретной версии продукта;
2. Анализ предлагаемых изменений требований, оценка воздействия и стоимости каждого изменения до его принятия;
3. Включение одобренных изменений при помощи определенной процедуры;
4. Согласование плана проекта с требованиями;
5. Отслеживание отдельных требований от проектирования до кода приложения и его тестирования;
6. Отслеживание статуса требований и действий по изменению на протяжении всего проекта.

# Процесс управления требованиями

определить:

- методы и средства управления версиями спецификации и отдельных требований;
- процесс разработки, согласования, экспертизы и утверждения базовой версии;
- процесс присвоения, контроля и изменения статуса требования;
- процесс передачи новых требований и изменений существующих требований заинтересованным лицам;
- методы анализа влияния и стоимости внесения изменения.

участников проекта, ответственных за выполнение каждой конкретной задачи.

# Статус требования

В процессе выполнения проекта требование, обычно, изменяет свое состояние от начального («предложено»), до конечного, например, «реализовано».

Статус требований позволяет оценить степень готовности проекта.

Состояние	Определение
Предложено (proposed)	Требование запрошено авторизованным источником
Одобрено (approved)	Требование проанализировано, его влияние на проект просчитано, и оно было размещено в базовой версии определенной версии продукта. Ключевые заинтересованные в проекте лица согласились с этим требованием, а разработчики обязались его реализовать
Реализовано (implemented)	Код, реализующий требование разработан, написан и протестирован. Требование отслежено до соответствующих элементов дизайна и кода

Проверено (verified)	Корректное функционирование реализованного требования подтверждено в соответствующем продукте. Требование отслежено до соответствующих вариантов тестирования. Теперь требование считается завершенным
Удалено (deleted)	Утвержденное требование удалено из базовой версии. Зафиксируйте причины и лицо, принявшее это решение
Отклонено (rejected)	Требование предложено, но не запланировано для реализации ни в одной из будущих версий. Зафиксируйте причины и лицо, принявшее это решение

# Управление изменениями

Описание процесса контроля изменений должно содержать:

1. Границы применения процесса. Должны быть перечислены те продукты или изменения, которые не принадлежат процессу.
2. Роли и ответственность. Должны быть определены члены проекта (роли), участвующие в контроле изменений и их ответственность.
3. Описание процедуры прохождения запроса на изменение.
4. Описание процедур анализа и оценки запроса на осуществимость, влияния и стоимости изменения, принятия решения и изменения состояния запроса.

# Управление связями требований

**Трассируемость требований** позволяет решить следующие задачи.

- 1.Продемонстрировать, что все требования проекта реализованы.
- 2.Снизить вероятность того, что при внесении изменений будет упущено требование, на которое оказывает влияние изменение.
- 3.Упростить внесение изменения на поздних фазах проектирования, а также фазе эксплуатации и сопровождения продукта.
- 4.Зафиксировать опыт проектирования, упростить повторное проектирование и повторное использование.
- 5.Упростить поиск, локализацию и исправление ошибок при тестировании.

<b>Пользовательское требование</b>	<b>Функциональное требование</b>	<b>Элемент дизайна</b>	<b>Модуль кода</b>	<b>Вариант тестирования</b>

Матрица трассирования

# Уровни зрелости процесса управления требованиями

В зрелой организации:

- имеются четко определенные процедуры создания программного продукта и управления программными проектами;
- оценки времени и стоимости выполнения работ основываются на накопленном опыте;

существуют стандарты на процессы разработки, кодирования, тестирования, внедрения и сопровождения продукта

**Модель зрелости способностей предприятий в области  
разработки программного обеспечения  
(Capability Maturity Model for Software – CMM)  
определяет уровни зрелости, характеризующей  
способности коллектива к разработке программного  
продукта**

В модели СММ различают пять уровней зрелости (наивысшим из которых является пятый) и насчитывается 24 ключевые области процесса, которые распределены по уровням зрелости.

*Ключевые области процесса* – это основные компоненты способностей предприятия по разработке программ.

**Capability Maturity Model** – система качества, разработанная SEI (Software Engineering Institute)

**Цель СММ:** гарантирование реализации проекта разработки ПО в заданный срок с заданным бюджетом и высоким качеством.

**СММ** - не технология, не стандарт, для нее нет никаких формальных описаний, и SEI не рекомендует использовать ориентированные на СММ CASE-системы

**СММ предназначена** для организации эффективного управления разработкой ПО. Она определяет ключевые действия, которые указывают, что надо сделать для достижения требуемого качества ПО (но не указывают, как).

## Уровни зрелости процесса управления требованиями

### **Уровень 0. Начальный (Initial) Отсутствие требований**

На начальном уровне процесс разработки программного продукта либо неустойчивый и хаотичный, либо про него ничего не известно. (На этот уровень попадают все предприятия-разработчики, поскольку на нем отсутствуют какие-либо требования к процессу.)

### **Уровень 1 Управляемый (Managed) Документирование требований**

На втором уровне процесс разработки программного продукта становится управляемым. Предприятие, находящееся на втором уровне зрелости, добилось того, что все процессы планируются, документируются, выполняются, оцениваются и контролируются на уровне программных проектов.

# Ключевые области

На втором уровне определены:

1. Управление требованиями (Requirements Management) – обеспечение возможности установления единого с заказчиком понимания требований к разрабатываемому продукту.
2. Управление конфигурацией (Configuration Management) – установление и поддержка целостности всех продуктов проекта или процесса за счет обеспечения единой системы идентификации компонентов продукта, контроля над изменениями и управления изменениями компонентов.

### **Уровень 3. Определенный (Defined) Организация требований**

Для третьего уровня зрелости процесса характерно, что все виды деятельности, связанные с процессом, основаны на одном, общем для всего предприятия, стандартном наборе процессов, которые подогнаны под конкретные условия, хорошо понимаемы и описаны в корпоративных стандартах, руководствах и т.п.

Разработанный на предприятии набор стандартных процессов постоянно усовершенствуется и является базисом для развития и внедрения согласованных процессов во всех проектах, которые ведутся на предприятии.

# Ключевые области

Третий уровень характеризуется включением ключевой области:

Разработка требований (Requirements Development)» – разработка и анализ требований заказчика, требований к программному продукту и отдельным его компонентам. Идентификация требований, доступность, целостность, версионность.

## **Уровень 3. Структурирование требований**

на третьем уровне зрелости выполняются активности по планированию процесса управления требованиями и группировки выявленных требований по типам в соответствии с объединяющими признаками, что облегчает управление и приводит к лучшему пониманию требований всеми участникам проекта.

# Ключевые области

## Использование CASE - СРЕДСТВ -

На данном уровне зрелости может возникнуть необходимость применения специализированных инструментальных средств, предназначенных для работы с требованиями.

## **Уровень 4. Трассировка требований**

Достижения предыдущих трех уровней зрелости приведет проектную команду к тому, что можно будет определять отношения между требованиями. Установка отношений (*трассировка*) позволяет отслеживать влияние требований друг на друга (*анализ влияния*) и определять избыточные и неучтенные требования (*анализ покрытия*).

# Ключевые области

Анализ влияния заключается в прослеживании воздействия изменений одного требования на изменение других требований.

Анализ покрытия позволяет после подготовки набора требований однозначно определить, что требования верхнего уровня (функция) полностью детализированы и описаны в требованиях более низкого уровня (программное требование).

## **Уровень 5. Интеграция требований**

Для создания программного обеспечения, соответствующее требованиям заказчика необходимо, чтобы команда проекта использовала требования как основную входную информацию для процесса разработки.

Целью пятого уровня зрелости как раз и является интеграция процесса управления требованиями в процессы управления изменениями, проектирования, разработки, тестирования и управления проектом в целом. Безусловно, такая плотная интеграция нужна в крупных проектах и требует существенных затрат на закупку специализированных инструментальных средств, применяемых в разработке программного обеспечения.

# Типовые инструменты

- моделирование требований;
- трассировка требований;
- управление версиями.

**CMM**

## **Первый уровень.**

ПРПО фирмы никак не организованы, и реальные попытки их улучшения не предпринимаются.

## **Второй уровень.**

Компания на основе анализа успешных проектов начинает повторно использовать подходящие ПРПО.

Положительная практика документируется, организуется учеба сотрудников и определяются пути улучшения ПРПО для их применения в смежных проектах.

Создаются внутрифирменные стандарты ПРПО, внедряются процессы планирования и контроля за работой.

Налаживается обратная связь с заказчиками.

Определяются примерные ресурсы на используемые ПРПО. Фирме уже удается укладываться в заданные сроки и бюджет (с определенной вероятностью отклонения). Проект разбивается на небольшие части и становится более понятным менеджерам и разработчикам

Однако, успех проектов по-прежнему зависит от ведущих специалистов, а ПРПО разработаны только для конкретных областей (бухгалтерское или математическое ПО и т. п.).

Обычно на этом уровне компания создает информационные и функциональные модели

# Третий уровень

. Компания детально стандартизует используемые ПРПО (пока для решения достаточно общих задач) с учетом их дальнейшего развития.

Все процессы интегрированы в жестко регламентированный общефирменный процесс разработки ПО.

Большинство ПРПО удается успешно адаптировать для работы над проектами из разных областей. Каждый раз по завершении проекта ПРПО улучшаются, причины улучшений документируются, и новые проекты реализуются на основе более зрелых процессов.

Широко внедряются всевозможные учебные программы.

Роль отдельных личностей перестает влиять на результат.

Этот уровень характеризуется тем, что в фирме должна быть создана специальная группа софт-инжиниринга (software engineering process group).

Каждый сотрудник - от кодировщика до руководителя - точно знает, что он должен сделать.

При возникновении у заказчика новых требований удастся оценить риск изменения проекта.

В сравнении со вторым уровнем проекты реализуются быстрее и с меньшими затратами. Снижается вероятность отклонения от срока и уменьшается величина этого отклонения. Фирма подготовлена к любым непредвиденным проблемам и способна их решить.

Заказчик в любой момент может получить детальную информацию о текущем состоянии проекта.

# Четвертый уровень

Все ПРПО компании могут быть использованы для работы над программными проектами разной тематики.

Процессы оценены по множеству критериев, максимально документированы и легко управляемы.

На первый план выходит эффективное управление ПРПО, благодаря чему повышается качество продуктов и продолжают снижаться требования к ресурсам.

Для областей, в которых компания уже работала, удается точно уложиться в сроки и бюджет, для новых областей детально оговаривается небольшая зона риска. ПО разрабатывается с заданным качеством. Возникающие проблемы оказывают минимальное воздействие на проект.

Фирма создает базу данных по используемым ПРПО и постоянно ее анализирует. От подобного формализованного опыта существенно зависит снижение сроков и затрат на проект.

Менеджеры не только в деталях понимают структуру проекта, но и сами начинают управлять ПРПО.

# Пятый уровень

Компания осуществляет непрерывную и неограниченную оптимизацию своих ПРПО.

Для каждого процесса определены сильные и слабые стороны и наиболее подходящие области применения.

При работе над проектом менеджеры постоянно улучшают используемые процессы, причем степень улучшения поддается количественной оценке.

В ПРПО вносятся новые идеи и технологии, анализируются и исправляются ошибки.

Менеджеры не просто понимают процессы, но и осознают возможные пути повышения их эффективности.