

# Методика составления спецификаций требований к программному обеспечению, рекомендуемая Институтом Инженеров по Электротехнике и Радиоэлектронике (IEEE)

Организатор

Комитет по Стандартам Разработок Программного Обеспечения  
Компьютерного Общества IEEE

Утверждено 25 июня 1998 Совет

по Стандартам IEEE-SA

**Выдержка:** Описывается содержание и качественные характеристики правильно составленной спецификации требований к программному обеспечению (SRS) и приводятся несколько образцовых SRS. Данная рекомендуемая методика имеет своей целью установление требований к разрабатываемому программному обеспечению, но также может применяться, чтобы помочь в выборе собственных и коммерческих программных изделий. Также приведены указания по согласованию со стандартом IEEE/EIA 12207.1-1997.

**Ключевые слова:** контракт, заказчик, макетирование, спецификация требований к программному обеспечению, поставщик, спецификации требований к системе.

Институт Инженеров по Электротехнике и Радиоэлектронике, Инк.,  
345 East 47<sup>th</sup>, New York, NY 10017-2394, USA

Авторское право © 1998 г. Института Инженеров по Электротехнике и Радиоэлектронике, Инк.  
Все права сохранены. Опубликовано в 1998 г. Отпечатано в Соединенных Штатах Америки

ISBN 0-7381-0332-2

Никакая часть данной публикации не может быть воспроизведена в любом виде на любой электронной информационно-поисковой системе или другим способом без предварительного письменного разрешения издателя

Документы по **Стандартам IEEE** были разработаны Обществами IEEE и Координирующими Комитетами Совета по Стандартам Ассоциации Стандартов IEEE. Члены комитетов выполняют свою работу добровольно и без материальной компенсации. Они не обязательно являются членами Института. Стандарты, разработанные IEEE, представляют согласованное мнение по широкомасштабной экспертизе, выполненной по данной теме в пределах Института, а также деятельность других организаций за пределами IEEE, которые выразили интерес к участию в разработке этого стандарта.

Использование Стандарта IEEE является полностью добровольным. Существование Стандарта IEEE не подразумевает, что отсутствуют какие-либо другие способы создания, тестирования, измерений, приобретения, коммерческой продажи или поставки других товаров и услуг, относящихся к области действия Стандарта IEEE. Кроме того, точка зрения, выраженная при утверждении и издании стандарта, подлежит изменениям, вызванным развитием современного уровня техники и комментариями, полученными от пользователей стандарта. Каждый Стандарт IEEE подлежит проверке, по крайней мере, один раз в пять лет с целью пересмотра или повторного подтверждения. Если срок действия документа составил более пяти лет, и он не прошел повторного подтверждения, то можно обоснованно заключить, что его содержание, хотя все еще и имеет некоторую ценность, не отражает полностью текущий уровень развития техники. Пользователям рекомендуется убедиться, что они имеют последнее издание любого из Стандартов IEEE.

Приветствуются любые комментарии для пересмотра Стандартов IEEE от любой заинтересованной стороны, независимо от наличия или отсутствия членства в IEEE. Предложения по изменениям в документах должны быть оформлены в виде предложенного изменения текста, вместе с соответствующими пояснительными комментариями.

Интерпретация: Иногда могут возникнуть вопросы относительно смысла отдельных частей стандартов, поскольку они относятся к специфическим приложениям. Когда внимание IEEE будет привлечено к необходимости в пояснениях, Институт примет меры по подготовке соответствующих ответов. Так как Стандарты IEEE представляют согласованное мнение всех заинтересованных сторон, важно обеспечить, чтобы любая интерпретация также была согласована относительно этих заинтересованных сторон. По этой причине IEEE и члены его технических комитетов не способны обеспечить мгновенный ответ на запросы, касающиеся интерпретации, за исключением тех случаев, когда вопрос был предварительно рассмотрен в соответствии с формальной процедурой.

Комментарии по стандартам и запросы, касающиеся интерпретации, следует отправлять по адресу:

Secretary, IEEE-SA Standards Board (Секретарю Совета по Стандартам IEEE-SA)  
445 Hoes Lane  
P.O. Box 1331  
Piscataway, NJ 08855-1331  
USA

Примечание: Необходимо обратить внимание на тот факт, что при реализации этого стандарта может потребоваться использование некоторых изделий, защищенных патентными правами. При публикации этого стандарта не занимается никакая позиция относительно существования или достоверности любых патентных прав по используемым изделиям. IEEE не будет нести ответственность за идентификацию патентов, для использования которых каким-либо стандартом IEEE может потребоваться разрешение, или за проведение запросов по юридической обоснованности или области действия этих патентов.

Разрешение на фотокопирование частей любого отдельного стандарта для внутреннего или персонального использования предоставляется Институту Инженеров по Электротехнике и Радиоэлектронике, Инк. при условии уплаты соответствующего взноса в Центр Расчеты по Авторским Правам. По вопросам уплаты лицензионного гонорара, пожалуйста обращайтесь в Центр Расчеты по Авторским Правам, Отдел обслуживания заказчиков, 222 Rosewood Drive, Danvers, MA 01923 США; (978) 750-8400. Разрешение на фотокопирование частей любого отдельного стандарта для использования в образовательных классах может также быть получено через Центр Расчеты по Авторским Правам.

## Введение

(Это введение не является частью стандарта IEEE 830-1998, Методика составления спецификаций требований к программному обеспечению, рекомендуемая IEEE.)

Данная методика описывает рекомендуемые принципы составления спецификации требований к программному обеспечению. Она основана на модели, в которой результат процесса спецификации программного обеспечения является однозначным и полным спецификационным документом. Она должна помочь

- а) Заказчикам программного обеспечения точно описать, что они хотят получить;
- б) Поставщикам программного обеспечения точно понять, что хочет заказчик;
- в) Отдельным лицам выполнить следующие задачи:
  - 1) Разработать схему стандартной спецификации программного обеспечения (SRS) для их собственных организаций;
  - 2) Определить формат и содержание конкретных спецификаций требований к программному обеспечению;
  - 3) Разработать дополнительные вспомогательные документы, такие как контрольный лист для проверки качества SRS или справочник составителя SRS..

Качественно составленная SRS должна принести заказчикам, поставщикам и другим лицам некоторые определенные выгоды, а именно:

- *Создать основу для соглашения между заказчиками и поставщиками по вопросу о том, какие функции должно выполнять программное изделие.* Полное описание функций программного обеспечения, приведенное в SRS, поможет потенциальным пользователям определить, отвечает ли программное обеспечение их потребностям или как необходимо изменить программное обеспечение, чтобы удовлетворить эти потребности.
- *Уменьшить объем работ по разработке.* Подготовка SRS вынуждает различные участвующие группы организации заказчика строго рассмотреть все требования прежде, чем приступить к выполнению проекта, и сокращает последующие повторные проектирование, кодирование и тестирование. Тщательный анализ требований, указанных в SRS, может вскрыть упущения, неправильное понимание и противоречия, допущенные на стадии разработки, когда их проще исправить.
- *Обеспечить основу для оценки расходов и планов.* Описание программы, разрабатываемой в соответствии с SRS, является практической основой для оценки затрат на проект и может использоваться для утверждения проекта на основании этих оценок.

*Обеспечить основу для проверки правильности и верификации.* Организации могут составлять планы проверки правильности и верификации намного более эффективно при использовании качественно разработанной SRS. Как часть контракта на разработку, SRS обеспечивает основу для определения соответствия.

*Облегчить передачу.* SRS делает более простой передачу программного изделия новым пользователям или его установку на новых машинах. Таким образом, заказчики могут более просто передавать программное обеспечение другим подразделениям их организации, а для поставщиков будет проще передавать его новым заказчикам.

- *Служить в качестве основы для расширения.* Поскольку в SRS обсуждается изделие, а не проект, которому оно разработано, то SRS служит основой для последующего расширения готового изделия. SRS может потребовать некоторых изменений, но это обеспечивает основу для его непрерывной оценки с производственной точки зрения.

В Приложении Б читатели этого документа найдут руководящие указания по использованию данной рекомендуемой методики в отношении соответствия требованиям стандартов IEEE/EIA 12207.1-1997, Руководство IEEE/EIA - Промышленная реализация ISO/IEC 12207: 1995, Стандарт Информационных Технологий - Процессы жизненного цикла программного обеспечения - Данные жизненного цикла.

## Участники

Эта рекомендуемая методика была подготовлена Рабочей Группой по Гармонизации Данных Жизненного Цикла, входящей в состав Комитета по Разработке Стандартов Программного Обеспечения Компьютерного Общества ШЕЕ. На момент утверждения данной методики рабочая группа состояла из следующих членов:

Леонард Л. Трипп, *Председатель*

Эдвард Бирн  
Пол Р. Кролл  
Перри Де Вис  
Робин Фрэлик  
Мэрилин Гинсберг-Финнер  
Джон Харас  
Марк Хэнли

Деннис Лоуренс  
Дэвид Мэйбор  
Рэй Милованович  
Джеймс Мур  
Тимоти Нисен  
Деннис Рилинг

Терри Роут Ричард  
Шмидт Норман Ф.  
Шнейдевинд Дэвид  
Шультц Бэзил Шерлунд  
Петер Вольднер  
Рональд Уэйд

Следующие лица вошли в состав комитета по голосованию:

Сиед Али  
Теодор К. Аткинсон  
Майкл Аугустон  
Роберт Е. Барри  
Лео Белтраччи  
Х. Рональд Берлак  
Ричард И. Бил  
Майкл А. Блэкледж  
Сандро Болонья  
Юрис Борзовс  
Кэтлин Л. Бриггс  
М. Скотт Бак  
Майкл Калдуэлл  
Джеймс Е. Кардоу  
Энрико А. Карраро  
Лоуренс Кэтчпол  
КэйтЧан  
Антонио М. Сику  
Тео Кларк  
Сильвайн Клермонт  
Розмари Коулман  
Вирджил Ли Купер  
В.В. Джефф Козенс  
Пол Р. Кролл  
Грегори Т. Дэйч  
Джеффри Дарнтон  
Таз Дотри  
Востиан К. Дерганк  
Перри Р. Де Вис  
Джеймс До  
Эвелин С. Доу  
Карл Эйнар Драгстедт  
Шерман Игле  
Кристоф Эберт  
Лео Эган  
Ричард Е. Фэйрли  
Джон У. Фендрик  
Джей Форстер  
Кирби Фортенберри  
Эва Фройнд  
Ричард К. Фрис  
Роджер Ю. Фуджи  
Адель Н. Ганнам  
Мэрилин Гинсберг-Финнер  
Джон Гарт Глинн  
Джулио Гонзалес Санц  
Л. М. Гюнтер

Дэвид А. Густафсон  
Джон Д.. Хагар Джон  
Харас Роберт Т. Харли  
Герберт Хехт Уидьям  
Хэдли Манфред Хайн  
Марк Хайнрих Марк  
Хэнли Дебра Херманн  
Джон У. Хорх Джерри  
Хуллер Петер Л. Ханг  
Джордж Джэклин  
Фрэнк В. Иоргенсен  
Уильям С. Джанк  
Джордж К. Камбик  
Ричард Карчич Рон С.  
Кеннет Юдит С.  
Кернер Роберт Дж.  
Кирзик Дуэйн Л.  
Книрк Шайе Кёниг  
Томас М. Курихара  
Джон Б. Лэйн Дж.  
Деннис Лоуренс Фанг  
Чинг Лим Уильям М.  
Ливли Джеймс Дж.  
Лонгбакко Дитер Лук  
Джон Лорд Стэн Маги  
Дэвид Мэйбор Гарольд  
Мэйнс Роберт А.  
Мартин Томо  
Мацубара Майк  
МакЭндрю Патрик Д.  
МакГрэй Кристофер  
МакМакен Джером У.  
Мерски Брет Майкл  
Алан Миллер Селиа Х.  
Моделл Джеймс У.  
Мур Павол Наврат  
Мирина Л. Олсон

Индрадеб П. Пал  
Алекс Полак  
Петер Т. Пун  
Лоуренс С. Пржибильский  
Кеннет Р. Птак Аннет Д.  
Рэйли Деннис Риллинг  
Эндрю П. Сэйдж Гельмут  
Сандмайер Стефен Р. Шах  
Ханс Шафер Норман  
Шнейдевинд Дэвид Дж.  
Шультц Лиза А. Селмон  
Роберт У. Шиллато Дэвид  
М. Зиверт Карл А. Сингер  
Джеймс М. Сивак Ричард  
С. Скай Нэнси М. Смит  
Мэлфорд Е. Смир Гарри  
М. Сид Алфред Р.  
Сорковитц Дональд У.  
Сова Лука Споторно  
Джулиа Стэсни Фред Дж.  
Страусе Кристин Браун  
Страйстик Тору Такешита  
Ричард Х. Тайер Букер  
Томас Патриция Треллу  
Теодор Дж. Урбанович  
Гленн Д. Венэблс Удо  
Вогес Дэвид Д. Уолден  
Долорес Уэллас Уильям  
М. Уолш Джон У. Уолс  
Камил С. Уайт-Партайн  
Скотт А. Витмайр П. А.  
Вольфганг Пол Р. Уорк  
Натали К. Йопконка Януш  
Залевски Джеральдин  
Зиммерман Петер Ф.  
Золль

На момент утверждения данной методики 25 июня 1998 года в состав Совета по Стандартам IEEE входили:

Ричард Дж. Холлеман, Председатель

Дональд Н. Хэйрман Вице-председатель

Юдит Горман, Секретарь

Сатиш А. Аггарвал  
Клайд Р. Кэмп  
Джеймс Т. Карло  
Гари Р. Энгман  
Гарольд И. Эпстайн  
Джей Форстер\*  
Томас Ф. Гаррити  
Рубен Д. Гарзон

Джеймс Х. Гарни  
Джим, Д Исаак Лоуэлл  
Г. Джонсон Роберт  
Кенелли И. Г. "Аль"  
Кинер Джозеф Л.  
Копфингер\* Стефан Р.  
Ламберт Джим  
Логотетис Дональд К.  
Логри

Л. Брюс МакКланг  
Луи-Франк По  
Рональд К. Петерсен  
Джеральд Х.  
Петерсон Джон Б.  
Рози Гари С.  
Робинсон Ганс И.  
Вайнрих Дональд У.  
Зипс

\*Заслуженный член в отставке

Валери И. Зеленти *Редактор*

*Проекта Стандартов IEEE*

## Содержание

1 . Краткий обзор.....	1
1.1 Область действия.....	1
2. Публикации .....	2
3. Определения .....	2
4. Критерии получения качественной SRS .....	3
4.1 Сущность SRS .....	3
4.2 Среда SRS .....	3
4.3 Характеристики качественной SRS .....	4
4.4 Совместная подготовка SRS.....	8
4.5 Развитие SRS .....	8
4.6 Макетирование .....	9
4.7 Встраивание структуры в SRS .....	9
4.8 Встраивание требований проекта в SRS .....	10
5. Разделы SRS.....	10
5.1 Введение (Раздел 1 SRS).....	11
5.2 Общее описание (Раздел 2 SRS) .....	12
5.3 Специфические требования (Раздел 3 SRS) .....	15
5.4 Дополнительная информация .....	19
Приложение А (информационное) Шаблоны SRS .....	21
Приложение Б (информационное) Указания по соответствию стандарту IEEE/EIA 12207/1-1997 .....	27

# **Методика составления спецификаций требований к программному обеспечению, рекомендуемая IEEE**

## **1. Краткий обзор**

Данная методика описывает рекомендуемые принципы составления спецификации требований к программному обеспечению. Она разделена на пять разделов. Раздел 1 указывает область действия этой методики. В разделе 2 перечисляются ссылки на другие стандарты. В разделе 3 даны определения используемых терминов. Раздел 4 предоставляет предварительную информацию для составления качественной SRS. В разделе 5 обсуждается каждая из необходимых частей SRS. Данная методика имеет два приложения; в одном из них приведены альтернативные шаблоны формата, а в другом обеспечиваются указания по обеспечению соответствия со стандартом IEEE/EIA 12207.1-1997.

### **1.1 Область действия**

Этот документ представляет рекомендуемую методику составления спецификаций требований к программному обеспечению. В нем описываются содержание и качественные характеристики правильно составленной спецификации требований к программному обеспечению (SRS) и представлено несколько типовых образцов SRS.

Данная методика имеет своей целью определение требований к разрабатываемому программному обеспечению, но также может применяться для обеспечения помощи в выборе внутренних и коммерческих программных изделий. Однако ее применение к уже разработанному программному обеспечению может оказаться непродуктивным.

При встраивании программного обеспечения в большую систему, такую, как например медицинское оборудование, может потребоваться обращение к другим изданиям кроме указанных в данной методике.

Данная рекомендуемая методика описывает процесс создания продукта и его содержание. Продуктом является SRS. Эта методика может использоваться для создания такой SRS непосредственно или может использоваться как модель для более специфических стандартов.

Данная методика не определяет какой-либо конкретный метод, терминологию или инструмент для подготовки SRS.

## 2. Публикации

Данная рекомендуемая методика должна использоваться вместе со следующими публикациями.

ASTM E1340-96, Руководство по стандартам по быстрому макетированию компьютеризованных систем.<sup>1</sup>

Стандарт IEEE 610.12-1990, Глоссарий стандартов IEEE по терминологии разработок программного обеспечения.<sup>2</sup>

Стандарт IEEE 730-1998, Стандарт IEEE по планам обеспечения качества программных средств.

Стандарт IEEE 730.1-1995, Руководство IEEE по планированию обеспечения качества программных средств.

Стандарт IEEE 828-1998, Стандарт IEEE по проектам управления конфигурацией программного обеспечения<sup>3</sup>

Стандарт IEEE 982.1-1988, Словарь стандартов IEEE по критериям создания надежного программного обеспечения.

Стандарт IEEE 982.2-1988, Руководство IEEE по использованию словаря стандартов IEEE по критериям создания надежного программного обеспечения.

Стандарт IEEE 1002-1987 (Повторно подтвержден в 1992), Классификация и систематизация Стандартов IEEE для стандартов разработок программного обеспечения.

Стандарт IEEE 1012-1998, Стандарт IEEE по аттестации и верификации программного обеспечения.

Стандарт IEEE 1012a-1998, Стандарт IEEE по аттестации и верификации программного обеспечения: План содержания к стандарту IEEE/EIA 12207.1-1997.<sup>4</sup>

Стандарт IEEE 1016-1998, Методика составления описаний разработок программного обеспечения, рекомендуемая IEEE.<sup>5</sup>

Стандарт IEEE 1028-1997, Стандарт IEEE по анализу программного обеспечения..

Стандарт IEEE 1042-1987 (Повторно подтвержден в 1993 году), Руководство IEEE по управлению конфигурацией программного обеспечения.

IEEE P1058/D2.1, Эскиз стандарта по планам управления проектами программного обеспечения, от 5 августа 1998.<sup>6</sup>

Стандарт IEEE 1058a-1998, Стандарт IEEE по планам управления проектами программного обеспечения: План содержания к стандарту IEEE/EIA 12207.1-1977.<sup>7</sup>

Стандарт IEEE 1074-1997, Стандарт IEEE по разработке процессов жизненного цикла программного обеспечения.

Стандарт IEEE 1233, Издание 1998 года, Руководство IEEE по разработкам спецификаций системных требований.<sup>8</sup>

<sup>1</sup> Публикации ASTM можно получить в Американском Обществе Тестирования и Материалов, 100 Ватт Harbor Drive, West Conshohocken. PA 19428-2959, USA.

<sup>2</sup> Публикации IEEE можно получить в Институте Инженеров по Электротехнике и Радиоэлектронике, 445 Hoes Lane, P.O. Box 1331. Piscataway, NJ 08855-1331, USA.

<sup>3</sup> По мере выхода этого стандарта в печать, стандарты IEEE 828-1998; IEEE 1012a-1998; IEEE 1016-1998 и IEEE 1233 1998 года издания утверждаются, но еще не издаются. Тем не менее, эскизы стандартов можно получить в IEEE. Ожидаемая дата публикации - осень 1998. За информацией о состоянии обращайтесь в Отдел Стандартов IEEE по телефону 1 (732) 562-3800.

<sup>4</sup> См. сноску 3.

<sup>5</sup> См. сноску 3.

<sup>6</sup> После утверждения стандарта IEEE P1058 Советом по Стандартам IEEE-SA, этот стандарт будет объединен со стандартом IEEE 1058a-1998 и опубликован как стандарт IEEE 1058, 1998 года издания. Утверждение ожидается 8 декабря 1998.

<sup>7</sup> По мере выхода этого стандарта в печать, стандарт IEEE 1058a-1998 утверждается, но еще не издается. Тем не менее, эскиз стандарта можно получить в IEEE. Ожидаемая дата публикации - декабрь 1998. За информацией о состоянии обращайтесь в Отдел Стандартов IEEE по телефону 1 (732) 562-3800. См. Сноску 6.

<sup>8</sup> См. Сноску 3.



### 3. Определения

В целом, определения терминов, используемых в данной рекомендуемой методике, соответствует определениям, приведенным в стандарте IEEE 610.12-1990. Определения, данные ниже, являются ключевыми терминами, поскольку они используются в данной методике.

**3.1 контракт:** обязательный официальный документ, согласованный заказчиком и поставщиком. Он включает технические и организационные требования, стоимость и план создания изделия. Контракт может также содержать неофициальную, но полезную информацию, такую как обязательства или ожидания участвующих сторон.

**3.2 заказчик:** лицо или лица, которые оплачивают изделие и обычно (но не обязательно) принимают решения относительно требований. В контексте данной рекомендуемой методики заказчик и поставщик могут быть членами одной и той же организации.

**3.3 поставщик:** лицо или лица, которые производят изделие для заказчика. В контексте данной рекомендуемой методики заказчик и поставщик могут быть членами одной и той же организации.

**3.4 пользователь:** лицо или лица, которые работают с изделием или непосредственно взаимодействуют с ним. Пользователь(-и) и заказчик(-и) часто не являются одним и тем же лицом(-ми).

### 4. Критерии создания качественной SRS

В этом разделе представлена предварительная информация, которую необходимо рассмотреть при составлении SRS. Она включает следующее:

- а) Сущность SRS;
- б) Среда SRS;
- в) Характеристики качественной SRS;
- г) Совместная подготовка SRS;
- д) Развитие SRS;
- е) Макетирование;
- ж) Внедрение структуры в SRS;
- з) Внедрение требований проекта в SRS.

#### 4.1 Сущность SRS

SRS - это спецификация для определенного программного изделия, программы или набора программ, которые выполняют определенные функции в специфической среде. SRS может состояться одним или более представителями поставщика, одним или более представителями заказчика, или обоими. Подраздел 4.4 рекомендует участие обоих.

Основными вопросами, которые должны рассматривать составитель (-ли) SRS, являются следующие:

- а) *Функциональные возможности.* Каковы предполагаемые функции программного обеспечения?
- б) *Внешние интерфейсы.* Как программное обеспечение взаимодействует с пользователями, аппаратными средствами системы, другими аппаратными средствами и другим программным обеспечением?
- в) *Рабочие характеристики.* Каково быстродействие, доступность, время отклика, время восстановления различных функций программного обеспечения и т.д.?
- г) *Атрибуты.* Каковы мобильность, правильность, удобство сопровождения, защищенность программного обеспечения и другие критерии?
- д) *Проектные ограничения, налагаемые на реализацию изделия.* Существуют ли требуемые стандарты на эффективном языке реализации, политика по сохранению целостности баз данных, ограничения ресурсов, операционная среда(-ы) и т.д.?

Составителю(-ям) SRS следует избегать размещения в SRS требований к разработке или проекту.

Рекомендуемое содержание SRS см. в Приложении 5.

## 4.2 Среда SRS

Важно рассмотреть ту роль, которую SRS играет в общем плане проекта, определяемого в стандарте IEEE 610.12-1990. По существу, программное обеспечение может содержать все функциональные возможности проекта или может являться частью большей системы. В последнем случае обычно имеется SRS, которая будет устанавливать интерфейсы между системой и частью ее программного обеспечения, и будет распространять внешние требования к характеристикам и функциям на эту часть программного обеспечения. Конечно, в этом случае SRS должна быть согласована и расширена в соответствии с этими системными требованиями.

Стандарт IEEE 1074-1997 описывает стадии жизненного цикла программного обеспечения и соответствующие входные данные для каждой стадии. Другие стандарты, такие как перечисленные в Разделе 2, относятся к другим частям жизненного цикла программного обеспечения и могут дополнять требования к программному обеспечению.

Так как SRS играет определенную роль в процессе разработки программного обеспечения, составителю (-ям) SRS следует проявлять осторожность, чтобы не выйти за пределы этой роли. Это означает, что SRS:

- а) Должна правильно определять все требования к программному обеспечению. Причиной существования какого-либо требования к программному обеспечению может являться характер решаемой задачи или особая характеристика проекта.
- б) Не должна описывать детали разработки или реализации. Они должны быть описаны на этапе разработки проекта.
- в) Не должна налагать дополнительные ограничения на программное обеспечение. Эти ограничения надлежащим образом определяются в других документах, таких как план обеспечения качества программных средств.

Следовательно, правильно составленная SRS ограничивает диапазон допустимых проектов, но не определяет какой-либо конкретный проект.

## 4.3 Характеристики правильно составленной SRS

SRS должна быть:

- а) Корректной;
- б) Однозначной;
- в) Полной;
- г) Непротиворечивой;
- д) Упорядоченной по ее значимости и/или устойчивости;
- е) Проверяемой;
- ж) Модифицируемой;
- з) Отслеживаемой.

### 4.3.1 Корректность

SRS является корректной, если, и только, если каждое требование, изложенное в ней, является требованием, которому должно удовлетворять программное обеспечение.

Не существует какого-либо средства или процедуры, которые гарантируют корректность. SRS должна сравниваться с любой качественной применимой спецификацией, такой как спецификация требований к системе, с другой документацией проекта и с другими применимыми стандартами, и должна гарантировать согласованность. В качестве альтернативы заказчик или пользователь может определить, правильно ли SRS отражает фактические потребности. Отслеживаемость делает эту процедуру более простой и менее подверженной ошибкам (см. 4.3.8).

### 4.3.2 Однозначность

SRS является однозначной, если и только, если каждое изложенное в ней требование может интерпретироваться только однозначно. Как минимум, для этого требуется, чтобы каждая характеристика конечного продукта была описана с использованием одного уникального термина.

В тех случаях, когда термин, используемый в специфическом контексте, может иметь множественные значения, этот термин должен быть включен в глоссарий, в котором его значение описывается более конкретно.

SRS является важной частью процесса составления требований или жизненного цикла программного обеспечения и используется в разработке, реализации, мониторинге проекта, верификации и проверке правильности, а также в обучении, как описано в стандарте IEEE 1074-1997. SRS должна быть однозначной как для тех, кто составляет ее, так и для тех, кто ее использует. Однако, эти группы часто не имеют одинаковую квалификацию и, следовательно, не имеют тенденции к описанию требований к программному обеспечению одним и тем же образом. Способы представления требований, которые улучшают спецификацию требований для разработчика, могут оказаться неэффективными в том, что они уменьшают их понимание пользователем, и наоборот.

В подразделах с 4.3.2.1 по 4.3.2.3 даются рекомендации, как избежать неоднозначности.

#### **4.3.2.1 "Ловушки" естественного языка**

Требования часто пишутся на естественном языке (например, английском). Естественный язык по существу является неоднозначным. Естественный язык SRS должен анализироваться независимой стороной с целью идентификации неоднозначного использования так, чтобы эту неоднозначность можно было скорректировать.

#### **4.3.2.2 Языки спецификаций требований**

Один из способов избежать неоднозначности, свойственной естественному языку, состоит в том, чтобы составлять SRS на особом языке спецификаций требований. Языковые процессоры автоматически обнаруживают многие лексические, синтаксические и семантические ошибки.

Одним из недостатков использования таких языков является длительность их изучения. К тому же, многие пользователи, не имеющие отношения к технике, считают их непонятными. Кроме того, эти языки имеют тенденцию к улучшению при точном выражении некоторых типов требований и обращении к некоторым типам систем. Таким образом, они могут влиять на требования неумолимым образом.

#### **4.3.2.3 Инструменты представления требований**

В целом, методы составления требований и языки, а также инструменты, которые их поддерживают, разделяются на три основных категории - объект, процесс и поведенческие характеристики. При использовании объектно-ориентированных методов, требования классифицируются языком реальных объектов, их атрибутов и функций, выполняемых этими объектами. Методы, ориентированные на процесс, классифицируют требования по иерархии функций, которые сообщаются через потоки данных. Поведенческие подходы описывают внешнее поведение системы языком неких абстрактных понятий (таких как исчисления предикатов), математических функций или конечных автоматов.

Степень, до которой такие средства и методы могут быть полезны в подготовке SRS, зависит от объема и сложности программы. В данном документе не делается никаких попыток описать или одобрить какой-либо специфический инструмент.

При использовании любого из этих подходов самым оптимальным будет придерживаться описания на естественном языке. При этом заказчики, незнакомые с перечисленными понятиями, будут в состоянии понять SRS.

#### **4.3.3 Завершенность**

SRS является полной, если и только, если она включает следующие элементы:

- а) Все существенные требования, независимо от того, относятся ли они к функциональным возможностям, рабочим характеристикам, проектным ограничениям, атрибутам или внешним интерфейсам. В частности, должны быть подтверждены и обработаны любые внешние требования, налагаемые спецификацией системы.

- б) Определение откликов программного обеспечения на все классы входных данных, которые могут быть реализованы, во всех возможных ситуациях. Следует заметить, что важно определить отклики, как на допустимые, так и недопустимые входные значения.
- в) Полные обозначения и ссылки на все рисунки, таблицы и схемы в SRS и определение всех терминов и единиц измерения.

#### 4.3.3.1 Использование условия TBD

Любая SRS, в которой используется фраза "должно быть определено" (условие TBD), не является полной SRS. Однако, иногда условие TBD необходимо и должно сопровождаться:

- а) Описанием условий, являющихся причиной возникновения условия TBD (например, почему ответ не известен) так, чтобы ситуацию можно было разрешить;
- б) Описанием того, что должно быть сделано, чтобы исключить условие TBD, кто ответственен за его устранение и когда оно должно быть устранено.

#### 4.3.4 Непротиворечивость

Непротиворечивость обозначает внутреннюю непротиворечивость. Если SRS не согласовывается с каким-то документом более высокого уровня, таким как, например, спецификации системных требований, то она является некорректной (см. 4.3.1).

##### 4.3.4.1 Внутренняя непротиворечивость

SRS является внутренне непротиворечивой, если и только, если никакой набор отдельных требований, описанных в ней, не находится в противоречии с ней. Тремя типами вероятных конфликтов в SRS являются следующие:

- а) Могут входить в конфликт заданные характеристики реальных объектов. Например,
  - 1) Формат отчета о выходных данных может быть описан в одном требовании в табличном виде, а в другом - в текстовом.
  - 2) Одно требование может устанавливать, что все индикаторы должны быть зелеными, в то время как в другом может быть указано, что все индикаторы должны быть синими.
- б) Между двумя заданными действиями может существовать логический или временной конфликт. Например,
  - 1) Одно требование может устанавливать, что программа будет добавлять два входа, а другое может указывать, что программа будет умножать их.
  - 2) Одно требование может устанавливать, что "А" должно всегда следовать за "Б", в то время как другое может требовать, чтобы события "А" и "Б" происходили одновременно.
- в) Два или более требований могут описывать один и тот же реальный объект, но использовать для этого объекта различные условия. Например, запрос программы о вводе пользователем может называться "подсказкой" в одном требовании и "репликой" в другом. Использование стандартной терминологии и определений поддерживает непротиворечивость.

#### 4.3.5 Упорядочивание по значимости и/или устойчивости

SRS является упорядоченной по значимости и/или устойчивости, если каждое требование в ней имеет идентификатор, указывающий или значимость или устойчивость этого конкретного требования.

Как правило, все требования, которые касаются программного изделия, не являются важными в равной степени. Некоторые требования могут быть необходимыми, особенно для приложений, критичных в течение жизненного цикла, в то время как другие могут быть желательными.

Каждое требование в SRS должно быть идентифицировано, чтобы сделать эти различия четкими и явными. Идентификация требований следующим образом помогает:

- а) заказчикам более тщательно рассмотреть каждое требование, что часто позволяет разъяснить любые скрытые допущения, которые могут быть заключены в них.
- б) разработчикам принять правильные проектные решения и приложить соответствующие усилия к различным составляющим программного изделия.

#### 4.3.5.1 Степень устойчивости

Один метод идентификации требований использует величину устойчивости. Устойчивость может быть выражена с точки зрения числа ожидаемых изменений для любого требования, основанных на опыте или знании предстоящих событий, которые влияют на организацию, функции, и пользователей, поддерживаемых системой программного обеспечения.

#### 4.3.5.2 Степень необходимости

Другой способ упорядочения требований состоит в том, чтобы различать классы требований как необходимые, условные и необязательные.

- а) *Необходимые*. Подразумевают, что программное обеспечение не будет пригодным, если эти требования не будут обеспечены согласованным образом.
- б) *Условные*. Подразумевают, что эти требования расширяют программное изделие, но не сделают его непригодным при их отсутствии.
- в) *Необязательные*. Подразумевают класс функций, которые могут заслуживать или не заслуживать внимания. Это дает поставщику возможность предложить что-либо, что выходит за пределы SRS.

#### 4.3.6 Проверяемость

SRS является проверяемой, если и только, если каждое требование, изложенное в ней, может быть проверено. Требования являются проверяемым, если и только, если существует некий конечный эффективный процесс, используя который пользователь или машина могут убедиться, что программное изделие удовлетворяет этому требованию. В целом, любое неоднозначное требование не проверяемо.

Непроверяемые требования включают формулировки типа "работает хорошо", "хороший интерфейс с пользователем" и "обычно должно происходить". Эти требования не могут быть проверены, так как невозможно определить термины "хороший", "хорошо" или "обычно". Утверждение о том, что "программа никогда не должна заикливаться" является непроверяемым, так как тестирование этого свойства теоретически невозможно.

Примером проверяемого утверждения является следующее:

*Выходные данные программы должны вырабатываться в пределах 20 секунд в течение 60 % временного интервала события; и должны вырабатываться в пределах 30 секунд в течение 100 % временного интервала события.*

Это утверждение может быть проверено, так как в нем используются конкретные термины и измеримые величины.

Если нельзя изобрести метод, чтобы определить, отвечает ли программное обеспечение определенному требованию, то это требование следует исключить или пересмотреть.

### 4.3.7 Модифицируемость

SRS является модифицируемой, если и только, если ее структура и стиль таковы, что любые изменения требований могут быть выполнены легко, полностью и непротиворечивым образом при сохранении структуры и стиля. Как правило, модифицируемость требует, чтобы SRS:

- а) Имела связанную и легкую в использовании структуру с оглавлением, алфавитным указателем и явно выраженными перекрестными ссылками;
- б) Не была избыточной (то есть, одно и то же требование не должно появляться в SRS более чем в одном месте);
- в) Выражала каждое требование отдельно, не смешивая его с другими требованиями.

Избыточность сама по себе не является ошибкой, но она легко может привести к появлению ошибок. Иногда избыточность может помочь сделать SRS более читаемой, но тогда могут возникнуть проблемы при модификации избыточного документа. Например, требование может быть изменено только в одном из тех мест, где оно появляется. Тогда SRS становится противоречивой. Каждый раз, когда избыточность необходима, SRS должна включать явные перекрестные ссылки, чтобы сделать ее модифицируемой.

### 4.3.8 Отслеживаемость

SRS является отслеживаемой, если четко прослеживается источник каждого из ее требований и если она облегчает обращение к каждому из требований при дальнейшей разработке или модернизации документации. Рекомендуются следующие два типа отслеживаемости:

- а) *Обратная отслеживаемость (то есть, к предыдущим стадиям разработки).* Зависит от каждого требования, которое в явном виде ссылается на его источник в более ранних документах.
- б) *Прямая отслеживаемость (то есть, ко всем документам, порождаемым SRS).* Зависит от каждого требования в SRS, имеющего однозначно определенное имя или номер ссылки.

Прямая отслеживаемость SRS особенно важна, когда программное изделие вступает в стадию функционирования и сопровождения. По мере изменения кода и проектных документов необходимо иметь возможность определить полный набор требований, на которые могут повлиять эти изменения.

## 4.4 Совместная подготовка SRS

Процесс разработки программного обеспечения должен начинаться с соглашения между поставщиком и заказчиком по вопросу о том, что должно выполнять завершено программное обеспечение. Это соглашение в форме SRS должно быть подготовлено совместно. Это важно, так как обычно ни заказчик, ни поставщик не имеют достаточной квалификации, чтобы составить качественную SRS по отдельности.

- а) Заказчики обычно не понимают в достаточной степени процесс проектирования и разработки программного обеспечения, чтобы составить SRS, пригодную для использования.
- б) Поставщики обычно не понимают в достаточной степени проблемы заказчика и поле приложения усилий, чтобы определить требования для системы, которая удовлетворит заказчика.

Следовательно, заказчик и поставщик должны работать вместе, чтобы создать хорошо написанную и полностью понятную SRS.

Особая ситуация возникает, когда система и программное обеспечение определяются одновременно. Тогда функциональные возможности, интерфейсы, рабочие характеристики и другие атрибуты и ограничения программного обеспечения не предопределяются, а задаются совместно и подлежат согласованию и изменению. При этом более трудно, но не менее важно сделать так, чтобы SRS удовлетворяла характеристикам, изложенным в пункте 4.3. В частности, SRS, которая не соответствует требованиям спецификации ее родительской системы, является не корректной.

Данная методика специально не обсуждает стиль, исполнение языка или методы качественного написания. Однако весьма важно, чтобы SRS была хорошо написана. В качестве руководства могут использоваться технические книги по общим методам написания.

#### 4.5 Развитие SRS

По мере разработки программного изделия может возникнуть необходимость в развитии SRS. Может оказаться невозможным определить некоторые детали во время начала разработки проекта (например, может оказаться невозможным определить все форматы экрана для интерактивной программы на стадии выработки требований). В результате в SRS обнаруживаются неточности, недостатки и погрешности, являющиеся результатом дополнительных изменений.

Двумя главными критериями в этом процессе являются следующие:

- а) Требования должны быть определены в том объеме и с той тщательностью, как они известны на текущий момент, даже если эволюционные изменения могут быть предсказаны как неизбежные. Должен быть отмечен тот факт, что они не являются полными.
- б) Должен быть инициализирован формальный процесс изменений, чтобы идентифицировать, управлять, проследивать и составлять отчет о запроецированных изменениях. Утвержденные изменения требований должны быть включены в SRS таким образом, чтобы:
  - 1) Обеспечить точную и полную проверку изменений;
  - 2) Обеспечить анализ текущих и замененных частей SRS..

#### 4.6 Макетирование

Макетирование часто используется на этапе выработки требований проекта. Существуют многие инструментальные средства, которые позволяют очень быстро и легко создать прототип, проявляющий некоторые характеристики системы. См. также ASTM E1340-96.

Прототипы удобны по следующим причинам:

- а) Заказчик может более удобным образом наблюдать прототип и оценивать его, нежели читать SRS и оценивать ее. Таким образом, прототип обеспечивает быструю обратную связь.
- б) Прототип отображает непредвиденные аспекты поведения систем. Таким образом, он генерирует не только ответы, но также и новые вопросы. Это помогает сосредоточиться на SRS.
- в) SRS, базирующаяся на прототипе, имеет тенденцию подвергаться меньшим изменениям во время разработки, сокращая, таким образом ее длительность.

Прототип должен использоваться как способ установления требований к программному обеспечению. Некоторые характеристики, такие как форматы экрана или отчета, могут быть выделены непосредственно из прототипа. Другие требования могут быть выведены посредством проведения экспериментов с прототипом.

#### 4.7 Встраивание структуры в SRS

Любое требование определяет внешне видимую функцию или атрибут системы. Структура описывает специфический подкомпонент системы и/или его интерфейс с другими подкомпонентами. Составитель (-ли) SRS должны ясно осознавать разницу между идентификацией требуемых ограничений структуры и проектированием специфической структуры. Следует заметить, что каждое требование в SRS ограничивает варианты структуры. Тем не менее, это не означает, что каждое требование является структурой.

SRS должна указывать, какие функции должны выполняться и с какими данными, чтобы получить то, что является результатом, в каком месте и для кого. SRS должна фокусироваться на выполняемых услугах. SRS обычно не должна указывать элементы структуры, такие как:

- а) Разбиение разделов программного обеспечения на модули;
- б) Присваивание функций модулям;
- в) Описание потока данных или управления между модулями;
- г) Выбор структур данных.

#### 4.7.1 Необходимые требования к структуре

В особых случаях некоторые требования могут строго ограничивать структуру. Например, требования к защищенности или безопасности могут отражать непосредственно в структуре потребность в:

- а) Сохранении некоторых функций в отдельных модулях;
- б) Разрешении только ограниченной связи между некоторыми областями программы;
- в) Проверке целостности данных для критических переменных.

Примерами применимых ограничений структуры являются физические требования, требования к техническим характеристикам, стандарты разработки программного обеспечения и стандарты обеспечения качества программных средств.

Поэтому, требования должны излагаться исключительно с внешней точки зрения. При использовании моделей для иллюстрации требований, помните, что модель только указывает внешнее поведение, но не определяет структуру.

#### 4.8 Встраивание требований к проекту в SRS

SRS должна относиться к программному изделию, а не к процессу его создания.

Требования к проекту представляют соглашение между заказчиком и поставщиком по договорным вопросам, имеющим отношение к созданию программного обеспечения и, таким образом, не должны включаться в SRS. Они обычно включают такие позиции, как

- а) Стоимость;
- б) Графики поставки;
- в) Процедуры составления отчетов;
- г) Методы разработки программного обеспечения;
- д) Обеспечение качества;
- е) Критерии утверждения и верификации;
- ж) Процедуры приемки.

Требования к проекту определяются в других документах, обычно в плане разработки программного обеспечения, плане обеспечения качества программных средств или формулировке работ.

#### 5. Части SRS

В этом разделе обсуждается каждая из необходимых частей SRS. Эти части показаны на рисунке 1 в виде эскиза, который может служить примером для составления SRS..

Несмотря на то, что SRS не должна повторять этот эскиз или использовать наименования, приведенные здесь для ее частей, качественно составленная SRS должна включать всю информацию, обсуждаемую здесь.



## Содержание

- 1..Введение
  - 1.1 Назначение
  - 1.2 Область действия
  - 1.3 Определения, акронимы и сокращения
  - 1.4 Публикации
  - 1.5 Краткий обзор
2. Полное описание
  - 2.1 Перспектива изделия
  - 2.2 Функции изделия
  - 2.3 Характеристики пользователя
  - 2.4 Ограничения
  - 2.5 Допущения и зависимости
3. Специфические требования (Объяснения возможных специфических требований см. в пунктах с 5.3.1 по 5.3.8. Несколько различных способов организации этого раздела SRS см. в Приложении)

Приложения

Алфавитный указатель

---

### Рисунок 1 - Эскиз макета SRS 5.1

## 5.1 Введение (Раздел 1 SRS)

Введение SRS должно обеспечивать краткий обзор всей SRS. Оно должно содержать следующие подразделы:

- а) Назначение;
- б) Область действия;
- в) Определения, акронимы и сокращения;
- г) Публикации;
- д) Краткий обзор.

### 5.1.1 Назначение (Подраздел 1.1 SRS)

Этот подраздел должен:

- а) Обрисовать назначение SRS;
- б) Указать аудиторию, для которой предназначена SRS.

### 5.1.2 Область действия (Подраздел 1.2 SRS)

Этот подраздел должен:

- а) Идентифицировать программное изделие (-я), которое будет создаваться под именем (например, Host DMBS (Главная система управления базой данных), Генератор отчетов и т.д.);
- б) Объяснять, что программное изделие будет и, в случае необходимости, не будет делать;
- в) Описывать применение задаваемого программного обеспечения, включая связанные с ним выгоды, цели и задачи;
- г) Согласовываться с аналогичными формулировками в спецификациях более высокого уровня (например, со спецификацией системных требований), если они существуют.

### **5.1.3 Определения, акронимы и сокращения (Подраздел 1.3 SRS)**

В этом подразделе должны быть представлены определения всех терминов, акронимы и сокращения, необходимые для правильной интерпретации SRS. Эта информация может быть обеспечена ссылками на одно или более приложений в SRS или ссылкой на другие документы.

### **5.1.4 Публикации (Подраздел 1.4 SRS)**

Этот подраздел должен:

- а) Представить полный список всех документов, на которые делаются ссылки в других местах SRS;
- б) Идентифицировать каждый документ по заголовку, номеру отчета (если применяется), дате и издательской организации;
- в) Определить источники, из которых могут быть получены ссылки.

Эту информацию можно обеспечить ссылкой на приложение или другой документ.

### **5.1.5 Краткий обзор (Подраздел 1.5 SRS)**

Этот подраздел должен:

- а) Описать, какие оставшиеся части содержатся в SRS;
- б) Объяснить, как организована SRS.

## **5.2 Общее описание (Раздел 2 SRS)**

Этот раздел SRS должен описывать общие факторы, которые влияют на программное изделие (-я) и требования, предъявляемые к нему. Этот раздел не устанавливает конкретные требования. Вместо этого, он обеспечивает предварительные сведения о тех требованиях, которые подробно определяются в разделе 3 SRS, и делает их более простыми для понимания.

Этот раздел обычно состоит из шести подразделов, а именно:

- а) Перспектива изделия;
- б) Функции изделия;
- в) Характеристики пользователей;
- г) Ограничения;
- д) Допущения и зависимости;
- е) Разделение требований.

### **5.2.1 Перспектива изделия ( Подраздел 2.1 SRS)**

Этот подраздел SRS должен оценивать изделие в перспективе с другими, связанными с ним изделиями. Если изделие является независимым и полностью автономным, оно так и должно быть сформулировано в документе. Если SRS определяет изделие, которое является компонентом большей системы, как это часто бывает, то данный подраздел должен устанавливать связь требований этой большей системы с функциональными возможностями программного обеспечения, и должен идентифицировать интерфейсы между этой системой и программным обеспечением.

Может оказаться полезным использование блок-схемы, показывающей главные компоненты большей системы, соединений и внешних интерфейсов.

Этот подраздел должен также описывать как программное обеспечение функционирует в рамках различных ограничений. Например, эти ограничения могут включать:

- а) Системные интерфейсы;
- б) Интерфейсы пользователя;
- в) Аппаратные интерфейсы;
- г) Интерфейсы программного обеспечения;
- д) Интерфейсы связи;
- е) Память;
- ж) Операции;
- з) Требования по адаптации места использования.

#### **5.2.1.1 Системные интерфейсы**

Этот подраздел должен перечислять каждый системный интерфейс и идентифицировать функции программного обеспечения, выполняющие системные требования, и описание интерфейса для согласования системы.

#### **5.2.1.2 Интерфейсы пользователя**

Этот подраздел должен указывать следующее:

- а) *Логические характеристики каждого интерфейса между программным изделием и его пользователями.* Это включает характеристики конфигурации (например, требуемые форматы экрана, компоновку страницы или окна, содержание любых отчетов или меню, или наличие программируемых функциональных клавиш), необходимые для выполнения требований к программному обеспечению.
- б) *Все аспекты оптимизации интерфейса с пользователем, который должен использовать систему.* Они могут просто включать список разрешений и запрещений различных способов представления системы пользователю. Одним из примеров может быть требование, предъявляемое к опции длинных или коротких сообщений об ошибках. Подобно всем другим, эти требования должны быть проверяемыми, например, "машинистка 4-го класса может выполнить задачу X через Z минут после 1 часа тренировки", а не "машинистка может выполнить задачу X" (Это также может быть определено в Атрибутах Системы Программного Обеспечения в разделе, озаглавленном Простота использования)

#### **5.2.1.3 Аппаратные интерфейсы**

Этот подраздел должен определять логические характеристики каждого интерфейса между программным изделием и аппаратными компонентами системы. Он включает конфигурационные характеристики (число портов, наборы команд и т.д.). Он также охватывает вопросы, касающиеся того, какие устройства должны поддерживаться, как они должны поддерживаться и протоколы. Например, поддержка терминала может указывать полноэкранную поддержку в противоположность строчной поддержке.

#### **5.2.1.4 Интерфейсы программного обеспечения**

Этот подраздел должен определять использование других необходимых программных изделий (например, системы управления данными, операционной системы или пакета математических программ) и интерфейсов с другими прикладными системами (например, связь между системой дебиторской задолженности и общей системой финансового учета). Для каждого требуемого программного изделия необходимо обеспечить следующее:

- Имя;
- Мнемонический код;
- Номер спецификации;
- Номер версии;
- Источник.

Для каждого интерфейса необходимо обеспечить следующее:

- а) Обсуждение назначения интерфейса программного обеспечения в отношении программного изделия.
- б) Определение интерфейса с точки зрения содержания и формата сообщений. Нет необходимости в детализации любого хорошо задокументированного интерфейса, но при этом требуется выполнить ссылку на документ, определяющий интерфейс.

### **5.2.1.5 Интерфейсы связи**

Этот подраздел должен определять различные интерфейсы связи, такие как локальные сетевые протоколы и т.д.

### **5.2.1.6 Ограничения памяти**

Этот подраздел должен определять любые применимые характеристики и ограничения первичной и вторичной памяти.

### **5.2.1.7 Операции**

Этот подраздел должен определять стандартные и специальные операции, требуемые пользователем, такие как:

- а) Различные режимы операций в организации пользователя (например, операции, инициализируемые пользователем);
- б) Периоды интерактивных операций и периоды автоматических операций;
- в) Функции поддержки обработки данных;
- г) Операции по резервированию и восстановлению.

ПРИМЕЧАНИЕ - Этот подраздел иногда указывается как часть раздела Интерфейсы пользователя.

### **5.2.1.8 Требования к адаптации места использования**

Этот подраздел должен:

- а) Определять требования к любым данным или последовательностям инициализации, которые являются специфическими для данного места, задачи, или режима работы (например, значения сетки, пределы надежности и т.д.);
- б) Определять характеристики места или характеристики, связанные с задачей, которые следует изменить, чтобы адаптировать программное обеспечение к конкретной установке.

## **5.2.2 Функции изделия (Подраздел 2.2 SRS)**

Этот подраздел SRS должен представлять сводку основных функций, выполняемых программным обеспечением. Например, SRS для программы бухгалтерского учета может использовать эту часть, чтобы обратиться к сопровождению счетов заказчика, отчету заказчика и подготовке счетов без упоминания большого количества деталей, которые требуются каждой из этих функций.

Иногда сводка функций, которая необходима для этой части, может быть взята непосредственно из раздела спецификации более высокого уровня (если он существует), который назначает отдельные функции программному изделию. Заметьте, что в целях обеспечения ясности:

- а) Функции должны быть организованы таким способом, который делает перечень функций понятным заказчику или любому человеку, читающему документ впервые.
- б) Могут быть использованы текстовые или графические методы, чтобы показать различные функции и их связи. Такая схема не предназначена для того, чтобы показать структуру изделия, а просто показывает логические связи между переменными.

### 5.2.3 Характеристики пользователя (Подраздел 2.3 SRS)

Этот подраздел SRS должен описывать общие характеристики пользователей изделия, включающие образовательный уровень, опыт и специальные технические знания. Он не должен использоваться для формулировки конкретных требований, а должен представлять основания, по которым некоторые специфические требования далее определяются в разделе 3 SRS.

### 5.2.4 Ограничения (Подраздел 2.4 SRS)

Этот подраздел SRS должен обеспечить общее описание любых других позиций, которые будут ограничивать опции разработчика. Они включают:

- а) Регулирующие политики;
- б) Аппаратные ограничения (например, требования к синхронизации сигналов);
- в) Интерфейсы с другими приложениями;
- г) Параллельные операции;
- д) Функции контроля;
- е) Функции управления;
- ж) Требования к языку более высокого порядка;
- з) Протоколы квитирования сигналов (например, XON-XOFF..ACK-NACK);
- и) Требования к надежности;
- к) Критичность применения;
- л) Критерии безопасности и защиты.

### 5.2.5 Допущения и зависимости (Подраздел 2.5 SRS)

В этом подразделе SRS должны быть перечислены все факторы, которые влияют на требования, сформулированные в SRS. Эти факторы не являются проектными ограничениями программного обеспечения, это, скорее, любые их изменения, которые могут воздействовать на требования в SRS. Например, может быть сделано допущение, что определенная операционная система будет присутствовать на аппаратных средствах, предназначенных для программного изделия. Если в действительности операционная система отсутствует, тогда SRS должна быть изменена соответствующим образом.

### 5.2.6 Распределение требований (Подраздел 2.6 SRS)

Этот подраздел SRS должен идентифицировать требования, которые могут быть отложены до появления будущих версий системы.

## 5.3 Специфические требования (Раздел 3 SRS)

Этот раздел SRS должен содержать все требования к программному обеспечению на уровне детализации, достаточной, чтобы дать проектировщикам возможность разработать систему, удовлетворяющую этим требованиям, а испытателям - убедиться, что система удовлетворяет этим требованиям. В этом разделе каждое сформулированное требование должно восприниматься пользователями, операторами или другими внешними системами. Эти требования должны включать как минимум описание каждого входного воздействия (стимула) на систему, каждого выходного сигнала (отклика) системы и всех функций, выполняемых системой в ответ на входное воздействие или для поддержания выходного сигнала. Поскольку часто эта часть является самой большой и наиболее важной частью SRS, применяются следующие принципы:

- а) Специфические требования должны быть сформулированы в соответствии со всеми характеристиками, описанными в пункте 4.3.
- б) Специфические требования должны иметь перекрестные ссылки к более ранним документам, к которым они имеют отношение.
- в) Все требования должны быть однозначно идентифицируемы.
- г) Необходимо уделить особое внимание оформлению требований, чтобы максимизировать их удобочитаемость.

Перед исследованием специфических способов организации требований полезно осмыслить различные позиции, которые включают требования, как описано в пунктах с 5.3.1 по 5.3.7.

### 5.3.1 Внешние интерфейсы

Этот раздел должен представлять детализированное описание всех входных и выходных данных системы программного обеспечения. Он должен дополнять описания интерфейса, содержащиеся в пункте 5.2 и не должен повторять эту информацию.

Он должен включать содержание и формат следующим образом:

- а) Наименование позиции;
- б) Описание назначения;
- в) Источник входных данных или адресат выходных данных;
- г) Допустимый диапазон, точность и/или допуск;
- д) Единицы измерения;
- е) Синхронизация;
- ж) Связи с другими входами/выходами;
- з) Форматы/организация экрана;
- и) Форматы/организация окна;
- к) Форматы данных;
- л) Форматы команд;
- м) Сообщения о конце.

### 5.3.2 Функции

Функциональные требования должны определять фундаментальные операции, которые должны выполняться программным обеспечением при принятии и обработке входных данных и обработке и генерации выходных данных. Они обычно перечисляются как формулировки типа "должно", начинающиеся с "Система должна .....",

Они включают:

- а) Проверку достоверности по входам;
- б) Точную последовательность операций;
- в) Отклики на ненормальные ситуации, включая:
  - 1) Переполнение
  - 2) Средства связи
  - 3) Обработка и устранение ошибок;
- г) Влияние параметров;
- д) Связь выходов с входами, включая:
  - 1) Последовательности ввода/вывода
  - 2) Формулы для преобразования ввода-вывода..

Может оказаться удобным разбить разделы функциональных требований на подфункции или подпроцессы. Это не означает, что структура программного обеспечения будет разделена таким же образом.

### 5.3.3 Требования к рабочим характеристикам

В этом подразделе должны быть определены требования к статическим и динамическим числовым характеристикам, включаемым в программное обеспечение или во взаимодействие пользователя с программным обеспечением в целом. Статические числовые требования могут включать:

- а) Число поддерживаемых терминалов;
- б) Число одновременно поддерживаемых пользователей;
- в) Количество и тип обрабатываемой информации.

Требования к статическим числовым характеристикам иногда указываются в отдельном разделе, озаглавленном "Диапазон значений".

Требования к динамическим числовым характеристикам могут включать, например, количество групповых операций и задач и количество данных, которые будут обрабатываться в пределах определенных периодов времени для нормальных и пиковых условий рабочей нагрузки.

Все эти требования должны быть установлены в измеряемых терминах.

Например,

*95 % групповых операций должно обрабатываться не более чем за 1 с. а не*

*Оператор не должен ждать завершения групповой операции.*

**ПРИМЕЧАНИЕ** - Числовые ограничения, применяемые к одной определенной функции, обычно указываются как часть описания подпункта по обработке этой функции.

### **5.3.4 Логические требования к базе данных**

В этом подразделе должны быть определены логические требования для любой информации, которая должна размещаться в базе данных. Она может включать следующее:

- а) Типы информации, используемой различными функциями;
- б) Частота использования;
- в) Возможности доступа;
- г) Информационные объекты и их связи;
- д) Ограничения целостности;
- е) Требования к сохранности данных.

### **5.3.5 Проектные ограничения**

Здесь должны указываться проектные ограничения, которые могут налагаться другими стандартами, аппаратными ограничениями и т.д.

#### **5.3.5.1 Согласованность стандартов**

Этот подраздел должен определять требования, происходящие из существующих стандартов или инструкций. Они могут включать следующее:

- а) Формат отчета;
- б) Поименование данных;
- в) Процедуры учета;
- г) Контроль трассировки.

Например, может быть указано требование для программного обеспечения в отношении операций по отслеживанию деятельности по обработке. Такие трассировки необходимы, чтобы обеспечить соответствие минимальным регулирующим или финансовым стандартам для некоторых прикладных задач. Требование к контролю трассировки может, например, утверждать, что все изменения базы данных платежных ведомостей должны быть записаны в файле трассировки с предыдущими и последующими значениями.

### **5.3.6 Атрибуты системы программного обеспечения**

Существует ряд атрибутов программного обеспечения, которые могут служить в качестве требований. Важно, чтобы необходимые атрибуты были определены таким образом, чтобы их выполнение можно было объективно проверить. В подпунктах с 5.3.6.1 по 5.3.6.5 приведен частичный перечень примеров.

### **5.3.6.1 Надежность**

Этот подраздел должен определять факторы, необходимые для установления требуемой надежности системы программного обеспечения при поставке.

### **5.3.6.2 Доступность**

Этот подраздел должен определять факторы, необходимые для обеспечения заданного уровня доступности для всей системы, такие как контрольная точка, восстановление и перезапуск.

### **5.3.6.3 Защита**

Этот подраздел должен определять факторы, которые защищают программное обеспечение от случайного или злонамеренного доступа, использования, изменения, разрушения или раскрытия. Специфические требования в этой области могут включать потребность в:

- а) Использовании некоторых методов криптографии;
- б) Сохранении специфического файла регистрации или наборов данных истории;
- в) Назначении некоторых функций различным модулям;
- г) Ограничении связи между некоторыми областями программы;
- д) Проверке целостности данных для критических переменных.

### **5.3.6.4 Удобство сопровождения**

Этот подраздел должен определять атрибуты программного обеспечения, которые относятся к простоте сопровождения самого программного обеспечения. Может иметься некоторое требование к определенной модульности системы, интерфейсам, степени интеграции и т.д. Требования не должны указываться в этом подразделе только потому, что они считаются хорошими методиками проектирования.

### **5.3.6.5 Мобильность**

Этот подраздел должен определять атрибуты программного обеспечения, которые относятся к простоте перенесения программного обеспечения на другие главные машины и/или операционные системы. Они могут включать следующее:

- а) Процентное соотношение компонентов с кодом, зависящим от главной машины;
- б) Процентное соотношение кода, зависящего от главной машины;
- в) Использование языка переноса программ;
- г) Использование определенного компилятора или подмножества языка;
- д) Использование определенной операционной системы.

## **5.3.7 Организация специфических требований**

Для нетривиальных систем имеется тенденция к расширению детализированных требований. По этой причине рекомендуется тщательно рассмотреть их организацию таким способом, который будет оптимальным для понимания. Не существует одного оптимального способа организации для всех систем. В разделе 3 SRS представлены различные способы организации требований для различных классов систем. Некоторые из них описаны в пунктах с 5.3.7.1 по 5.3.7.7.

### **5.3.7.1 Режим системы**

Некоторые системы ведут себя совершенно по-разному в зависимости от режима работы. Например, система управления может иметь различные наборы функций в зависимости от режима: обучение, нормальный режим или аварийный. При организации этого раздела по режимам следует использовать шаблон, представленную в приложении А.1 или А.2. Выбор зависит оттого, являются ли интерфейсы и характеристики зависимыми от режима.



### **5.3.7.2 Класс пользователей**

Некоторые системы обеспечивают различные наборы функций для разных классов пользователей. Например, система управления лифтом предоставляет различные возможности для пассажиров, обслуживающего персонала и пожарных. При организации этого раздела по классам пользователей следует использовать шаблон, представленный в приложении А.3.

### **5.3.7.3 Объекты**

Объекты - это реальные сущности, которые имеют аналог в пределах системы. Например, в системе контроля за пациентом объекты включают пациентов, датчики, медсестер, помещения, врачей, лекарства и т.д. С каждым объектом связан набор атрибутов (этого объекта) и функций (выполняемых этим объектом). Эти функции также называются услугами, методами или процессами. При организации данного раздела по объектам следует использовать шаблон, представленный в приложении А.4. Обратите внимание, что наборы объектов могут совместно использовать атрибуты и услуги. Они сгруппированы вместе по классам.

### **5.3.7.4 Свойство**

Свойство - это услуга, требуемая системой извне, для которой может оказаться необходимой последовательность входных воздействий для получения желаемого результата. Например, в телефонной системе свойства включают локальный вызов, переадресацию вызовов и циркулярный вызов. Каждое свойство в целом описывается в последовательности пар входное воздействие(стимул)-отклик. При организации этого раздела по свойствам следует использовать шаблон, представленный в приложении А.5.

### **5.3.7.5 Стимул**

Некоторые системы могут быть лучше всего организованы посредством описания их функций на языке стимулов. Например, функции автоматической системы посадки самолета могут быть организованы в разделы по энергетическим потерям, сдвигу ветра, внезапному изменению направления качения, избыточной вертикальной скорости и т.д. При организации этого раздела по стимулам следует использовать шаблон, представленный в приложении А.6.

### **5.3.7.6 Отклик**

Некоторые системы могут быть лучше всего организованы посредством описания всех функций для поддержания генерации откликов. Например, функции системы учета персонала могут быть организованы в разделы, соответствующие всем функциям, связанным с составлением чеков по оплате, всем функциям, связанным с составлением текущего списка служащих и т.д. В этом случае следует использовать шаблон, представленный в приложении А.6 (со всеми событиями стимулов, замененных откликом).

### **5.3.7.7 Функциональная иерархия**

Если ни одна из вышеупомянутых организационных схем оказывается непригодной, полные функциональные возможности могут быть организованы в иерархию функций, организованных или по общим входным воздействиям, или по общим выходным данным, или по общему доступу к внутренним данным. Чтобы показать связи между функциями и данными, можно использовать схемы потоков данных и словари данных. При организации этого раздела по функциональной иерархии следует использовать шаблон, представленный в приложении А.7.

### **5.3.8 Дополнительные комментарии**

При рассмотрении новой SRS подходящими может оказаться более одного из методов организации, приведенных в пункте 5.3.7.7. В таких случаях можно организовать специфические требования для нескольких иерархий, приспособленные к конкретным потребностям специфицируемой системы. Например, способ организации, объединяющий класс пользователей и свойства, показан в приложении А.8. Любые дополнительные требования могут быть включены в отдельный раздел в конце SRS.

Существует множество систем обозначений, методов и автоматизированных средств поддержки, доступных для обеспечения помощи в документировании требований. По большей части, их полезность является функцией способа организации. Например, при организации требований по

режимам могут оказаться полезными конечные автоматы или диаграммы состояний; при организации по объектам - метод объектно-ориентированного анализа; при организации по свойствам - последовательности стимул-отклик; а при организации по функциональной иерархии - схемы потоков данных и словари данных.

В любой из схем, представленных в Приложениях с A.1 по A.8, разделы, озаглавленные "Функциональное требование", могут быть описаны на оригинальном языке (например, английском), в псевдокоде, на языке определений системы или в четырех подразделах, озаглавленных: Введение, Входные данные, Обработка и Выходные данные.

## **5.4 Вспомогательная информация**

Вспомогательная информация делает SRS более легкой для использования. Она включает следующие пункты:

- а) Содержание;
- б) Алфавитный указатель;
- в) Приложения.

### **5.4.1 Содержание и алфавитный указатель**

Содержание и алфавитный указатель являются весьма важными пунктами и должны подчиняться общим методикам их составления.

### **5.4.2 Приложения**

Приложения не всегда рассматриваются как часть фактической SRS и не всегда необходимы. Они могут включать:

- а) Типовые форматы ввода/вывода, описания исследования калькуляции себестоимости или результаты пользовательских обзоров;
- б) Дополнительную или предварительную информацию, которая может помочь читателям SRS;
- в) Описание проблем, которые должны решаться программным обеспечением;
- г) Специальные команды для кодов и носителей, обеспечивающие соответствие требованиям защиты, экспорта, начальной загрузки или другим.

При включении приложений в SRS необходимо в явном виде сформулировать, должны ли эти приложения считаться частью требований.

о

## Приложение А

(информационное)

### Шаблоны SRS

#### А.1 Шаблон раздела 3 SRS, организованного по режимам: Версия 1

##### 3. Специфические требования

##### 3.1 Требования к внешним интерфейсам

- 3.1.1 Интерфейсы пользователя
- 3.1.2 Аппаратные интерфейсы
- 3.1.3 Интерфейсы программного обеспечения
- 3.1.4 Интерфейсы связи

##### 3.2 Функциональные требования

- 3.2.1 Режим 1
  - 3.2.1.1 Функциональное требование 1.1
  - 3.2.1.*n* Функциональное требование 1..*n*
- 3.2.2 Режим 2
- 3.2..*m* Режим *m*
  - 3.2.7*m*.1 Функциональное требование от *m*..1

3.2..*m*.*n* Функциональное требование *m*..*n*

- 3.3 Требования к рабочим характеристикам
- 3.4 Проектные ограничения
- 3.5 Атрибуты системы программного обеспечения
- 3.6 Другие требования

#### А.2 Шаблон раздела 3 SRS, организованного по режимам: Версия 2

##### 3. Специфические требования 3.1

##### Функциональные требования 3.1.1

##### Режим 1

- 3.1.1.1 Внешние интерфейсы
  - 3.1.1.1.1 Интерфейсы пользователя
  - 3.1.1.1.2 Аппаратные интерфейсы
  - 3.1.1.1.3 Интерфейсы программного обеспечения
  - 3.1.1.1.4 Интерфейсы связи
- 3.1.1.2 Функциональные требования
  - 3.1.1.2.1 Функциональное требование 1

- 3.1.1.2.*n* Функциональное требование *n*
- 3.1.1.3 Рабочие характеристики
- 3.1.2 Режим 2

- 3.1 *m* Режим *m*
- 3.2 Проектные ограничения
- 3.3 Атрибуты системы программного обеспечения
- 3.4 Другие требования

### **A.3 Шаблон раздела 3 SRS, организованного по классам пользователей**

- 3. Специфические требования
- 3.1 Требования к внешним интерфейсам
  - 3.1.1 Интерфейсы пользователя
  - 3.1.2 Аппаратные интерфейсы
  - 3.1.3 Интерфейсы программного обеспечения
  - 3.1.4 Интерфейсы связи
- 3.2 Функциональные требования
  - 3.2.1 Класс пользователей 1
    - 3.2.1.1 Функциональное требование 1.1
    - 3.2.1.*n* Функциональное требование 1.*n*
  - 3.2.2 Класс пользователей 2
    - 3.2.2 *m* Класс пользователей *m*
      - 3.2.*m*.1 Функциональное требование *m*.1
      - 3.2.*m*.*n* Функциональное требование *m*.*n*
- 3.3 Требования к рабочим характеристикам
- 3.4 Проектные ограничения
- 3.5 Атрибуты системы программного обеспечения
- 3.6 Другие требования

### **A.4 Шаблон раздела 3 SRS, организованного по объектам**

- 3. Специфические требования
- 3.1 Требования к внешним интерфейсам
  - 3.1.1 Интерфейсы пользователя
  - 3.1.2 Аппаратные интерфейсы ..
  - 3.1.3 Интерфейсы программного обеспечения
  - 3.1.4 Интерфейсы связи
- 3.2 Классы/Объекты
  - 3.2.1 Класс/Объект1

3.2.1.1 Атрибуты (прямые или унаследованные)

3.2.1.1.1 Атрибут 1

3.2.1.1.*n* Атрибут *n*

3.2.1.2 Функции (услуги, методы, прямые или унаследованные) 3.2.1.2.1

Функциональное требование 1.1

3.2.1.2.*m* Функциональное требование 1.*m*

3.2.1.3 Сообщения (полученные или отправленные)

3.2.2 Класс/Объект 2

3.2..*p* Класс/Объект *p*

3.3 Требования к рабочим характеристикам

3.4 Проектные ограничения

3.5 Атрибуты системы программного обеспечения

3.6 Другие требования

**A.5 Шаблон раздела 3 SRS, организованного по свойствам**

3. Специфические требования

3.1 Требования к внешним интерфейсам

3.1.1 Интерфейсы пользователя

3.1.2 Аппаратные интерфейсы

3.1.3 Интерфейсы программного обеспечения

3.1.4 Интерфейсы связи

3.2 Свойства системы

3.2.1 Свойство системы 1

3.2.1.1 Введение/Назначение свойства

3.2.1.2 Последовательность стимулов/откликов

3.2.1.3 Ассоциированные функциональные требования

3.2.1.3.1 Функциональное требование 1

3.2.1.3.*n* Функциональное требование *n*

3.2.2 Свойство системы 2

3.2.*m* Свойство системы *m*

3.3 Требования к рабочим характеристикам

3.4 Проектные ограничения

3.5 Атрибуты системы программного обеспечения

3.6 Другие требования



- 3.2.1.*n*.1 Информационные объекты
- 3.2.1.*n*.2 Релевантные потоки
- 3.2.1.*n*.3 Топология
- 3.2.2 Описания процессов
  - 3.2.2.1 Процесс 1
    - 3.2.2.1.1 Объекты входных данных
    - 3.2.2.1.2 Алгоритм или формула процесса
    - 3.2.2.1.3 Объекты обрабатываемых данных
  - 3.2.2.2 Процесс 2
    - 3.2.2.2.1 Объекты входных данных
    - 3.2.2.2.2 Алгоритм или формула процесса
    - 3.2.2.2.3 Объекты обрабатываемых данных
  - 3.2.2.*m* Процесс *m*
    - 3.2.2.*m*.1 Объекты входных данных *m*.1,
    - 3.2.2.*m*.2 Алгоритм или формула процесса
    - 3.2.2.*m*.3 Объекты обрабатываемых данных
- 3.2.3 Спецификации структуры данных
  - 3.2.3.1 Структура 1
    - 3.2.3.1.1 Тип записи
    - 3.2.3.1.2 Составляющие поля
  - 3.2.3.2 Структура 2
    - 3.2.3.2.1 Тип записи
    - 3.2.3.2.2 Составляющие поля
  - 3.2.3.*p* Структура *p*
    - 3.2.3.*p*.1 Тип записи
    - 3.2.3.*p*.2 Составляющие поля
- 3.2.4 Словарь данных
  - 3.2.4.1 Элемент данных 1
    - 3.2.4.1.1 Имя
    - 3.2.4.1.2 Представление
    - 3.2.4.1.3 Единицы/Формат
    - 3.2.4.1.4 Разрядность/Точность
    - 3.2.4.1.5 Диапазон
  - 3.2.4.2 Элемент данных 2
    - 3.2.4.2.1 Имя
    - 3.2.4.2.2 Представление
    - 3.2.4.2.3 Единицы/Формат
    - 3.2.4.2.4 Разрядность/Точность
    - 3.2.4.2.5 Диапазон
  - 3.2.4.*q* Элемент данных *q*
    - 3.2.4.*q*.1 Имя
    - 3.2.4.*q*.2 Представление
    - 3.2.4.*q*.3 Единицы/Формат
    - 3.2.4.*q*.4 Разрядность/Точность
    - 3.2.4.*q*.5 Диапазон

- 3.3 Требования к рабочим характеристикам
- 3.4 Проектные ограничения
- 3.5 Атрибуты системы программного обеспечения
- 3.6 Другие требования

## **A.8 Шаблон раздела 3 SRS, показывающий множественную организацию**

### **3. Специфические требования**

#### **3.1 Требования к внешним интерфейсам**

- 3.1.1 Интерфейсы пользователя
- 3.1.2 Аппаратные интерфейсы
- 3.1.3 Интерфейсы программного обеспечения
- 3.1.4 Интерфейсы связи

#### **3.2 Функциональные требования**

- 3.2.1 Класс пользователей 1
  - 3.2.1.1 Свойство 1.1
    - 3.2.1.1.1 Введение/Назначение свойства
    - 3.2.1.1.2 Последовательность стимулов/откликов
    - 3.2.1.1.3 Ассоциированные функциональные требования
  - 3.2.1.2 Свойство 1.2
    - 3.2.1.2.1 Введение/Назначение свойства
    - 3.2.1.2.2 Последовательность стимулов/откликов
    - 3.2.1.2.3 Ассоциированные функциональные требования

#### **3.2.1.*m* Свойство 1..*m***

- 3.2.1. *m*..1 Введение/Назначение свойства
- 3.2.1. *m* ..2 Последовательность стимулов/откликов
- 3.2.1. *m*..3 Ассоциированные функциональные требования

#### **3.2.2 Класс пользователей 2**

#### **3.2.*n* Класс пользователей *n***

- 3.3 Требования к рабочим характеристикам
- 3.4 Проектные ограничения
- 3.5 Атрибуты системы программного обеспечения
- 3.6 Другие требования