Проектирование ОС и компонентов

Преп. Душутина Е.В.

Выборка из стандарта IEEE Std 830-1998 (Ревизия IEEE Std 830-1993) для выполнения **практического задания (часть2) для заданной ранее ОС**

**SRS -**

законченное описание поведения программы, которую требуется разработать

[SRS — Software Requirement Specification](http://seal.ifi.uzh.ch/fileadmin/User_Filemount/Vorlesungs_Folien/SOPRA/IEEE-SRS-Standard.pdf) — специальная документация для ПО которая содержит в себе информацию о том, как должна себя вести система, какие функции должна выполнять, какую нагрузку должна выдерживать и тд.

В стандарте IEEE830 содержатся рекомендации к структуре и методах описания программных требований — «Recommended Practice for Software Requirements Specifications»

Выборка из стандарта IEEE Std 830-1998 (Ревизия IEEE Std 830-1993) для выполнения практического задания (часть2) для заданной ранее ОС

**5. Части SRS**

В этом разделе обсуждается каждая основная часть SRS. Эти части упорядочены ниже в виде плана, который может использоваться как образец при написании SRS. Хотя SRS не обязана в точности следовать этому плану или использовать такие же заголовки частей, хорошая SRS должна включать в себя всю приведенную информацию.

Содержание SRS:

1. Введение
   1. Назначение
   2. Область применения
   3. Определения, акронимы и сокращения
   4. Обзор
2. Общее описание
   1. Позиционирование продукта
   2. Функции продукта
   3. Пользовательские характеристики
   4. Ограничения
   5. Предположения и зависимости
3. Специфические требования (см. п.п. с [5.3.1](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#5_3_1) по [5.3.8](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#5_3_8) с пояснением возможных специфических требований. См. также в [Приложении А](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#A) различные способы организации этого раздела SRS)
4. Приложения
5. Индекс

**5.1. Введение (Раздел 1 SRS)**

Введение должно представлять обзор SRS В целом. Оно должно включать следующие подразделы:

1. Назначение.
2. Область применения.
3. Определения, акронимы и сокращения.
4. Ссылки.
5. Обзор.

**5.1.1. Назначение (1.1 SRS)**

В этом подразделе следует:

1. Определить назначение SRS;
2. Задать целевую аудиторию SRS.

**5.1.2. Область применения (1.2 SRS)**

Этот подраздел должен:

1. Идентифицировать производимый продукт по имени (например, Host DBMS, Report Generator и т.д.);
2. Пояснять, что должен делать программный продукт, а также, при необходимости, чего он не должен делать;
3. Описать применение программного обеспечения, включая выгоды, намерения и цели;
4. Согласовываться со сходными положениями спецификаций верхнего уровня (например, спецификацией требований к системе), если они существуют.

**5.1.3. Определения, акронимы и сокращения (1.3 SRS)**

Этот подраздел должен представлять определения всех терминов, акронимов и сокращений, необходимых для правильной интерпретации SRS. Эта информация может быть представлена в виде ссылок на одно или более приложений к SRS либо на другие документы.

**5.1.4. Ссылки (1.4 SRS)**

Данный подраздел должен:

1. Представлять полный перечень документов, на которые есть ссылки где-либо в SRS;
2. Идентифицировать каждый документ по названию, отчетному номеру (если применимо), дате и опубликовавшей организации;
3. Задавать источники, из которых могут быть получены документы, на которые имеются ссылки.

Эта информация может быть представлена в виде ссылки на приложение или другой документ.

**5.1.5. Обзор (1.5 SRS)**

Данный подраздел должен:

1. Описывать содержимое остальной части SRS;
2. Пояснять организацию SRS.

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

Этот раздел SRS должен описывать общие факторы, оказывающие влияние на продукт и требования к нему. В этом разделе не приводятся специфические требования. В нем подготавливается основа для требований, которые детально определяются в Разделе 3 SRS, и приводится информация, облегчающая их понимание.

Этот раздел обычно состоит из шести подразделов:

1. Позиционирование продукта;
2. Функции продукта;
3. Характеристики пользоватеолей;
4. Ограничения;
5. Предположения и зависимости;
6. Распределение требований.

**5.2.1. Позиционирование продукта (2.1 SRS)**

Этот подраздел позиционирует продукт среди других связанных продуктов. Если продукт является независимым и полностью самодостаточным, это следует отразить здесь. Если SRS определяет продукт, который является компонентом большей системы, как это часто бывает, то данный подраздел должен связать требования к большей системе с функциональностью программного обеспечения и определить интерфейсы между системой и программным обеспечением.

Могут быть полезными блок-схемы, показывающие основные компоненты большей системы, связи между ними и внешние интерфейсы.

Этот подраздел должен также описывать, как программное обеспечение работает под действием различных ограничений. Например, эти ограничения могут включать:

1. Системные интерфейсы;
2. Пользовательские интерфейсы;
3. Аппаратные интерфейсы;
4. Программные интерфейсы;
5. Коммуникационные интерфейсы;
6. Память;
7. Операции;
8. Требования к адаптируемости на месте.

**5.2.1.1. Системные интерфейсы**

Следует перечислить все системные интерфейсы и идентифицировать функциональность программного обеспечения для выполнения требований к системе, а также описание интерфейса для соответствия системе.

**5.2.1.2. Пользовательские интерфейсы**

Следует задать:

1. Логические характеристики каждого интерфейса между программным продуктом и его пользователями. Сюда входят конфигурационные характеристики (например, требуемые экранные формы, страницы или раскладки окон, содержимое всех отчетов или меню, а также доступность программируемых функциональных клавиш), необходимые для выполнения программных требований.
2. Все аспекты оптимизации интерфейса с персоналом, который должен использовать систему. Они могут просто включать список, как система должна выглядеть с точки зрения пользователя, а как не должна. Например, может быть требование наличия опции полных или кратких сообщений об ошибках. Подобно всем остальным, эти требования должны быть верифицируемыми, например: «оператор 4-го разряда должен научиться выполнять действие X за Z минут после часового обучения», а не «оператор должен выполнять действие X».

(Это также можно задать в «Системных атрибутах программного обеспечения» в разделе под названием «Простота использования»).

**5.2.1.3. Аппаратные интерфейсы**

Здесь следует задать логические характеристики каждого интерфейса между программным продуктом и аппаратными компонентами системы. Сюда входят конфигурационные характеристики (число портов, набор инструкций и т.д.). Также освещаются такие моменты, как: какие устройства поддерживаются, как они поддерживаются, а также протоколы. Например, поддержка терминала может задавать поддержку полноэкранного интерфейса, в противоположность построчному вводу.

**5.2.1.4. Программные интерфейсы**

Следует задать использование прочих необходимых программных продуктов (например, система управления данными, операционная система или математический пакет), а также интерфейсы с другими прикладными системами (например, связь между системой приема оплаты счетов и общей бухгалтерской системой). Для каждого необходимого программного продукта следует предоставить следующую информацию:

1. Название;
2. Мнемоника;
3. Номер спецификации;
4. Номер версии;
5. Источник.

Для каждого интерфейса следует предоставить следующую информацию:

1. Обсуждение назначения интерфейсного программного обеспечения с точки зрения программного продукта.
2. Определение интерфейса в терминах содержания и формата сообщений. Не обязательно детально описывать хорошо документированные интерфейсы, но требуется ссылка на документ, определяющий требуемый интерфейс.

**5.2.1.5. Коммуникационные интерфейсы**

Следует задать различные интерфейсы коммуникаций: протоколы локальных сетей и т.д.

**5.2.1.6. Ограничения по памяти**

Следует задать все значимые характеристики и ограничения, касающиеся первичной и вторичной памяти.

**5.2.1.7. Операции**

Следует задать обычные и специфические операции, которые требуются пользователю, например:

1. Различные модели операций в организации пользователя (например, операции, инициируемые пользователем);
2. Периоды интерактивных операций и периоды операций, не требующих ручного вмешательства;
3. Функции поддержки обработки данных;
4. Операции резервного копирования и восстановления.

*Примечание.* Иногда они задаются как часть раздела «Пользовательские Интерфейсы».

**5.2.1.8. Требования к адаптации на месте**

Следует:

1. Определить требования ко всем данным или последовательностям инициализации, специфичным для данного места, задачи или режима работы (например, таблицы значений, безопасные пределы и т.д.);
2. Задать особенности, относящиеся к месту или задаче, которые следует модифицировать с целью адаптации программного обеспечения к конкретной инсталляции.

**5.2.2. Функции продукта (2.2 SRS)**

В этом подразделе SRS следует представить сводку основных функций, выполняемых системой. Например, SRS для бухгалтерской программы может посвятить эту часть работе со счетами, обслуживанию клиентов и подготовке платежных поручений, не вдаваясь в обширную детализацию этих функций.

Иногда сводка функций, необходимых для данной части, берется прямо из соответствующего раздела спецификации верхнего уровня (если она есть), которая размещает некоторые функции в программном продукте. Заметим для ясности, что:

1. Функции должны быть организованы таким образом, чтобы сделать перечень функций понятным потребителю или другим читателям при первом прочтении документа.
2. Можно использовать текстовые или графические методики для представления различных функций и отношений между ними. Подобные диаграммы не должны представлять реализацию продукта, а лишь показывать логические взаимосвязи между переменными.

**5.2.3. Характеристики пользователей (2.3 SRS)**

Этот подраздел SRS должен описывать общие характеристики предполагаемых пользователей, включая уровень образования, опыт и техническую грамотность. В нем не следует устанавливать специфические требования, но следует привести причины, по которым некоторые специфические требования заданы далее в [Разделе 3](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#3) SRS.

**5.2.4. Ограничения (2.4 SRS)**

В этом подразделе должны быть приведены общие описания всего того, что может ограничить действия разработчика. Они включают:

1. Правовые вопросы;
2. Аппаратные ограничения (например, требования к длительности сигналов);
3. Интерфейсы с другими приложениями;
4. Параллельные операции;
5. Функции аудита;
6. Функции управления;
7. Языковые ограничения высшего порядка;
8. Протоколы синхронизации сигналов (например, XON-XOFF, ACK-NACK);
9. Требования к надежности;
10. Критичность приложения;
11. Соображения безопасности и секретности.

**5.2.5. Предположения и зависимости (2.5 SRS)**

В этом подразделе следует перечислить все факторы, которые влияют на требования, устанавливаемые SRS. Эти факторы не являются проектными ограничениями, а скорее относятся к их изменениям, которые могут повлиять на требования SRS. Например, может быть сделано предположение, что некоторая операционная система будет доступна для оборудования, на которое ориентирован программный продукт. Если в действительности операционная система недоступна, потребуется соответствующее изменение SRS.

**5.2.6. Распределение требований (2.6 SRS)**

Этот подраздел должен представлять требования, которые могут быть отложены до будущих версий системы.

**5.3. Специфические требования (Раздел 3 SRS)**

Этот раздел SRS должен содержать все требования к программному обеспечению на уровне детализации, достаточном, чтобы позволить разработчикам создать систему, удовлетворяющую этим ограничениям, а тестерам — проверить, удовлетворяет ли система этим ограничениям. В данном разделе каждое требование должно быть ориентировано на пользователей, операторов или другие системы, внешние по отношению к данной. Эти требования должны включать минимальное описание для каждого ввода в систему, каждого ответа системы, а также всех функций, выполняемых системой в ответ на ввод или для поддержки вывода. Поскольку зачастую эта часть SRS является самой большой и наиболее важной, применяются следующие принципы:

1. Специфические требования должны соответствовать характеристикам, описанным в [подразделе 4.3](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#4_3).
2. Специфические требования должны снабжаться перекрестными ссылками на ранние документы.
3. Все требования должны быть уникально идентифицируемыми.
4. Следует уделить особое внимание организации требований с целью достижения максимальной читабельности.

Перед изучением способов организации требований полезно рассмотреть различные элементы, включающие в себя требования, которые описаны в п.п. с [5.3.1](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#5_3_1) по [5.3.7](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#5_3_7).

**5.3.1.Внешние интерфейсы**

Здесь должно быть детальное описание всех входных и выходных данных программной системы. Оно должно дополнять описания интерфейсов из [подраздела 5.2](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#5_2) и не содержать повторяющейся информации. Оно должно иметь формат и содержание, как указано ниже:

1. Имя элемента;
2. Описание назначения;
3. Источник входных данных или получатель выходных;
4. Допустимый диапазон, точность и отклонения;
5. Единицы измерения;
6. Временные диаграммы;
7. Взаимосвязи с другими входами/выходами;
8. Форматы и организация экранов;
9. Форматы и организация окон;
10. Форматы данных;
11. Форматы команд;
12. Завершающие сообщения.

**5.3.2. Функции**

Функциональные требования должны определять фундаментальные действия, которые должны выполняться программным обеспечением для приема и обработки входных данных, а также обработки и вывода выходных данных. Они обычно перечисляются в виде предложений, начинающихся со слов «Система должна…».

Они включают:

1. Проверку допустимости входных значений;
2. Точный порядок действий;
3. Реакцию на нештатные ситуации, включающие:
   1. Переполнение;
   2. Коммуникационные проблемы;
   3. Обработку ошибок и восстановление;
4. Влияние параметров;
5. Взаимосвязь между входными и выходными данными, включая:
   1. Порядок ввода/вывода;
   2. Формулы преобразования входных данных в выходные

Может оказаться удобным разделение функциональных требований на подфункции или подпроцессы. Это не значит, что программный проект будет разделен таким же образом.

**5.3.3. Требования к производительности**

Этот подраздел должен задавать как статические, так и динамические численные требования, предъявляемые в целом к программному обеспечению или к взаимодействию человека с программой. Статические численные требования могут включать следующие:

1. Число поддерживаемых терминалов;
2. Число одновременно поддерживаемых пользователей;
3. Объем и тип обрабатываемой информации.

Статические численные требования иногда оформляются в виде отдельного раздела.

Динамические численные требования могут включать, например, число транзакций и задач, или объем данных, обрабатываемых в некоторый период данных в условиях как нормальной, так и пиковой нагрузки.

Все эти требования следует формулировать в терминах измеримых величин.

Например:

95% транзакций должны обрабатываться менее чем за 1 секунду

вместо:

Оператор не должен ждать, пока завершится транзакция.

**Примечание**. Численные границы, применимые к отдельной функции, обычно задаются в описании данной функции, в подразделе обработки.

**5.3.4. Логические требования к базе данных**

Здесь следует задать логические требования к информации, которая должна размещаться в базе данных. Они могут включать следующие:

1. Типы информации, используемой различными функциями;
2. Частоту использования;
3. Возможность доступа;
4. Сущности и отношения между ними;
5. Ограничения целостности;
6. Требования к хранению данных.

**5.3.5. Ограничения проектирования**

Здесь задаются ограничения проектирования, налагаемые другими стандартами, аппаратурой и т.д.

**5.3.5.1. Соответствие стандартам**

Этот подраздел должен задавать ограничения, вытекающие из существующих стандартов и правил. Они могут включать следующее:

1. Форматы отчетов;
2. Именование данных;
3. Бухгалтерские процедуры;
4. Протоколирование работы.

Например, требования к программному обеспечению по трассировке вычислительных действий. Подобные трассировки необходимы для некоторых приложений для выполнения требований правовых или финансовых стандартов. Требование протоколирования работы может, например, устанавливать, что при изменениях в платежной ведомости прежнее и новое значения должны записываться в файл трассировки.

**5.3.6. Атрибуты программной системы**

Некоторые атрибуты программного обеспечения могут служить требованиями. Важно, чтобы требуемые атрибуты были заданы таким образом, чтобы можно было объективно проверить выполнение данного требования. В п.п. с [5.3.6.1](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#5_3_6_1) по [5.3.6.5](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#5_3_6_5) приведен список примеров.

**5.3.6.1. Надежность**

Следует перечислить факторы, необходимые для установления требуемого уровня надежности программного обеспечения.

**5.3.6.2. Доступность**

Следует перечислить факторы, призванные гарантировать определенный уровень доступности системы, такие как контрольные точки, восстановление и перезапуск.

**5.3.6.3. Безопасность**

Следует перечислить факторы, защищающие программное обеспечение от случайного или злонамеренного доступа, использования, модификации, разрушения или разглашения. Специфические требования в этой области включают необходимость:

1. Использования криптографии;
2. Хранение логов или истории;
3. Назначать некоторые функции различным модулям;
4. Ограничивать коммуникации между некоторыми областями программы;
5. Проверять целостность данных для критических переменных.

**5.3.6.4. Поддерживаемость**

Следует перечислить атрибуты программного обеспечения, относящиеся к легкости поддержки самого программного обеспечения. Это могут быть некоторые требования, относящиеся к модульности, интерфейсам, сложности и т.д. Не следует помещать здесь требования лишь потому, что принято считать их хорошим тоном разработки.

**5.3.6.5. Переносимость**

Следует перечислить атрибуты программного обеспечения, касающиеся легкости переноса программного обеспечения на другие компьютеры и/или операционные системы. Они могут включать следующее:

1. Доля компонентов с машинно-зависимым кодом;
2. Доля машинно-зависимого кода;
3. Использование испытанного переносимого языка;
4. Использование особенного компилятора или подмножества языка;
5. Использование особенной операционной системы.

**5.3.7. Организация специфических требований**

Для любой нетривиальной системы детальные требования обширно разрастаются. Поэтому рекомендуется тщательный подход к их организации в виде, оптимальном для понимания. Не существует единой организации, оптимальной для всех систем. Различные классы систем придерживаются различной организации требований в Разделе 3 SRS. Некоторые способы организации описаны в п.п. с [5.3.7.1](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#5_3_7_1) по [5.3.7.7](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#5_3_7_7).

**5.3.7.1. Режим системы**

Поведение некоторых систем кардинально различается в зависимости от режима системы. Например, система управления может иметь различные наборы функций для разных режимов: обучение, нормальный или аварийный. При организации этого раздела по режимам, можно использовать планы [А.1](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#A_1) или [А.2](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#A_2). Выбор определяется тем, зависят ли от режима интерфейсы и производительность.

**5.3.7.2. Класс пользователя**

Некоторые системы обеспечивают различные наборы функций для разных классов пользователей. Например, система управления лифтом предоставляет различные возможности пассажирам, обслуживающему персоналу и пожарным. При организации этого раздела по классам пользователей следует использовать план [А.3](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#A_3).

**5.3.7.3. Объекты**

Объекты — это сущности реального мира, которые имеют соответствие в системе. Например, в системе мониторинга пациентов в число объектов входят пациенты, датчики, медсестры, кабинеты, врачи, медикаменты и т.д. С каждым объектом связаны наборы атрибутов (данного объекта) и функций (выполняемых данным объектом). Эти функции называются также сервисами, методами или процессами. При организации этого раздела по объектам следует использовать план [А.4](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#A_4). Заметим, что наборы объектов могут иметь общие атрибуты и сервисы. Они группируются вместе как классы.

**5.3.7.4. Функциональные возможности**

Функциональная возможность — это внешний сервис системы, который может потребовать последовательность ввода данных для достижения желаемого результата. Например, в телефонной системе функциональные возможности включают местный вызов, переадресацию звонка и конференцию. Обычно функциональные возможности описывают в виде последовательности пар запрос-ответ. При организации этого раздела в соответствии с функциональными возможностями следует использовать план [А.5](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#A_5).

**5.3.7.5. Внешние воздействия**

Некоторые системы могут быть лучше организованы путем описания их функций в терминах внешних воздействий. Например, функции системы автоматического приземления самолета могут быть разбиты на секции для потери тяги, порывов ветра, неожиданной смены курса, избыточной вертикальной скорости и т.д. При организации этого раздела в соответствии с внешними воздействиями следует использовать план [А.6](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#A_6).

**5.3.7.6. Отклики**

Некоторые системы могут быть лучше организованы путем описания всех функций, поддерживающих генерацию откликов. Например, функции системы управления персоналом могут быть разбиты на секции, соответствующие всем функциям, связанным с генерацией чеков оплаты, всем функциям, связанным с генерацией текущего списка сотрудников, и т.д. Следует использовать план A.6, в котором все внешние воздействия заменены откликами.

**5.3.7.7. Функциональная иерархия**

Если ни одна из вышеперечисленных организационных схем не подходит, общая функциональность может быть организована в виде иерархии функций, организованных в соответствии с входными данными, выходными данными или внутренними данными. Могут быть использованы диаграммы потоков данных и словари данных, чтобы показать взаимосвязи между функциями и данными. При организации этого раздела в соответствии с функциональной иерархией следует использовать план [А.7](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#A_7).

**5.3.8. Дополнительные комментарии**

В процессе работы над SRS могут оказаться применимыми более одной организационной схемы из перечисленных в п.п. [5.3.7.7](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#5_3_7_7). В таких случаях организуйте специфические требования по нескольким иерархиям, приспособленным под специфические нужды системы, для которой разрабатываются спецификации. Например, см. [А.8](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#A_8) как пример организации, комбинирующей классы пользователей и функциональные возможности. Все дополнительные требования могут быть размещены в отдельном разделе в конце SRS.

Имеется множество нотаций, методик и автоматизированных инструментов для поддержки документирования требований. В основном они полезны в части организационной функции. Например, при организации по режимам могут оказаться полезными конечные автоматы диаграммы состояний; при организации по объектам может оказаться полезным объектно-ориентированный анализ; при организации по функциональным возможностям могут оказаться полезными последовательности запрос-ответ, а при организации по функциональной иерархии могут пригодиться диаграммы потоков данных и словари данных.

В любом из планов, описанных с [А.1](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#A_1) по [А.8](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md#A_8), разделы с именем «Функциональное требование i» могут быть описаны на естественном языке, на псевдокоде, на языке описания систем либо в виде четырех подразделов с именами «Введение», «Входные данные», «Обработка» и «Выходные данные».

**5.4. Вспомогательная информация**

Вспомогательная информация делает SRS более удобочитаемой. Она включает следующее:

1. Оглавление;
2. Индекс;
3. Приложения.

**5.4.1. Оглавление и индекс**

Оглавление и индекс крайне важны и должны следовать общим принципам композиции.

**5.4.2. Приложения**

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

* Примеры форматов входных и выходных данных, описания примеров ценового анализа или результаты пользовательских обзоров;
* Вспомогательная и дополнительная информация, которая может помочь читателю SRS;
* Описание проблемы, решаемой программным обеспечением;
* Специальные инструкции по упаковке для кода и носителей для соответствия требованиям по безопасности, экспорту начальной загрузке и т.д.

Если приложения включены в состав SRS, следует явно указать, являются ли они частью требований.

**Приложения**

**A.5. Образец Раздела 3 SRS, организованный по Функциональным возможностям**

* 3. Специфические требования
  + 3.1. Внешние требования к интерфейсу
    - 3.1.1. Пользовательские интерфейсы
    - 3.1.2. Аппаратные интерфейсы
    - 3.1.3. Программные интерфейсы
    - 3.1.4. Коммуникационные интерфейсы
  + 3.2. Функциональные возможности системы
    - 3.2.1. Системная функция 1
      * 3.2.1.1. Введение (назначение функции)
      * 3.2.1.2. Последовательность «запрос/ответ»
      * 3.2.1.3. Связанные функциональные требования
        + 3.2.1.3.1. Функциональное требование 1
        + …
        + 3.2.1.3.n. Функциональное требование n
    - 3.2.2. Системная функция 2
    - …
    - 3.2.m. Системная функция m
  + 3.3. Требования к производительности
  + 3.4. Проектные ограничения
  + 3.5. Атрибуты программной системы
  + 3.6. Прочие требования