ЛЕКЦИЯ 12

Тема: IEEE Standard SRS Template (спецификации требования к программному обеспечению). Раздел специфические требования к программному продукту. Приложения . Индексы.

Цель: изучить спецификацию требований к ПО.

Содержание SRS:

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 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 подразделах с 3.1 по 3.7.

3.1.Внешние интерфейсы

Здесь должно быть детальное описание всех входных и выходных данных программной системы. Оно должно дополнять описания интерфейсов из 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.

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.

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.

3.4. Логические требования к базе данных

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

- Типы информации, используемой различными функциями;
- Частоту использования;
- Возможность доступа;
- Сущности и отношения между ними;
- Ограничения целостности;
- Требования к хранению данных.

3.5. Ограничения проектирования

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

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.

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 подразделах с 3.6.1 по 3.6.5 приведен список примеров.

3.6.1. Надежность

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

3.6.2. Доступность

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

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.

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.

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.

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 подразделах с 3.7.1 по 3.7.7.

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 A.1 или A.2. Выбор определяется тем, зависят ли от режима интерфейсы и производительность.

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 A.3.

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 A.4. Заметим, что наборы объектов могут иметь общие атрибуты и сервисы. Они группируются вместе как классы.

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 A.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 A.6.

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.

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 A.7.

3.8. Дополнительные комментарии

В процессе работы над SRS могут оказаться применимыми более одной организационной схемы из перечисленных в 3.7.7. В таких случаях организуйте специфические требования по нескольким иерархиям, приспособленным под специфические нужды системы, для которой разрабатываются спецификации. Например, см. 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 A.1 по 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».

4. ВСПОМОГАТЕЛЬНАЯ ИНФОРМАЦИЯ

Вспомогательная информация делает SRS более удобочитаемой. Она включает следующее:

- Оглавление;
- Индекс;
- Приложения.

4.1. Оглавление и индекс

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

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.

Контрольные вопросы:

1. Опишите из чего состоит общий раздел документации?
2. Опешите специфический раздел требований
3. Опешите раздел приложений
4. Опишите раздел индекса

Список использованных источников:

1. Государственный стандарт Российской Федерации. Информационная технология. Сопровождение программных средств
2. Основы маркетинга учебное пособие Суркова Е.В.
3. Рекомендуемая стандартом IEEE 830структура SRS