

Данный документ переведен В. Ведмедским.

Уточнение и устранение ошибок перевода С. Козлов

IEEE Std 830-1993. Рекомендации по разработке спецификаций требований программного обеспечения.

Одобрено 2 декабря, 1993 Комитетом Стандартов IEEE

Резюме: описано содержание и качество хорошей спецификации требований программного обеспечения (СТПО) и представлены несколько различных образцов схем СТПО. Эти рекомендации нацелены не только на определение требований к разрабатываемому программному обеспечению, но также могут применяться при выборе существующих внутренних и коммерческих изделий программного обеспечения.

Ключевые слова: контракт, клиент, прототипирование, спецификация требований программного обеспечения, поставщик, спецификация требований системы

Введение

(Введение не является частью «IEEE Std 830-1993 Рекомендации по разработке спецификаций требований программного обеспечения»)

Этот документ описывает рекомендуемые подходы к созданию спецификации требований программного обеспечения. Он основан на модели, в которой результатом процесса разработки спецификации требований программного обеспечения является однозначный и полный документ спецификации. Это должно помочь

- a) Клиентам программного обеспечения точно описать, что они желают получить
- b) Поставщикам программного обеспечения понять точно, чего хочет заказчик
- c) Индивидуумам выполнить следующие цели:
 - 1) Разработать стандартную схему спецификации требований программного обеспечения для их собственных организаций
 - 2) Определить формат и содержание собственных спецификаций требований программного обеспечения
 - 3) Включить дополнительные пункты требований, такие как спецификация требований контроля качества (quality checklist) или спецификация требований к написанию руководства (writer's handbook).

Хорошие спецификации должны обеспечить клиентам, поставщикам и другим индивидуумам несколько определенных преимуществ, таких как:

- a) Установить основание для соглашения между клиентами и поставщиками о том, что программное обеспечение должно делать. Полное описание функций, которые будут выполнены программным обеспечением, указанным в СТПО поможет потенциальному пользователю определить, выполняет ли программное обеспечение их потребности или как программное обеспечение должно быть изменено, чтобы выполнить их потребности.
- b) Уменьшить усилия на разработку. Подготовка СТПО вынуждает различные заинтересованные группы в организации клиента строго рассматривать все требования до начала разработки проекта и уменьшает в дальнейшем объем работ по перепроектированию, перекодированию, и

повторному тестированию. Внимательное рассмотрение требований в СТПО может выявить упущения, недоразумения и нелогичности на ранних стадиях разработки, когда эти проблемы более легко исправляются.

- c) Обеспечить основание для оценки затрат и календарного плана. Описание изделия, которое будет разработано так, как определено в СТПО является реалистическим основанием для оценки проектных затрат и может использоваться для согласования предложений или оценки стоимости.
- d) Обеспечить основу для испытаний (validation) и проверок. Организации могут развивать собственные планы испытаний и проверок намного более продуктивно при наличии хороших СТПО. Как часть контракта на разработку, СТПО обеспечивает основу, у которой соответствие может быть измерено.
- e) Облегчить передачу. СТПО обеспечивают более легкую передачу изделия программного обеспечения новым пользователям или установку на новые машины. Таким образом, заказчикам более легко передавать программное обеспечение другим частям их организации, а поставщикам более легко передавать программное обеспечение новым клиентам.
- f) Служить основанием для улучшения. Поскольку СТПО описывают изделие, но не проект, который развивается, то СТПО могут служить основой для дальнейшего улучшения готового изделия. СТПО могут быть изменены, но это обеспечивает фундамент для непрерывной оценки производства.

Оглавление

| | | |
|----------|--|-----------|
| 1 | КРАТКИЙ ОБЗОР (OVERVIEW) | 6 |
| 1.1 | Возможности (<i>SCOPE</i>) | 6 |
| 2 | ССЫЛКИ (REFERENCES) | 7 |
| 3 | ОПРЕДЕЛЕНИЯ (DEFINITIONS) | 8 |
| 3.1 | КОНТРАКТ (CONTRACT)..... | 8 |
| 3.2 | КЛИЕНТ (CUSTOMER)..... | 8 |
| 3.3 | ПОСТАВЩИК (SUPPLIER)..... | 8 |
| 3.4 | ПОЛЬЗОВАТЕЛЬ (USER)..... | 8 |
| 4 | СООБРАЖЕНИЯ ПО СОЗДАНИЮ ХОРОШИХ СТПО (CONSIDERATIONS FOR PRODUCING A GOOD SRS) | 9 |
| 4.1 | СУЩНОСТЬ СТПО (NATURE OF THE SRS) | 9 |
| 4.2 | КОНТЕКСТ СТПО (ENVIRONMENT OF THE SRS) | 9 |
| 4.3 | ХАРАКТЕРИСТИКИ ХОРОШИХ СТПО (CHARACTERISTICS OF A GOOD SRS)..... | 10 |
| 4.3.1 | <i>Правильный (Correct)</i> | 10 |
| 4.3.2 | <i>Непротиворечивый (Unambiguous)</i> | 11 |
| 4.3.2.1 | Сложности естественного языка (Natural language pitfalls)..... | 11 |
| 4.3.2.2 | Языки спецификаций требований (Requirements specification languages) 11 | |
| 4.3.2.3 | Инструменты представления (Representation tools)..... | 11 |
| 4.3.3 | <i>Полный (Complete)</i> | 12 |
| 4.3.3.1 | Использование фразы “будет определено” (Use of TBDs) | 12 |
| 4.3.4 | <i>Целостный (Consistent)</i> | 12 |
| 4.3.4.1 | Внутренняя целостность (Internal consistency) | 12 |
| 4.3.5 | <i>Оцениваемый по важности и/или стабильности (Ranked for importance and/or stability)</i> | 13 |
| 4.3.5.1 | Степень стабильности (Degree of stability) | 13 |
| 4.3.5.2 | Степень потребности (Degree of necessity)..... | 13 |
| 4.3.6 | <i>Проверяемость (Verifiable)</i> | 13 |
| 4.3.7 | <i>Изменяемость (Modifiable)</i> | 14 |
| 4.3.8 | <i>Трассируемость (Traceable)</i> | 14 |
| 4.4 | СОВМЕСТНАЯ ПОДГОТОВКА СТПО (JOINT PREPARATION OF THE SRS)..... | 15 |
| 4.5 | РАЗВИТИЕ СТПО (SRS EVOLUTION)..... | 15 |
| 4.6 | ПРОТОТИПИРОВАНИЕ (PROTOTYPING) | 16 |
| 4.7 | ВКЛЮЧЕНИЕ ПРОЕКТА В СТПО (EMBEDDING DESIGN IN THE SRS)..... | 16 |
| 4.7.1 | <i>Необходимые требования проекта (Necessary design requirements)</i> | 16 |
| 4.8 | ВКЛЮЧЕНИЕ ТРЕБОВАНИЙ ПРОЕКТИРОВАНИЯ В СТПО (EMBEDDING PROJECT REQUIREMENTS IN THE SRS) | 17 |
| 5 | РАЗДЕЛЫ СТПО (THE PARTS OF AN SRS) | 18 |
| 5.1 | ВВЕДЕНИЕ (INTRODUCTION) (РАЗДЕЛ 1 СТПО)..... | 18 |
| 5.1.1 | <i>Цели (Purpose) (1.1 из СТПО)</i> | 18 |
| 5.1.2 | <i>Границы применения (Scope) (1.2 из СТПО)</i> | 19 |
| 5.1.3 | <i>Термины, акронимы (аббревиатуры), сокращения (Definitions, acronyms, and abbreviations) (1.3 из СТПО)</i> | 19 |
| 5.1.4 | <i>Ссылки (References) (1.4 из СТПО)</i> | 19 |
| 5.1.5 | <i>Краткий обзор (Overview) (1.5 из СТПО)</i> | 19 |

| | | |
|----------|--|-----------|
| 5.2 | ОБЩЕЕ ОПИСАНИЕ (OVERALL DESCRIPTION) (РАЗДЕЛ 2 ИЗ СТПО) | 19 |
| 5.2.1 | Описание изделия (<i>Product perspective</i>) (2.1 из СТПО) | 20 |
| 5.2.1.1 | Интерфейсы системы (System interfaces) | 20 |
| 5.2.1.2 | Интерфейсы пользователя (User interfaces) | 20 |
| 5.2.1.3 | Интерфейсы аппаратных средств ЭВМ (Hardware Interfaces)..... | 21 |
| 5.2.1.4 | Интерфейсы программного обеспечения (Software Interfaces) | 21 |
| 5.2.1.5 | Интерфейсы коммуникаций (Communications interfaces)..... | 21 |
| 5.2.1.6 | Ограничения памяти (Memory constraints)..... | 21 |
| 5.2.1.7 | Действия (Operations) | 22 |
| 5.2.1.8 | Требования настройки рабочих мест (Site adaptation requirements) 22 | |
| 5.2.2 | Функции изделия (<i>Product functions</i>) (2.2 из СТПО) | 22 |
| 5.2.3 | Характеристики пользователей (<i>User characteristics</i>) (2.3 из СТПО) .. | 22 |
| 5.2.4 | Ограничения (<i>Constraints</i>) (2.4 из СТПО) | 23 |
| 5.2.5 | Предположения и зависимости (<i>Assumptions and dependencies</i>) (2.5 из СТПО) 23 | |
| 5.2.6 | Распределение требований (<i>Apportioning of requirements</i>) (2.6 из СТПО) 23 | |
| 5.3 | ДЕТАЛЬНЫЕ ТРЕБОВАНИЯ (SPECIFIC REQUIREMENTS) (СЕКЦИЯ 3 ИЗ СТПО)... | 23 |
| 5.3.1 | Внешние интерфейсы (<i>External interfaces</i>) | 24 |
| 5.3.2 | Функции(<i>Functions</i>)..... | 24 |
| 5.3.3 | Требования исполнения (<i>Performance requirements</i>)..... | 25 |
| 5.3.4 | Требования логики базы данных (<i>Logical database requirements</i>) | 25 |
| 5.3.5 | Ограничения проекта (<i>Design constraints</i>)..... | 26 |
| 5.3.5.1 | Соглашение о стандартах (Standards compliance)..... | 26 |
| 5.3.6 | Характеристики программного обеспечения системы (<i>Software system attributes</i>) 26 | |
| 5.3.6.1 | Надежность (Reliability) | 26 |
| 5.3.6.2 | Эксплуатационная готовность (Availability)..... | 26 |
| | Безопасность (Security) | 26 |
| 5.3.6.4 | Ремонтопригодность (Maintainability) | 27 |
| 5.3.6.5 | Переносимость (Portability) | 27 |
| 5.3.7 | Структурирование детальных требований (<i>Organizing the specific requirements</i>) 27 | |
| 5.3.7.1 | Режим системы (System mode) | 27 |
| 5.3.7.2 | Классы пользователей (User class) | 27 |
| 5.3.7.3 | Объекты (Objects) | 27 |
| 5.3.7.4 | Особенности (Feature) | 28 |
| 5.3.7.5 | Воздействие (Stimulus)..... | 28 |
| 5.3.7.6 | Реакция (Response) | 28 |
| 5.3.7.7 | Функциональные иерархии (Functional hierarchy)..... | 28 |
| 5.3.8 | Дополнительные комментарии (<i>Additional comments</i>) | 28 |
| 5.4 | ИНФОРМАЦИОННАЯ ПОДДЕРЖКА (SUPPORTING INFORMATION) | 29 |
| 5.4.1 | Оглавления и индексы (<i>Table of contents and index</i>)..... | 29 |
| 5.5 | ПРИЛОЖЕНИЯ (APPENDIXES) | 29 |
| 6 | ПРИЛОЖЕНИЕ А | 30 |
| 6.1 | ШАБЛОН РАЗДЕЛА 3 СТПО ОРГАНИЗОВАННЫЙ ПО РЕЖИМАМ: ВЕРСИЯ 1 | 30 |
| 6.2 | ШАБЛОН РАЗДЕЛА 3 СТПО ОРГАНИЗОВАННЫЙ ПО РЕЖИМАМ: ВЕРСИЯ 2 | 30 |
| 6.3 | ШАБЛОН РАЗДЕЛА 3 СТПО ОРГАНИЗОВАННЫЙ ПО КЛАССАМ ПОЛЬЗОВАТЕЛЕЙ 31 | |
| 6.4 | ШАБЛОН РАЗДЕЛА 3 СТПО ОРГАНИЗОВАННЫЙ ПО ОБЪЕКТАМ | 31 |
| 6.5 | ШАБЛОН РАЗДЕЛА 3 СТПО ОРГАНИЗОВАННЫЙ ПО ОСОБЕННОСТЯМ..... | 32 |

Внешние стандарты
IEEE Std 830-1993

| | | |
|-----|--|----|
| 6.6 | ШАБЛОН РАЗДЕЛА 3 СТПО ОРГАНИЗОВАННЫЙ ПО ВОЗДЕЙСТВИЯМ | 33 |
| 6.7 | ШАБЛОН РАЗДЕЛА 3 СТПО ОРГАНИЗОВАННЫЙ ПО ФУНКЦИОНАЛЬНОЙ ИЕРАРХИЕЙ | 33 |
| 6.8 | ШАБЛОН РАЗДЕЛА 3 СТПО ПОКАЗЫВАЮЩИЙ МНОЖЕСТВЕННУЮ ОРГАНИЗАЦИЮ..... | 35 |

1 Краткий обзор (Overview)

Стандарт описывает подходы к созданию спецификации требований программного обеспечения. Раздел имеет пять пунктов.

Пункт 1 объясняет возможности этого стандарта.

Пункт 2 содержит ссылки, сделанные к другим стандартам.

Пункт 3 содержит используемые определения.

Пункт 4 обеспечивает информационную основу для разработки СТПО.

Пункт 5 обсуждает каждую из существенных частей СТПО.

Стандарт также имеет приложения, в которых приводятся альтернативные шаблоны форматов СТПО.

1.1 Возможности (Scope)

Дает рекомендации по созданию спецификаций требований программного обеспечения. Описывает содержание и признаки хорошей спецификации требований программного обеспечения (СТПО) и представляет несколько образцов схем СТПО.

Эти рекомендации нацелены на определение требований создаваемого программного обеспечения. Может также применяться, чтобы помочь в выборе внутренних и коммерческих изделий программного обеспечения. Однако применение этих рекомендаций к уже созданному программному обеспечению может быть непродуктивно.

Когда программное обеспечение включается в какую-нибудь крупную систему, типа медицинского оборудования, могут возникнуть вопросы, выходящие за рамки настоящего стандарта.

Стандарт описывает процесс создания изделия и содержание изделия. Изделие - спецификация требований программного обеспечения. Стандарт может использоваться для создания спецификации требований программного обеспечения непосредственно или может использоваться как модель для определения более специального стандарта.

Стандарт не определяет никаких специальных методов, терминологии (nomenclature) или инструментов для подготовки СТПО.

2 Ссылки (References)

ASTM 1340-90, Standard Guide for Rapid Prototyping of Computerized Systems

IEEE Std 610.12- 1990, IEEE Standard Glossary of Software Engineering Terminology (ANSI).

IEEE Std 730-1989, IEEE Standard for Software Quality Assurance Plans (ANSI).

IEEE Std 828-1990, IEEE Standard for Software Configuration Management Plans (ANSI).

IEEE Std 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software (ANSI).

IEEE Std 982.2-1988, IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software (ANSI).

IEEE Std 983-1986, IEEE Guide to Software Quality Assurance Planning.

IEEE Std 1002-1987, IEEE Standard Taxonomy for Software Engineering Standards (ANSI).

IEEE Std 1012-1986, IEEE Standard for Software Verification and Validation Plans (ANSI).

IEEE Std 1016-1987, IEEE Recommended Practice for Software Design Descriptions (ANSI).

IEEE Std 1028- 1988, IEEE Standard for Software Reviews and Audits (ANSI).

IEEE Std 1042-1987, IEEE Guide to Software Configuration Management (ANSI).

IEEE Std 1058.1-1987, IEEE Standard for Project Management Plans (ANSI).

IEEE Std 1074-1991, IEEE Standard for Developing Software Life Cycle Processes (ANSI).

IEEE P1233, October 1993, Draft Guide to Developing System Requirements Specifications.

3 Определения (Definitions)

Вообще определения терминов, используемых в этом документе соответствуют определениям, сформулированным в IEEE Std 610.12-1990.5. Ниже даны определения ключевых терминов, поскольку они используются в этом документе.

3.1 Контракт (contract).

Юридически обязательный документ, согласованный клиентом и поставщиком. Он включает технические и организационные требования, стоимость и календарный план для изделия. Контракт может также содержать неофициальную, но полезную информацию типа обязательств или ожиданий вовлеченных сторон.

3.2 Клиент (customer).

Человек, или люди, которые оплачивают изделие и обычно (но не обязательно) определяет требования. В контексте этого документа клиент и поставщик могут быть членами одной организации.

3.3 Поставщик (supplier).

Человек, или люди, которые производят изделие для клиента. В контексте этого документа, могут быть членами одной организации.

3.4 Пользователь (user).

Человек, или люди, которые работают или взаимодействуют непосредственно с изделием. Пользователь (и) и клиент (ы) - часто не тот же самый человек (люди).

4 Соображения по созданию хороших СТПО (Considerations for producing a good SRS)

Этот пункт обеспечивает информационную основу для разработки СТПО. Включает следующее:

- a) Сущность СТПО
- b) Контекст СТПО
- c) Характеристики хороших СТПО
- d) Совместная подготовка СТПО
- e) Процесс изменения СТПО
- f) Прототипирование
- g) Включение проекта в СТПО
- h) Включение требований проектирования в СТПО

4.1 Сущность СТПО (Nature of the SRS)

СТПО - технические условия для отдельного программного продукта, программы, или набора программ, которые выполняют определенные задачи в определенной окружающей среде. СТПО могут быть созданы одним или несколькими представителями поставщика, одним или несколькими представителями клиента или обоими. Подпункт 4.4 рекомендует разработку при участии представителей поставщика и клиента.

Основные вопросы, которые должен рассматривать разработчик СТПО:

- a) Функциональные возможности. Что программа должна делать?
- b) Внешние интерфейсы. Как программное обеспечение взаимодействует с людьми, системным аппаратным обеспечением, другим оборудованием и другим программным обеспечением?
- c) Производительность. Какова скорость, готовность, время отклика, время восстановления различных функций программного обеспечения и т.д.?
- d) Характеристики (*Attributes*). Какова переносимость, корректность, ремонтпригодность, безопасность, и т.д.?
- e) Ограничения проекта, наложенные на реализацию. Требования любых существующих стандартов, язык выполнения, политика обеспечения целостности базы данных, ограничения на ресурсы, операционная среда (ы) исполнения и т.д.?

Разработчики СТПО должны избегать размещения в СТПО любых требований разработки или требований проекта.

Рекомендации по содержанию СТПО см. в разделе 5.

4.2 Контекст СТПО (Environment of the SRS)

Важно учитывать роль, которую СТПО играет в общем плане проекта, что определено стандартом IEEE Std 610.12-1990. Программное обеспечение может содержать по существу все функциональные возможности проекта, или может быть частью большей системы. В последнем случае обычно имеется СТПО, которая определяет интерфейсы между системой и частью программного обеспечения и содержит требования к характеристикам и функциональности данной части программного обеспечения. Конечно,

СТПО в этом случае должна быть согласована и расширена общесистемными требованиями.

IEEE Std 1074-1991 описывает этапы жизненного цикла программного обеспечения и соответствующие входы для каждого этапа. Другие стандарты, типа тех, которые включены в список в разделе 2, относятся к другим этапам жизненного цикла программного обеспечения и также могут дополнять требования программного обеспечения.

Так как СТПО играют особую роль в процессе развития программного обеспечения, то автор (ы) СТПО должны быть внимательными, чтобы не выходить за границы этой роли. Это означает, что СТПО

- a) Должен правильно определить все требования программного обеспечения. Требования программного обеспечения определяются сущностью задачи, которая будет решена, или специальной характеристикой проекта.
- b) Не должен описывать никакого проектирования или деталей выполнения. Они должны быть описаны в стадии разработки проекта.
- c) Не должны накладывать дополнительные ограничения на программное обеспечение. Правильно, если они определены в других документах типа плана обеспечения качества. Поэтому, правильно, если созданные СТПО ограничивают рамки проектирования, но не определяют никаких деталей проектирования.

4.3 Характеристики хороших СТПО (Characteristics of a good SRS)

СТПО должны иметь следующие характеристики

- d) Правильный
- e) Однозначный
- f) Полный
- g) Последовательный
- h) Оцениваемы по важности и/или стабильности
- i) Поддающийся проверке
- j) Поддающийся изменению
- k) Трассируемый

4.3.1 Правильный (Correct)

СТПО правильны, тогда и только тогда, когда каждое заявленное в них требование должно быть удовлетворено в программном обеспечении.

Нет никакого инструмента или процедуры, которая определяет правильность. СТПО должны быть сравнены с любыми соответствующим спецификациями более высокого уровня, такими как спецификация требований системы, с другой проектной документацией и с другими соответствующими стандартами, чтобы убедиться, что имеется соответствие. Дополнительно, клиент или пользователь может определить, правильно ли СТПО отражает фактические потребности. Трассируемость делает эту процедуру более легкой и менее подверженной ошибкам (см. 4.3.8).

4.3.2 Непротиворечивый (Unambiguous)

СТПО непротиворечивы тогда и только тогда, когда каждое заявленное требование имеет только одну интерпретацию. Необходимо, как минимум, чтобы каждая характеристика конечного продукта была описана с использованием единственного уникального термина. В случаях, когда используемый в отдельном контексте термин мог бы иметь несколько значений, он должен быть включен в глоссарий, где его смысл сделан более определенным.

СТПО - важная часть жизненного цикла программного обеспечения и используются при проектировании, реализации, контроле проекта, проверке, испытаниях и в обучении как описано в IEEE Std 1074-1991. СТПО должны быть интерпретированы однозначно, и теми, кто создают СТПО и теми, кто его использует. Однако эти группы часто не имеют одинаковой подготовки и поэтому имеют тенденцию интерпретировать требования программного обеспечения различным способом. Представления, которые улучшают спецификацию требований для разработчика, могут быть непродуктивными в понимании их пользователем и наоборот.

Для устранения неоднозначности рекомендуется использовать приемы, описанные в следующих подпунктах.

4.3.2.1 Сложности естественного языка (Natural language pitfalls)

Требования часто написаны на естественном языке (например, английском языке). Естественный язык неотъемлемо неоднозначен. Для того чтобы определить неоднозначное использование естественного языка, СТПО должны быть рассмотрены независимой стороной, с тем, чтобы выявленные неоднозначности были исправлены.

4.3.2.2 Языки спецификаций требований (Requirements specification languages)

Одним из способов избежать двусмысленности, свойственной естественному языку состоит в том, чтобы писать СТПО на специальном языке спецификаций требований. Процессоры языка автоматически обнаруживают многие лексические, синтаксические и семантические ошибки.

Такие языки имеют недостатки. Требуется время для их освоения, и многие пользователи считают их непонятными. Кроме того, эти языки лучше всего подходят для выражения определенных требований и относятся к системам определенного типа. Таким образом, они могут оказывать косвенное влияние на требования.

4.3.2.3 Инструменты представления (Representation tools).

Вообще, методы создания требований, языки и инструменты, которые поддерживают их, выделяются в три общих категории - объекты, процессы, поведение. Объектно-ориентированные подходы организовываются в терминах реальных объектов, их атрибутов, и услуг, выполняемых этими объектами. Подходы, основанные на процессах, организовывают требования в виде иерархии функций, которые связываются через потоки данных. Поведенческие подходы описывают внешнее поведение системы в терминах некоторого абстрактного понятия (типа исчислимого предиката), математических функций или конечного автомата.

Степень полезности таких инструментов и методов при создании СТПО зависит от размера и сложности программы. Здесь не делается попыток описать или рекомендовать какой-либо конкретный инструмент.

При использовании любого из этих подходов лучше сохранить описания на естественном языке. При этом те клиенты, которые не знакомы с нотацией языка, будут понимать СТПО.

4.3.3 Полный (Complete)

СТПО полны тогда, и только тогда, когда они включают следующие элементы:

- a) Все существенные требования, касающиеся функциональных возможностей, выполнения, ограничений проектирования, характеристик или внешних интерфейсов. В частности, любые внешние требования, накладываемые спецификацией системы, должны быть выявлены и рассмотрены.
- b) Все реакции программного обеспечения на все возможные типы входных данных во всех возможных ситуациях. Важно определить реакции, как на допустимые значения данных, так и на недопустимые.
- c) Все подписи и ссылки на все рисунки, таблицы и диаграммы в СТПО и определение всех терминов и единиц измерения.

4.3.3.1 Использование фразы “будет определено” (Use of TBDs)

Любые СТПО, которые используют фразу “будет определено” (TBD) – не являются полными, однако, они иногда необходимы и должны сопровождаться следующим

- a) Описанием условий, порождающих фразу “будет определено” (например, почему ответ не известен) таким образом, чтобы ситуация могла быть разрешена.
- b) Описанием того, что должно быть выполнено, чтобы устранить фразу “будет определено”, кто ответственен за устранение, когда должно быть устранено.

4.3.4 Целостный (Consistent)

Целостность относится к внутренней целостности. Если СТПО противоречат документам более высокого уровня, типа спецификации требований системы, то это неправильно (см. 4.3.1).

4.3.4.1 Внутренняя целостность (Internal consistency)

СТПО внутренне целостны тогда и только тогда, когда нет подмножеств отдельных требований, описанных в СТПО, находящихся в противоречии между собой. Имеются три типа вероятных конфликтов в СТПО:

- a) Указанные характеристики объектов могут находиться в противоречии. Например,
 - 1) Выходной отчет может быть описан в одном требовании как табличный, а в другом как - текстовый.
 - 2) Одно требование может установить, что все индикаторы должны быть зеленого цвета, а другое требование, что голубого.
- b) Может иметься логический или временной конфликт между двумя указанными действиями. Например,
 - 1) Одно требование может определить, что программа сложит две входных величины, другое - умножит их.

- 2) Одно требование устанавливает, что событие “А” должно всегда следовать за событием “В”, в то время как, другое требование устанавливает, что события “А” и “В” происходят одновременно.
- 3) Два или более требований могут описывать один и тот же объект, но использовать различные термины для этого объекта. Например, запрос программы на ввод пользователя может называться "prompt" в одном требовании и "cue" в другом. Использование стандартных терминологии и определений улучшает целостность.

4.3.5 Оцениваемый по важности и/или стабильности (Ranked for importance and/or stability)

СТПО оценивается по важности и/или стабильности, если каждое требование имеет идентификатор, который определяет важность и/или стабильность данного требования.

Обычно, не все требования, которые касаются программного обеспечения, одинаково важны. Некоторые требования могут быть обязательны, особенно для критических применений, в то время как другие могут быть только желательны.

Каждое требование в СТПО должно быть оценено для того, чтобы делать эти различия ясными и явными. Идентификация требований помогает:

- a. Потребителям давать более точное толкование каждому требованию, что часто вскрывает неявные предположения,
- b. Разработчикам принимать правильные проектные решения и уделять необходимое внимание различным частям программного продукта.

4.3.5.1 Степень стабильности (Degree of stability)

Один метод идентификации требований использует меру стабильности. Стабильность может быть выражена числом ожидаемых изменений любого требования, основанного на опыте и знании вероятных событий, воздействующих на организацию, функций и персонала, поддерживаемого программным обеспечением.

4.3.5.2 Степень потребности (Degree of necessity)

Другой способ оценки требований состоит в том, чтобы отличить классы требований как обязательные, условные и необязательные.

- a) Обязательный. Подразумевает, что программное обеспечение не будет приемлемо, если эти требования не будут включены и согласованы.
- b) Условный. Подразумевает, что эти требования улучшат изделие программного обеспечения, но программный продукт будет приемлем и при их отсутствии.
- c) Необязательный. Подразумевает класс функций, которые могут заслуживать внимания, а могут и не заслуживать внимания. Это дает поставщику возможность предложить какие-то возможности, выходящие за рамки СТПО.

4.3.6 Проверяемость (Verifiable)

СТПО проверяемо тогда и только тогда, когда каждое требование, определенное в нем, проверяемо. Требование проверяемо тогда и только тогда, когда существует конечный приемлемый по стоимости процесс, с помощью которого человек или машина может проверить, что программное обеспечение выполняет требование. Любое неоднозначное требование не проверяемо.

Непроверяемые требования включают утверждения типа "хорошо работать", "хороший человеческий интерфейс", "будет обычно случаться". Эти требования не проверяемы, потому что невозможно определить термины "хороший", "хорошо", или "обычно". Утверждение "программа никогда не должна входить в бесконечную рекурсию" не проверяемо, потому что испытание этого качества теоретически невозможно.

Пример проверяемого утверждения

Завершение программы должно быть произведено в пределах 20 с в 60 % случаев и в пределах 30 с в 100 % случаев

Это утверждение проверяемо, потому что использует конкретные термины, измеримые количественно.

Если не может быть разработан процесс, с помощью которого определяется, выполнено ли данное требование в программном обеспечении, то требование должно быть удалено или пересмотрено.

4.3.7 Изменяемость (Modifiable)

СТПО изменяемо, тогда и только тогда, когда имеет такие структуру и стиль, при которых любые изменения требований могут быть сделаны легко, полностью, последовательно и при сохранении структуры. Вообще изменяемость требует в СТПО

- a) Иметь последовательную и удобную в работе структуру с оглавлением, индексом, и явными взаимными ссылками
- b) Не быть избыточным; то есть одно и то же требование не должно встречаться больше чем одном месте в СТПО
- c) Выражать каждое требование предпочтительнее отдельно, чем смешанным с другими требованиями

Сама по себе избыточность не является ошибкой, но она может легко приводить к ошибкам. Избыточность может иногда помогать делать СТПО удобочитаемыми, но могут возникать проблемы при изменении документа. Например, если требование было изменено только одним из мест, где оно появляется, то СТПО становится неоднозначным. В случае необходимости избыточности, СТПО должны включить явные перекрестные ссылки.

4.3.8 Трассируемость (Traceable)

СТПО трассируемы, если происхождение каждого из требований ясно и если СТПО обеспечивает легкий доступ к каждому требованию из документации, создаваемой при последующей разработке или из документации, которая расширяет СТПО. Рекомендуются два типа трассируемости.

- a) Трассируемость назад (то есть к предыдущим стадиям разработки). Это зависит от каждого требования, явно ссылающегося на источник в более ранних документах.
- b) Трассируемость вперед (то есть ко всем документам, порожденным СТПО). Это зависит от каждого требования в СТПО, имеющего уникальное имя или ссылочный номер.

Трассируемость вперед СТПО особенно важна, когда программное обеспечение переходит к стадии эксплуатации и поддержки. Так как коды и проектные документы изменяются, необходимо иметь возможность установить полный набор требований, на которые возможно воздействие при таких изменениях.

4.4 Совместная подготовка СТПО (Joint preparation of the SRS)

Процесс разработки программного обеспечения должен начинаться с соглашения поставщика и клиента о том, что законченное программное обеспечение должно делать. Это соглашение в виде СТПО должно быть совместно подготовлено. Это важно, потому что обычно ни клиент, ни поставщик не квалифицированы настолько, чтобы разработать хороший документ в отдельности.

- c) Клиенты обычно недостаточно хорошо понимают процесс создания и развития программного обеспечения, чтобы создавать СТПО пригодные к употреблению.
- d) Поставщики обычно недостаточно хорошо понимают предметную область клиента, чтобы определить требования для удовлетворительной системы.

Поэтому, клиент и поставщик должны работать вместе, чтобы произвести хорошо написанный и полностью понятный СТПО.

Существует ситуация, когда система и ее программное обеспечение разрабатываются одновременно. Тогда функциональные возможности, интерфейсы, производительность и другие характеристики и ограничения программного обеспечения заранее не определены, а совместно определяются более точно в процессе разработки и являются предметом переговоров и изменений. Это более трудно, но не менее важно, поскольку необходимо обеспечить характеристики, заявленные в 4.3. В частности СТПО, которые не исполняют требования родительской спецификации системы, являются неправильными.

Здесь не обсуждаются стиль, использование языка или методы хорошего языка. Это весьма важно, поскольку СТПО должны иметь хороший язык. Для руководства могут использоваться пособия по созданию технических книг.

4.5 Развитие СТПО (SRS evolution)

СТПО могут развиваться, поскольку изделия программного обеспечения улучшаются. Бывает не всегда возможным определить некоторые детали к тому времени, когда проект начинается. Например, бывает невозможным определить все форматы экрана для диалоговой программы на стадии разработки требований. Дополнительные изменения могут возникать при обнаружении в СТПО отсутствующих частей, недостатков и погрешностей.

В этом процессе необходимо учитывать два следующих главных момента:

- a) Требования должны быть определены настолько полно и тщательно, насколько это возможно на текущий момент времени, даже если эволюционные изменения можно предвидеть как неизбежные. Тот факт, что требования являются неполными, должны быть отражены в примечаниях.
- b) Формальный процесс изменения должен быть начат, чтобы идентифицировать, управлять, отслеживать и формировать отчеты об изменениях. Утвержденные изменения в требованиях должны быть включены в СТПО так, чтобы
 - 1) Обеспечить точный и полный контроль изменений.
 - 2) Разрешить просмотр текущих и измененных частей СТПО

4.6 Прототипирование (Prototyping)

Прототипирование часто используется на этапе разработки требований проекта. Имеется много инструментов, которые позволяют из шаблона, путем задания параметров легко и быстро получать прототипы системы. См. также ASTM Std 1340-90.

Прототипы полезны по трем причинам:

- a) Очень вероятно, что клиент захочет рассмотреть прототип и реагировать на него, а не читать СТПО и реагировать на него. Таким образом, опытный образец обеспечивает быструю обратную связь.
- b) Опытный образец показывает непредвиденные аспекты поведения систем. Таким образом, он не только отвечает на возникшие вопросы, но и задает новые. Это помогает достичь завершения разработки СТПО.
- c) Основанные на опытном образце СТПО имеют тенденцию подвергаться меньшему количеству изменений в течение разработки, таким образом, время разработки укорачивается.

Опытный образец должен использоваться, как способ выявить требования программного обеспечения. Некоторые характеристики, такие как типы экранов или форматы сообщений могут быть получены непосредственно из прототипа. Другие требования могут быть получены при проведении экспериментов с прототипом.

4.7 Включение проекта в СТПО (Embedding design in the SRS)

Требование определяет видимую извне функцию или характеристику системы. Проект описывает отдельный подкомпонент системы и/или интерфейсы с другими подкомпонентами. Создатели СТПО должны ясно различать между идентификацией требуемых ограничений проекта и проектированием определенного проекта. Обратите внимание, что каждое требование в СТПО ограничивает альтернативы проекта. Это не означает, что каждое требование является проектом.

СТПО должны определить, какие функции должны быть выполнены, какие данные произвести, какой результат получен, где расположены, для кого. СТПО должны сосредоточиться на услугах, которые будут оказаны. СТПО обычно не должны определять пункты проекта типа следующих:

- a) Разделение программного обеспечения на модули
- b) Распределение функций по модулям
- c) Описание потока информации или взаимодействия между модулями
- d) Выбор структур данных

4.7.1 Необходимые требования проекта (Necessary design requirements)

В частных случаях некоторые требования могут строго ограничить проект. Например, требования секретности или безопасности могут отражаться непосредственно в проекте. Это требования типа

- a) Держать некоторые функции в отдельных модулях
- b) Разрешить ограниченную связь между некоторыми областями программы
- c) Проверить целостность данных для критических переменных

Примеры допустимых ограничений проекта - физические требования, требования производительности, требования поддержки стандартов развития программного обеспечения, стандартов качества программного обеспечения.

Поэтому, требования должны быть сформулированы с чисто внешней точки зрения. Когда используются модели, чтобы иллюстрировать требования, помните, что модель только указывает внешнее поведение и не определяет проект.

4.8 Включение требований проектирования в СТПО (Embedding project requirements in the SRS)

СТПО должны определять изделие программного обеспечения, а не процесс его создания.

Проектные требования дают понимание между клиентом и поставщиком относительно договорных вопросов, имеющих отношение к производству программного обеспечения и таким образом не должны включаться в СТПО. Обычно это такие пункты

- a) Стоимость
- b) План-графики поставки
- c) Отчетные мероприятия
- d) Методы разработки программного обеспечения
- e) Гарантии качества
- f) Испытания и критерии проверки
- g) Процедуры приемки

Проектные требования определяются в других документах, обычно в плане разработки программного обеспечения, плане тестирования программного обеспечения или в актах приема работы.

5 Разделы СТПО (The parts of an SRS)

В этом разделе обсуждается каждая из необходимых частей СТПО. Разделы расположенные на схеме, показанной на рис. 1 могут служить образцом для разработки СТПО.

| |
|--|
| Оглавление |
| 1. Введение |
| 1.1 Цель |
| 1.2 Границы применения |
| 1.3 Термины, акронимы, сокращения |
| 1.4 Ссылки |
| 1.5 Краткий обзор |
| 2. Общее описание |
| 2.1 Перспективы изделия |
| 2.2 Функции изделия |
| 2.3 Характеристики пользователя |
| 2.4 Ограничения |
| 2.5 Предположения и зависимости |
| 3. Детальные требования (См. 5.3.1-5.3.8 для объяснений возможных детальных требований. См. также приложение А для нескольких различных структур этого раздела СТПО) |
| Приложения |
| Индексы |

Рисунок 1 Образец схемы СТПО

Хотя СТПО не обязательно должна следовать приведенному плану или использовать данные названия разделов, правильная СТПО должна включать всю информацию, которая здесь обсуждается.

5.1 Введение (Introduction) (Раздел 1 СТПО)

Введение СТПО должно обеспечить краткий обзор всего СТПО. Оно должно содержать следующие подразделы:

- a) Цель
- b) Границы применения
- c) Определения, акронимы (аббревиатуры) и сокращения
- d) Краткий обзор
- e) Ссылки

5.1.1 Цели (Purpose) (1.1 из СТПО)

Этот подраздел должен выполнить следующее:

- a) Очертить цель детальных СТПО
- b) Определить аудиторию, для которой предназначено СТПО

5.1.2 Границы применения (Scope) (1.2 из СТПО)

Этот подраздел должен

- c) Идентифицировать изделие (я) программного обеспечения, которое будет произведено, наименование (например, “Серверная СУБД”, “Генератор Отчетов” и т.д.)
- d) Объяснить, что изделие (я) программного обеспечения будет, и, если необходимо, не будет делать
- e) Описать применения определяемого программного обеспечения, включая важные преимущества, объекты и цели
- f) Согласовываться со сходными утверждениями в спецификациях более высокого уровня (например, спецификацией требований системы), если они существуют

5.1.3 Термины, акронимы (аббревиатуры), сокращения (Definitions, acronyms, and abbreviations) (1.3 из СТПО)

Этот подраздел должен обеспечить определения всех терминов, акронимов (аббревиатур) и сокращений, которые необходимы для интерпретации СТПО должным образом. Эта информация может быть обеспечена посредством ссылок на одно или большее количество приложений к СТПО или посредством ссылок на другие документы.

5.1.4 Ссылки (References) (1.4 из СТПО)

Этот подраздел должен

- a) Обеспечить полный список всех документов, упомянутых в каком либо месте в СТПО
- b) Идентифицировать каждый документ наименованием, номером (если периодическое издание), датой, издающей организацией.
- c) Определить источники, в которых ссылки можно получить

Эта информация может быть обеспечена посредством ссылок на одно или большее количество приложений к СТПО или посредством ссылок на другие документы.

5.1.5 Краткий обзор (Overview) (1.5 из СТПО)

Этот подраздел должен

- a) Описать, что содержит остальная часть СТПО
- b) Объяснить, как СТПО структурировано

5.2 Общее описание (Overall description) (Раздел 2 из СТПО)

Эта раздел СТПО должен описать общие факторы, которые затрагивают изделие и требования. Этот раздел не определяет специфические требования. Вместо этого, он обеспечивает, основу для их подробного определения в разделе 3, и делает их более легкими для понимания.

Этот раздел обычно состоит из шести следующих подразделов:

- a) Описание изделия
- b) Функции изделия
- c) Характеристики пользователей
- d) Ограничения
- e) Предположения и зависимости
- f) Поднаборы требований

5.2.1 Описание изделия (*Product perspective*) (2.1 из СПТО)

В этом подразделе СПТО данный продукт необходимо рассмотреть с точки зрения связанных с ним продуктов. Если изделие независимо и полностью автономно, здесь об этом должно быть сказано. Если СПТО определяет изделие, которое является компонентом большей системы, как это часто бывает, то этот подраздел должен связать требования той большей системы с функциональными возможностями данного изделия и должен идентифицировать интерфейсы между той системой и данным изделием.

Может быть полезна диаграмма, показывающая главные компоненты большей системы, их взаимодействие и внешние интерфейсы.

Этот подраздел должен также описать, как программное обеспечение соответствует различным внутренним ограничениям. Например, он мог бы включать:

- a) Интерфейсы системы
- b) Интерфейсы пользователя
- c) Интерфейсы аппаратных средств ЭВМ
- d) Интерфейсы программного обеспечения
- e) Интерфейсы коммуникаций
- f) Ограничения памяти
- g) Функционирование (Operations)
- h) Требования настройки рабочих мест

5.2.1.1 Интерфейсы системы (*System interfaces*)

Для того чтобы обеспечить требования системы, описание интерфейсов и соответствие системе, необходимо внести в список каждый интерфейс системы и идентифицировать функциональные возможности ее программного обеспечения.

5.2.1.2 Интерфейсы пользователя (*User interfaces*)

Необходимо определить:

- a) Логические характеристики каждого интерфейса между изделием программного обеспечения и пользователями. Необходимо включить те характеристики конфигурации (например, требуемые форматы экрана, расположение страниц или окон, содержание любых сообщений или меню, наличие программируемых функциональных клавиш) которые требуются для выполнения требований программного обеспечения.

- б) Все аспекты оптимизации интерфейса между продуктом программного обеспечения и пользователями, которые этот продукт используют. Это может просто включать список того, что система должна и чего она не должна делать с точки зрения пользователя. Одним из примеров является требование выбора длинных или коротких сообщений об ошибках. Аналогично другим требованиям эти требования должны быть проверяемыми, например, " машинистка 4-го разряда может выполнять функцию X за Z минут после 1 часа обучения" предпочтительнее, чем "машинистка может выполнять функцию X" (Это может также быть определено в системных характеристиках программного обеспечения под разделом, озаглавленным "Удобство использования" (*Ease of Use*)).

5.2.1.3 Интерфейсы аппаратных средств ЭВМ (Hardware Interfaces)

Здесь необходимо определить логические характеристики каждого взаимодействия (invoice) между изделием программного обеспечения и компонентами аппаратных средств ЭВМ системы. Это включает характеристики конфигурации (номера портов, наборов инструкций и т.д.), охватывает такие вопросы как, перечень поддерживаемых устройств, способы поддержки, протоколы. Например, терминал может быть специфицированным как поддерживающим полный экран, так и в противоположность этому поддерживающим строку.

5.2.1.4 Интерфейсы программного обеспечения (Software Interfaces)

Здесь необходимо определить использование других требуемых изделий программного обеспечения (например, системы управления данными, операционной системы или математического пакета) и интерфейсов с другими прикладными системами (например, связь между системой расчета с дебиторами и системой ведения главной книги). Для каждого требуемого изделия программного обеспечения необходимо обеспечить следующее:

- а) Наименование
- б) Мнемоническое наименование
- с) Номер спецификации
- д) Номер версии
- е) Источник

Для каждого интерфейса необходимо обеспечить следующее:

- а) Обсуждение назначения взаимодействующих программ по отношению к данному продукту.
- б) Определение интерфейса в терминах содержания и формата сообщений. Правильно зарегистрированные интерфейсы не требуют детального описания, но ссылка на документ, описывающий интерфейс необходима.

5.2.1.5 Интерфейсы коммуникаций (Communications interfaces)

Здесь необходимо определить различные интерфейсы средств связи типа протоколов локальной сети и т.д.

5.2.1.6 Ограничения памяти (Memory constraints)

Здесь необходимо определить любые применимые характеристики и их значения для оперативной и постоянной памяти.

5.2.1.7 Действия (Operations)

Здесь необходимо определить нормальные и специальные действия, необходимые пользователям, типа

- a) Различные способы действий в организации пользователя; например операции, инициируемые пользователем
- b) Периоды диалоговых действий и периоды оставленных без отклика действий
- c) Функции поддержки обработки данных
- d) Действия резервного копирования и восстановления

ПРИМЕЧАНИЕ – этот пункт иногда определяется как часть раздела интерфейсов пользователя.

5.2.1.8 Требования настройки рабочих мест (Site adaptation requirements)

Здесь необходимо

- a) Определить требования для любых данных или командных строк (initialization sequences), которые определяются для данного рабочего места, задачи, или режима эксплуатации, например, разрешение экрана (grid values), пределы безопасности (safety limits) и т.д.
- b) Определите характеристики рабочего места или решаемых задач, которые должны быть изменены, чтобы настроить программное обеспечение на специальную конфигурацию.

5.2.2 Функции изделия (Product functions) (2.2 из СТПО)

Этот подраздел СТПО должен содержать резюме основных функций, которые программное обеспечение выполняет. Например, СТПО для программы бухгалтерского учета могут использовать этот подраздел, чтобы определить функции обслуживания счета клиента, заявление клиента и подготовку счета без упоминания обширного количества деталей, которые каждая из этих функций требует.

Иногда резюме функции, которое является необходимым для этой части, может быть получено непосредственно из разделов спецификаций более высокого уровня (если такие существуют) которые закрепляют специфические функции в изделии программного обеспечения. Обратите внимание, что ради ясности:

- a) Функции должны быть организованы в группы, которые делают список функций понятными клиенту или кому-либо, кто читает документ впервые.
- b) Могут использоваться текстовые или графические методы, чтобы показать различные функции и их отношения. Такие диаграммы не предназначены для того, чтобы показать проект изделия, но они просто показывают логические отношения между переменными.

5.2.3 Характеристики пользователей (User characteristics) (2.3 из СТПО)

Этот подраздел СТПО должен описывать те общие характеристики пользователей, для которых предназначены изделия, включая образовательный уровень, опыт, и техническую квалификацию. Не следует использовать этот раздел для того, чтобы

сформулировать детальные требования, здесь скорее необходимо объяснить причины, по которым эти требования появятся позже в разделе 3 СТПО.

5.2.4 Ограничения (Constraints) (2.4 из СТПО)

Этот подраздел СТПО должен обеспечить общее описание любых других факторов, которые ограничивают выбор разработчика. Они включают:

- a) Регулирующую политику
- b) Ограничения аппаратных средств ЭВМ (например, требования по времени выбора)
- c) Интерфейсы с другими приложениями
- d) Параллельную работу
- e) Функции протоколирования
- f) Функции управления
- g) Требования к языкам высокого уровня
- h) Протоколы интерфейсов синхронизации сигналов (например, XON-XOFF, ACK-NACK)
- i) Требования надежности
- j) Критичность приложения
- k) Соображения безопасности и секретности

5.2.5 Предположения и зависимости (Assumptions and dependencies) (2.5 из СТПО)

Этот подраздел СТПО определяет перечень факторов, которые затрагивают требования, определенные в СТПО. Эти факторы не являются ограничениями на программное обеспечение проекта, но любое их изменение может затронуть требования в СТПО. Например, может быть предположение, что на аппаратных средствах ЭВМ, предназначенных для изделия программного обеспечения, будет доступна определенная операционная система. Если, фактически, операционная система не доступна, СТПО должны были бы измениться.

5.2.6 Распределение требований (Apportioning of requirements) (2.6 из СТПО)

Этот подраздел СТПО должен определить требования, которые могут быть отсрочены до будущих версий системы.

5.3 Детальные требования (Specific requirements) (Секция 3 из СТПО)

Этот раздел СТПО должен содержать все требования программного обеспечения с достаточной степенью детализации, чтобы позволить разработчикам спроектировать и тестировщикам проверить систему, удовлетворяющую этим требованиям. Везде в этом разделе каждое требование должно обеспечивать взаимодействие с пользователями, операторами или другими внешними системами. Эти требования должны включать, как минимум, описание всех входов системы, всех выходов системы и всех функций, которые выполняются системой в ответ на вход или для получения результатов на выходе.

Поскольку это - часто самая большая и наиболее важная часть СТПО, то необходимо применять следующие принципы:

- a) Детальные требования должны быть заявлены в соответствии с правилами, описанными в 4.3 данного стандарта.
- b) Детальные требования должны иметь перекрестные ссылки к более ранним документам, к которым имеют отношение.
- c) Все требования должны быть уникально идентифицированы.
- d) Для повышения удобочитаемости необходимо особое внимание уделить структурированию требований.

Перед изучением конкретных способов структурирования требований полезно рассмотреть различные разделы требований, представленные ниже.

5.3.1 Внешние интерфейсы (*External interfaces*)

Здесь должно быть детальное описание всех входов и выходов системы программного обеспечения. Данный подраздел является дополнением к описанию интерфейса в разделе 5.2. Он не должен повторять информацию, находящуюся в разделе 5.2.

Описание содержания и формата внешнего интерфейса определяется следующим образом:

- a) Наименование пункта
- b) Описание цели
- c) Источник входных или назначение выходных данных
- d) Диапазон допустимых значений, точность и/или допустимые отклонения
- e) Единицы измерения
- f) Временные характеристики
- g) Отношения к другим входам / выходам
- h) Форматы / организация экрана
- i) Форматы / организация окна
- j) Форматы данных
- k) Форматы команд
- l) Конечные сообщения

5.3.2 Функции (*Functions*)

Функциональные требования определяют фундаментальные действия, которые должны иметь место в программном обеспечении при получении и обработке входных данных и при формировании выходных данных. Они обычно включаются в список, как "должна" в начале определения функционального требования. "Система должна ..."

Описание включает:

- a) Проверку допустимости данных на входах
- b) Точную последовательность действий
- c) Действия при возникновении исключительных ситуаций, включая

- 1) Переполнение
 - 2) Средства связи (Communication facilities)
 - 3) Обработку ошибок и восстановление
- d) Влияние параметров
- e) Отношения входных данных к выходным данным, включая
- 1) Последовательности Входных данных / выходных данных
 - 2) Формулы для преобразования входных данных в выходные

Может быть определено деление функциональных требований на подфункции или подпроцессы. Это не подразумевает, что разрабатываемое программное обеспечение будет иметь такое же деление.

5.3.3 Требования исполнения (*Performance requirements*)

Этот подраздел определяет требования для статических и динамических численных характеристик. Эти требования могут включать:

- a) Число терминалов, которые должны быть поддержаны
- b) Количество одновременно работающих пользователей, которое должно быть поддержано
- c) Количество и тип информации, которая должна обрабатываться

Требования для статических численных характеристик иногда выделяются в отдельный раздел под названием Нагрузка системы.

Требования для динамических численных характеристик могут включать, например, количество транзакций и задач, объем данных, которые будут обработаны в пределах некоторых интервалов времени для нормальных условий и условий пиковой рабочей нагрузки.

Все эти требования должны быть заявлены в терминах единиц измерения.

Например,

95 % сделок должны быть обработаны меньше чем 1 s

предпочтительнее чем,

оператор не должен ждать, чтобы завершить сделку.

ПРИМЕЧАНИЕ - численные ограничения, применяемые к одной специфической функции обычно определяются как часть подпараграфа описания работы этой функции.

5.3.4 Требования логики базы данных (*Logical database requirements*)

Здесь определяются требования логики для любой информации, которая должна быть помещена в базу данных. Может включать:

- d) Типы информации, используемые различными функциями
- e) Частота использования
- f) Способы доступа.
- g) Сущности данных и их отношения
- h) Ограничения целостности
- i) Требования хранения данных

5.3.5 Ограничения проекта (Design constraints)

Здесь определяются ограничения разработки, которые могут накладываться другими стандартами, ограничениями аппаратных средств ЭВМ и т.д.

5.3.5.1 Соглашение о стандартах (Standards compliance)

Этот подраздел должен определить требования, которые вытекают из существующих стандартов или инструкций. Он может включать:

- a) Формат сообщений
- b) Именованние данных
- c) Отчетные процедуры (Accounting procedures)
- d) Протоколирование (Audit tracing)

Например, можно определить требование к программному обеспечению протоколировать процесс обработки. Такой протокол необходим для некоторых приложений, чтобы выполнить минимальные требования регулирующих или финансовых стандартов.

Требование протоколирования может, например, определить, что все изменения в базе данных платежной ведомости должны быть зарегистрированы в файле протокола со значениями, которые были до и после изменений.

5.3.6 Характеристики программного обеспечения системы (Software system attributes)

Имеется множество характеристик программного обеспечения, которые могут служить как требования. Важно, чтобы эти требования были определены таким образом, чтобы их можно было объективно проверить. Следующие пункты обеспечивают примеры некоторых характеристик.

5.3.6.1 Надежность (Reliability)

Здесь определяются факторы, требуемые для того, чтобы установить требуемую надежность программного обеспечения системы во время поставки.

5.3.6.2 Эксплуатационная готовность (Availability)

Здесь определяются факторы, требуемые для того, чтобы гарантировать определенный уровень эксплуатационной готовности всей системы, такие как точки останова (checkpoint), восстановление и перезапуск.

5.3.6.3 Безопасность (Security)

Здесь определяются факторы, которые должны обеспечить защиту программного обеспечения от случайного или злонамеренного доступа, использования, модификации, разрушения или раскрытия. Специфические требования в этой области могли бы включать следующее:

- a) Использовать некоторые методы шифрования
- b) Содержать детальные журналы или исторические наборы данных
- c) Назначать некоторые функции в различные модули
- d) Ограничить связи между некоторыми областями программы
- e) Проверять целостность данных для критических переменных

5.3.6.4 Ремонтпригодность (Maintainability)

Здесь определяются параметры программного обеспечения, которые непосредственно касаются легкости обслуживания программного обеспечения. Могут иметься некоторые требования для определения модульности, интерфейсов, сложности и т.д. Здесь не следует размещать требования только потому, что они, как полагают, являются хорошей проектной практикой.

5.3.6.5 Переносимость (Portability)

Здесь определяются параметры программного обеспечения, которые касаются легкости переноса программного обеспечения на другие машины и/или операционные системы. Может включать:

- f) Процент компонентов с машинозависимым кодом
- g) Процент кода, который является машинозависимым
- h) Использование проверенного переносимого языка
- i) Использование конкретного компилятора или поднабора языков
- j) Использование конкретной операционной системы

5.3.7 Структурирование детальных требований (Organizing the specific requirements)

Для нетривиальных систем детальные требования могут быть обширными. По этой причине, рекомендуется обеспечивать такое их структурирование, которое делает их оптимальными для понимания. Не существует оптимальной структуры для всех систем. Различные способы структурирования требований для различных классов систем предоставлены в разделе 3 СТПО. Некоторые структуры описаны в следующих подразделах.

5.3.7.1 Режим системы (System mode)

Некоторые системы ведут себя весьма различно в зависимости от режима работы. Например, система управления может иметь различные наборы функций в зависимости от использования: обучение, нормальная или аварийная ситуация. При организации деления на режимы используйте схемы, показанные в разделах 1 или 2 приложения А. Выбор определяется тем, зависят ли интерфейсы и выполнение программы от режима работы.

5.3.7.2 Классы пользователей (User class)

Некоторые системы обеспечивают различные наборы функций для различных классов пользователей. Например, система управления подъемника представляет различные возможности пассажирам, рабочим обслуживания и пожарным. При организации этого раздела используйте схему, показанную в разделе 3 приложения А.

5.3.7.3 Объекты (Objects)

Объекты - реальные сущности, которые имеют дубликат внутри системы. Например, в системе контроля пациента объектами являются пациенты, датчики, медсестры, комнаты, врачи, лекарства и т.д. С каждым объектом связан набор атрибутов (этого объекта) и функций (выполняемых этим объектом). Эти функции также вызывают услуги, методы или процессы. При организации этого раздела, используйте схему,

показанную в разделе 4 приложения А. Обратите внимание, что наборы объектов могут разделяться по атрибутам и услугам. Они сгруппированы вместе как классы.

5.3.7.4 Особенности (Feature)

Особенность – желаемый извне сервис, предоставляемый системой, который может требовать последовательность входных данных, чтобы произвести желательный результат. Например, в телефонной системе особенности включают внутригородской звонок, ускоренный вызов и вызов конференции. Каждая особенность вообще описывается в последовательности пар воздействие - реакция. При организации этого раздела используйте схему, показанную в разделе 5 в приложении А.

5.3.7.5 Воздействие (Stimulus)

Некоторые системы могут быть лучше организованы, если их функции описаны в терминах воздействия. Например, функции автоматической системы приземления самолета могут быть организованы в разделы для следующих воздействий: потеря мощности, порыв ветра, внезапное изменение крена, чрезмерная вертикальная скорость и т.д. При организации этого раздела, используйте схему, показанную в разделе 6 приложения А.

5.3.7.6 Реакция (Response)

Некоторые системы могут быть лучше организованы, если их функции описаны в терминах генерации реакции. Например, функции системы учета персонала могут быть организованы в разделы, соответствующие всем функциям, связанным с созданием платежной ведомости, соответствующие всем функциям, связанным с созданием текущего списка служащих и т.д. Использовать раздел 6 в приложении А (все встречающиеся слова “воздействие” заменить на “реакция”).

5.3.7.7 Функциональные иерархии (Functional hierarchy)

Когда ни одна из вышеупомянутых организационных схем не подходит, полные функциональные возможности могут быть организованы в иерархию функций, организованных по принципу общих входов, общих выходов или общим доступом к внутренним данным. Могут использоваться диаграммы потоков данных и словари данных для того, чтобы показать отношения между функциями и данными. При организации этого раздела используйте схему 7 приложения А.

5.3.8 Дополнительные комментарии (Additional comments)

Когда определено новое требование, оно может соответствовать нескольким способам структурирования, описанным в 5.3.7.7. В таких случаях, структурируйте детальные требования во множественные иерархии, приспособленные к специальным потребностям специфицируемой системы. Например, см. 8 в приложении А для организации, объединяющей класс пользователя и особенность. Любые дополнительные требования могут быть помещены в отдельный раздел в конце СТПО.

Существует большое количество доступных описаний, методов и автоматизированных инструментов поддержки, помогающих в документировании требований. Главным образом, их ценность - это функция структурирования. Например, при структурировании по режимам (when organizing by mode), могут быть полезными конечные автоматы или диаграммы состояний; при структурировании по объектам, может быть полезным объектно-ориентированный анализ; при структурировании по особенностям могут быть полезными последовательности «воздействие – реакция»; при

структурировании по функциональной иерархии могут быть полезными диаграммы потоков данных и словари данных.

В любой из схем, приведенных в 1-8, разделы, которые именуются как "Функциональные Требования", могут быть написаны на родном языке (например, Английский язык), в псевдокодах, на языке описания систем, или в виде четырех подразделов: Введение, Входы, Обработка (Processing), Выходы.

5.4 Информационная поддержка (Supporting information)

Информационная поддержка облегчает использование СТПО. Она включает

- a) Оглавление
- b) Индексы
- c) Приложения

5.4.1 Оглавления и индексы (Table of contents and index)

Оглавление и индексы весьма важны и должны следовать принятым методам построения.

5.5 Приложения (Appendixes)

Приложения не всегда рассматриваются как часть фактической спецификации требований и не всегда необходимы. Они могут включать

- a) Образцы форматов ввод - вывода, описания анализа стоимости обучения, или результатов обследования пользователей
- b) Сопровождающая или подготовительная информация, которая может помочь читателям СТПО
- c) Описание проблем, которые будут решены с помощью программного обеспечения
- d) Специальные упаковочные инструкции (Special packaging instructions) для кода и информационные средства, необходимые для обеспечения безопасности, экспорта, начальной загрузки или другие требования

Если имеются приложения, СТПО должны явно объявить, рассматриваются или нет приложения как часть требований.

6 Приложение А

(Информативно)

6.1 Шаблон раздела 3 СТПО организованный по режимам: Версия 1

3 Детальные требования

3.1 Требования на внешние интерфейсы

3.1.1 Интерфейсы пользователя

3.1.2 Интерфейсы аппаратных средств ЭВМ

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

3.2.2 Режим 2

3.2. m. Режим m

3.2. m.1 Функциональное требование m.

3.2. m.n Функциональное требование m.n

3.3 Требования исполнения

3.4 Ограничения проекта

3.5 Характеристики программного обеспечения системы

3.6 Другие требования

6.2 Шаблон раздела 3 СТПО организованный по режимам: Версия 2

3. Детальные 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 Интерфейсы аппаратных средств ЭВМ

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 Функциональных требование 1

3.1.1.2.n Функциональное требование n

3.1.1.3 Требования исполнения

3.1.2 Режим 2

3.1.m. Режим m.

3.2 Ограничения проекта

3.3 Характеристики программного обеспечения системы

3.4 Другие требования

6.3 Шаблон раздела 3 СТПО организованный по классам пользователей

3 Детальные требования

3.1 Требования внешних интерфейсов

3.1.1 Интерфейсы пользователя

3.1.2 Интерфейсы аппаратных средств ЭВМ

3.1.3 Интерфейсы программного обеспечения

3.1.4 Интерфейсы связи

3.2 Функциональные требования

3.2.1 Класс пользователя 1

3.2.1.1 Функциональное требование 1.1

3.2.1.n Функциональное требование 1. n

3.2.2 Класс пользователя 2

3.2.m. Класс пользователя m.

3.2.m.1 Функциональное требование m.1

3.2.m.n Функциональное требование m.n

3.3 Требования исполнения

3.4 Ограничения проекта

3.5 Характеристики программного обеспечения системы

3.6 Другие требования

6.4 Шаблон раздела 3 СТПО организованный по объектам

3 Детальные требования

3.1 Требования внешних интерфейсов

3.1.1 Интерфейсы пользователя

3.1.2 Интерфейсы аппаратных средств ЭВМ

3.1.3 Интерфейсы программного обеспечения

3.1.4 Интерфейсы связи

3.2 Классы / объекты

- 3.2.1 Класс / объект 1
 - 3.2.1.1 Атрибуты (собственные или унаследованные)
 - 3.2.1.1.1 Атрибут 1
 - 3.2.1.1.n Атрибут n
 - 3.2.1.2 Функции (методы. Собственные или унаследованные)
 - 3.2.1.2.1 Функциональное требование 1.
 - 3.2.1.2.m Функциональное требование m
 - 3.2.1.3 Сообщения (полученные или отправленные)
- 3.2.2 Класс / объект 2
- 3.2.p Класс / объект p
- 3.3 Требования исполнения
- 3.4 Ограничения проекта
- 3.5 Характеристики программного обеспечения системы
- 3.6 Другие требования

6.5 Шаблон раздела 3 СТПО организованный по особенностям

- 3 Детальные требования
 - 3.1 Требования внешних интерфейсов
 - 3.1.1 Интерфейсы пользователя
 - 3.1.2 Интерфейсы аппаратных средств ЭВМ
 - 3.1.3 Интерфейсы программного обеспечения
 - 3.1.4 Интерфейсы связи
 - 3.2 Особенности системы
 - 3.2.1 Особенность системы 1
 - 3.2.1.1 Введение / цель
 - 3.2.1.2 Последовательность воздействие / реакция
 - 3.2.1.3 Связанные функциональные требования
 - 3.2.1.3.1 Функциональное требование 1
 - 3.2.1.2.n Функциональное требование n
 - 3.2.2 Особенность системы 2
 - 3.2.m Особенность системы m.
 - 3.3 Требования исполнения
 - 3.4 Ограничения проекта
 - 3.5 Характеристики программного обеспечения системы
 - 3.6 Другие требования

6.6 Шаблон раздела 3 СТПО организованный по воздействиям

3 Детальные требования

3.1 Требования внешних интерфейсов

3.1.1 Интерфейсы пользователя

3.1.2 Интерфейсы аппаратных средств ЭВМ

3.1.3 Интерфейсы программного обеспечения

3.1.4 Интерфейсы связи

3.2 Функциональные требования

3.2.1 Воздействие 1

3.2.1.1 Функциональное требование 1.1

3.2.1.n Функциональное требование 1.n

3.2.2 Воздействие 2

3.2.m Воздействие m

3.2.m.1 Функциональное требование m.1

3.2.m.n Функциональное требование m.n

3.3 Требования исполнения

3.4 Ограничения проекта

3.5 Характеристики программного обеспечения системы

3.6 Другие требования

6.7 Шаблон раздела 3 СТПО организованный по функциональной иерархии

3 Детальные требования

3.1 Требования внешних интерфейсов

3.1.1 Интерфейсы пользователя

3.1.2 Интерфейсы аппаратных средств ЭВМ

3.1.3 Интерфейсы программного обеспечения

3.1.4 Интерфейсы связи

3.2 Функциональные требования

3.2.1 Информационные потоки

3.2.1.1 Диаграмма потока данных 1

3.2.1.1.1 Сущности данных

3.2.1.1.2 Подходящие процессы (Pertinent processes)

3.2.1.1.3 Топология (Topology)

3.2.1.2 Диаграмма потока данных 2

3.2.1.2.1 Сущности данных

- 3.2.1.2.2 Подходящие процессы (Pertinent processes)
- 3.2.1.2.3 Топология (Topology)
- 3.2.1.n Диаграмма потока данных n
 - 3.2.1.n.1 Сущности данных
 - 3.2.1.n.2 Подходящие процессы (Pertinent processes)
 - 3.2.1.n.3 Топология (Topology)
- 3.2.2 Описание процессов
 - 3.2.2.1 Процесс 1
 - 3.2.2.1.1 Сущности данных на входе
 - 3.2.2.1.2 Алгоритм или формула процесса
 - 3.2.2.1.3 Сущности данных, оказывающие воздействие (Affected data entities)
 - 3.2.2.2 Процесс 2
 - 3.2.2.2.1 Сущности данных на входе
 - 3.2.2.2.2 Алгоритм или формула процесса
 - 3.2.2.2.3 Сущности данных, оказывающие воздействие (Affected data entities)
 - 3.2.2.m Процесс m
 - 3.2.2.m.1 Сущности данных на входе
 - 3.2.2.m.2 Алгоритм или формула процесса
 - 3.2.2.m.3 Сущности данных, оказывающие воздействие (Affected data entities)
- 3.2.3 Спецификации структур данных
 - 3.2.3.1 Структура 1
 - 3.2.3.1.1 Тип записи
 - 3.2.3.1.2 Элементарные поля
 - 3.2.3.2 Структура 2
 - 3.2.3.2.1 Тип записи
 - 3.2.3.2.2 Элементарные поля
 - 3.2.3.p Структура p
- 3.2.4 Словарь данных
 - 3.2.4.1 Элемент данных 1
 - 3.2.4.1.1 Наименование
 - 3.2.4.1.2 Представление
 - 3.2.4.1.3 Единицы / формат
 - 3.2.4.1.4 Точность представления / точность округления (Precision/Accuracy)

- 3.2.4.1.5 Диапазон значений
- 3.2.4.2 Элемент данных 2
 - 3.2.4.2.1 Наименование
 - 3.2.4.2.2 Представление
 - 3.2.4.2.3 Единицы / формат
 - 3.2.4.2.4 Точность представления / точность округления (Precision/Accuracy)
 - 3.2.4.2.5 Диапазон значений
- 3.2.4.1.q Элемент данных q
 - 3.2.4.q.1 Наименование
 - 3.2.4.q.2 Представление
 - 3.2.4.q.3 Единицы измерения / формат
 - 3.2.4.q.4 Точность представления / точность округления (Precision/Accuracy)
 - 3.2.4.q.5 Диапазон
- 3.3 Требования исполнения
- 3.4 Ограничения проекта
- 3.5 Характеристики программного обеспечения системы
- 3.6 Другие требования

6.8 Шаблон раздела 3 СТПО показывающий множественную организацию

3 Детальные требования

- 3.1 Требования внешних интерфейсов
 - 3.1.1 Интерфейсы пользователя
 - 3.1.2 Интерфейсы аппаратных средств ЭВМ
 - 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 Особенность 1.1
 - 3.2.1.1.1 Введение / цель
 - 3.2.1.1.2 Последовательность воздействие / реакция
 - 3.2.1.1.3 Связанные функциональные требования
 - 3.2.1.2 Особенность 1.2
 - 3.2.1.2.1 Введение / цель
 - 3.2.1.2.2 последовательность Стимула / ответа

3.2.1.2.3 Связанные функциональные требования

3.2.1.m. Особенность 1.m

3.2.1.m.1 Введение / цель.

3.2.1.m.2 Последовательность воздействие / реакция

3.2.1.m.3 Связанные функциональные требования

3.2.2 Класс пользователей 2

3.2.n Класс пользователей n

3.3 Требования исполнения

3.4 Ограничения проекта

3.5 Характеристики программного обеспечения системы

3.6 Другие требования