

Основы программной инженерии (ПОИТ)

Технологии разработки ПО. Документирование

План лекции:

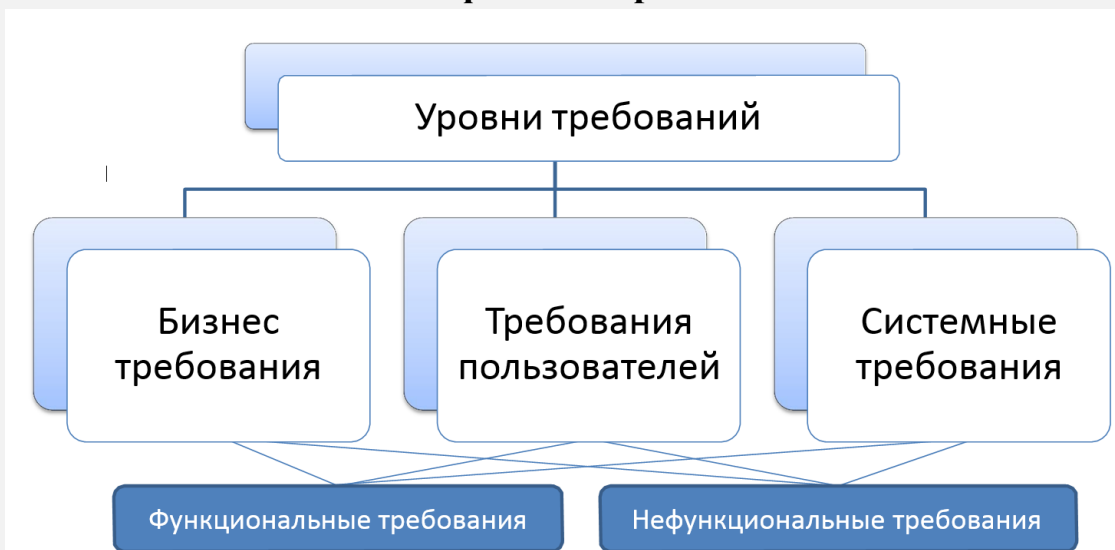
- проектная документация;
- состав и назначение документации;
- стандарты документирования.

На прошлых лекциях:

1. Управление требованиями

Требование – это утверждение, которое идентифицирует эксплуатационные, функциональные параметры, характеристики или ограничения проектирования продукта или процесса, которое *однозначно, проверяемо и измеримо*.

Классификация требований



Требование –

- ✓ *условие или возможность, необходимые пользователю для решения его задач или достижения цели (1)*
- ✓ *условие или возможность, которым должна отвечать или которыми должна обладать система или ее компонента, чтобы удовлетворить контракт,*

Цели разработки требований

- ✓ обеспечение наиболее полного и точного отражения условий или возможностей, необходимых заказчику для решения его проблем и достижения бизнес-целей;
- ✓ снижение затрат на разработку, обслуживание и поддержку сложного программного обеспечения.

<p><i>стандарт, спецификацию или иной формальный документ (2)</i></p> <p>✓ <i>документированное представление условия или возможности, указанное в (1) или (2)</i></p>	
<p>Требования делятся на:</p> <ul style="list-style-type: none"> • функциональные (то, что система позволяет сделать, желаемая функциональность); • нефункциональные (требования к оборудованию, операционной системе и т.п.). 	

Формализация функциональных требований

Диаграмма вариантов использования (англ. use-case diagram) –

диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

Диаграмма вариантов использования = Диаграмма прецедентов



- А ты строишь диаграммы при проектировании?
 - Да, как видишь

Диаграммы вариантов использования

- ✓ показывают взаимодействия между **вариантами использования** и **действующими лицами**, отражая функциональные требования к системе с точки зрения **пользователя**.
- ✓ являются исходной концептуальной моделью системы в процессе ее проектирования и разработки.

Введение

Проектов без документации не существует, вопрос в том, что что именно должно входить в конкретный базовый набор на старте и как оптимально поддерживать документацию в процессе.

Для создания действительно полезных документов надо учитывать два фактора:

- цель – для чего создается конкретный документ;
- потребитель – кто будет использовать то, что вы написали.

Базовый набор стартовой документации включает:

1. **Паспорт проекта** – документ, в котором описано кто клиент, что делаем, сроки, цель и т.д.
2. **Фич-лист** обязателен, именно с него начинается проект: РМ оценивает фич-лист, согласовывает с заказчиком, и на его основе создает план проекта.
3. **Иерархическая структура работ** – тот же фич-лист, только более структурированный и распределенный по ролям.
4. **Оценка проекта** – обязательная часть.
5. **План проекта** – важно правильно выбрать инструмент, в котором будете вести план, это может быть: Excel, Miro, другое.

Список самых используемых инструментов:

- **Google Drive** для хранения разного вида файлов с общим доступом и редактированием онлайн.
- **Dropbox** – альтернатива Google Drive.
- **Confluence** – общее рабочее пространство, база знаний для создания, хранения, редактирования документов.
- **Notion** – «википодобный» ресурс. Хорошо работает для базовой документации, особенно о каком-то продукте.
- **Miro** помогает сделать быстрое ревью документа.

Базовые правила ведения документации:

- документация должна быть **синхронизированной и, по-возможности, онлайн**. Для удаленных команд это особенно актуально: можно быстро редактировать документ и все это видят;
- **всегда проверяйте права доступа**.

Соглашения об именовании

Обычно достаточно трех простых правил:

- 1) использовать в качестве префикса дату в формате: YYYY-MM-DD;
- 2) значимое обозначение документа: например, Test_Concept_Module-A;
- 3) номер версии документа: v01.

Понятная навигация

2. Документирование

Документация – это рабочее пространство проекта.

- ✓ документирование позволяет четко разграничить зоны ответственности между участниками проекта;
- ✓ документы избавляют от ненужных конфликтов;
- ✓ позволяет в любой момент времени быстро найти нужную информацию и понять, как решать конкретные задачи;
- ✓ в документах четко прописано, кто что делает, кто за что отвечает, как работает система и что делать, если что-то пошло не так;
- ✓ команда говорит «на одном языке»;
- ✓ только тщательно описанные требования могут быть проверены на полноту и непротиворечивость.

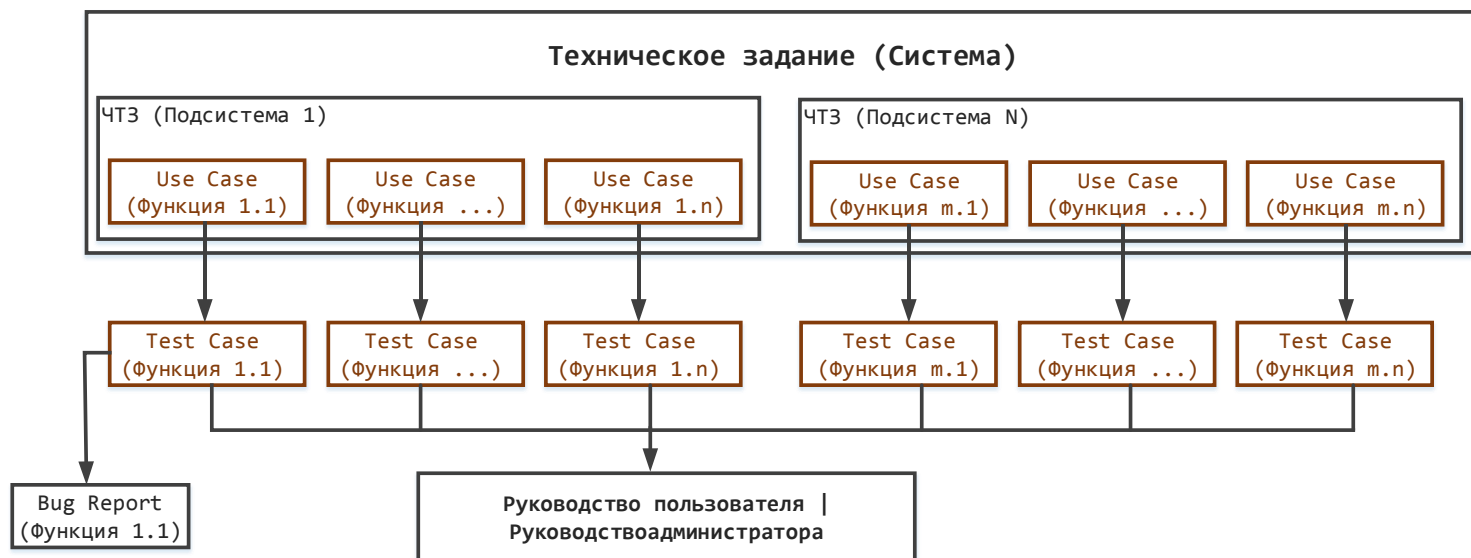
Правила хорошей документации:

- 1) Документация не должна быть избыточной и объемной. Избыточное количество текста – раздражает и затрудняет восприятие.
- 2) Вся схема документирования проекта должна быть взаимоувязанной и логичной. Если в схеме существует документ, который не связан ссылкой с каким бы то ни было другим документом, то его можно безболезненно из схемы исключить.
- 3) Вся оценка трудозатрат должна производиться только на основании описанных атомарных задач. Чем мельче оцениваемый элемент – тем точнее будет агрегированная оценка.
- 4) Всегда необходимо формировать списки оповещения заинтересованных участников.

Типы документов:

- 1) Техническое задание.
- 2) Частное техническое задание (ЧТЗ) – опционально.
- 3) Сценарий использования (Use Case).
- 4) Сценарий тестирования (Test Case).
- 5) Отчет об ошибке (Bug Report).
- 6) Руководство пользователя.
- 7) Руководство администратора – опционально.

Схема связи между документами.



где

Техническое задание

В первую очередь формируется **Техническое задание** системы.

ТЗ включает в себя:

- словарь терминов предметной области;
- описание предметной области;
- описание ролевой системы (действующие лица);
- описание функциональных требований;
- описание нефункциональных требований.

Все требования формируются на основании описания бизнес-процессов заказчика (функциональные требования).

Описание требований в ТЗ фиксирует необходимые функции на верхнем уровне.

Требования оптимально разбивать на смысловые группы по подсистемам.

Например

На верхнем уровне системы «Калькулятор» описываем подсистему «Калькулятор. Арифметические операции» с функциями «Выполнить сложение», «Выполнить вычитание» и т.д.

Частные технические задания (ЧТЗ) подсистем должны содержать:

- ссылку на пункт ТЗ;
- максимально подробную информацию по каждой функции;
- список Use Cases для функции.

Таким образом реализуется преемственность документов, что позволяет:

- во-первых, унифицировать их форму;
- во-вторых – частично реализовать повторное использование, то есть снизить затраты времени на повторное написание кода.

Например

формируем ЧТЗ на подсистему «Калькулятор. Арифметические операции». Тогда описание функции:

«Выполнить сложение.

Необходимо реализовать следующий функционал:

выполнить бинарную арифметическую операцию сложения целых чисел;

входные данные: два целых числа.

выходные данные: целое число;

...».

Use Case

Use Case – вариант использования, который описывает все действия, которые пользователь может произвести, и реакцию системы на эти действия.

Каждый вариант использования должен быть привязан к своему пункту ЧТЗ.

Наиболее оптимальным и простым форматом описания является UML-диаграммы вариантов использования, выполненные в Visio или с помощью аналогичного инструмента.

Необходимо включать описание входных и выходных данных: название переменной, её тип, ограничение на ввод данных (логические проверки и т.д.); для вычисляемых переменных указать формулу для расчета значения.

Test Case

Тестовый случай должен содержать описание тестовых сценариев.

Каждый такой документ привязывается к соответствующему Use Case

Структура Test Case:

Action (Действие)	>	Expected Result (Ожидаемый результат)	>	Test Result (Результат тестирования) (passed/failed/blocked)
----------------------	---	--	---	--

Результат тест кейса: *passed / failed / blocked* (успешно / ошибка / заблокирован)

Не забыть про проверку валидности выходных данных!

Bug Report

Отчет об ошибке (Bug Report) возникает в процессе тестирования системы как реакция на ошибку. Каждый документ должен обязательно ссылаться на соответствующий Test Case.

Документ должен содержать:

- описание предшествующих действий (можно разработать удобный для всех шаблон такого описания – это позволит сэкономить время разработчикам при воспроизведении бага);
- текстовое описание самой ошибки;
- скриншот возникшей ошибки (при необходимости).

Руководство пользователя / Руководство администратора

Это жизненно необходимые документы.

Они являются документацией для пользователей – это пользователи приложений. Можно использовать подход к построению документации по принципу описания вариантов использования (Use Case). По списку задач, которые выполняют пользователи, составляются пошаговые инструкции.

Если все Use Cases и Test Cases были тщательно описаны, аккуратно выполнены и правильно оформлены, то эти документы будут легко сформированы (почти автоматически).

Документация для разработчиков: это могут быть комментарии в коде, readme- файлы, стиль оформления кода. Логика приложения следует описать в виде блок-схем или псевдокода.

Начинать процесс документирования всегда проще «сверху» (принцип модульного проектирования сверху вниз).

Руководство пользователя

- введение
 - область применения
 - краткое описание возможностей
 - требования к уровню подготовки пользователя
 - перечень эксплуатационных документов, с которыми необходимо ознакомиться пользователю
- назначение и условия применения
 - виды деятельности и функции для автоматизации которых предназначено данное ПО
 - условия, при соблюдении которых обеспечивается применение ПО в соответствии с назначением
- подготовка к работе
 - состав и содержание дистрибутивного носителя данных
 - порядок загрузки данных и программ
 - порядок контроля и проверки работоспособности

- описание операций – для каждой операции обработки данных должно быть указано
 - наименование
 - условия, при соблюдении которых возможно выполнение операции
 - подготовительные действия
 - основные действия в требуемой последовательности
 - заключительные действия
 - ресурсы, расходуемые на операцию
 - описание всех выполняемых функций, задач, комплексов задач, процедур
 - описание операций технологического процесса обработки данных, необходимых для выполнения функций, комплексов программ, процедур.
- аварийные ситуации
 - действия в случае несоблюдения условий выполнения технологического процесса, в том числе при длительных отказах технических средств
 - действия по восстановлению программ и данных при отказе или обнаружении ошибок в данных
 - действия в случае обнаружения несанкционированного вмешательства в данные системы

3. Разработка документации программных средств и её стандартизация

Процесс документирования по стандарту ISO/IEC 12207:1995

Рост применения программных средств и их сложность требуют полной, точной и понятной документации на ПО, доступной пользователям.

Процесс документирования программных средств и систем регламентирует международный стандарт *ISO/IEC 12207:1995* и *СТБ ИСО/МЭК12207-2003*.

В данном стандарте *процесс документирования* определяется как процесс формализованного описания информации, созданной в процессе или работе жизненного цикла.

Документирование включает планирование, проектирование, разработку, выпуск, редактирование, распространение и сопровождение документов по программному продукту.

Документация разработки

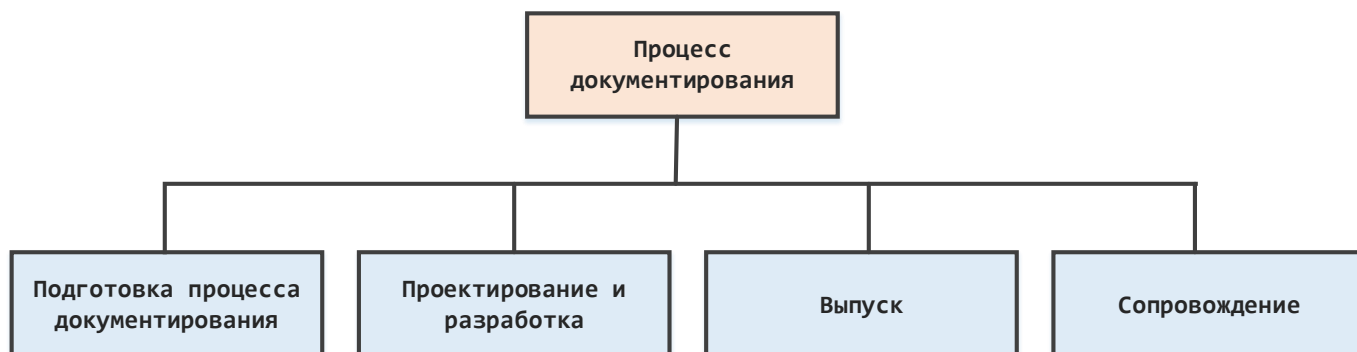
Разработка документов преследует пять целей;

- 1) документы являются средством связи между всеми вовлеченными в процесс разработки. Они описывают подробности решений, принятых относительно требований к программному обеспечению, проекту, программированию и тестированию;
- 2) документы описывают обязанности группы разработки. Они определяют, кто, что и когда делает, учитывая роль программного обеспечения, предмета работ, документации, персонала, обеспечивающего качество, и каждого вовлеченного в процесс разработки;
- 3) документы выступают как контрольные точки, которые позволяют руководителям оценивать ход разработки. Если документы разработки отсутствуют, неполны или устарели, руководители теряют важное средство для отслеживания и контроля проекта программного обеспечения;
- 4) документы образуют основу документации сопровождения программного обеспечения;
- 5) документы описывают историю разработки программного обеспечения.

Типовыми документами разработки являются:

- анализы осуществимости и исходные заявки;
- спецификации требований;
- спецификации функций;
- проектные спецификации, включая спецификации программ и данных;
- планы разработки;
- планы сборки и тестирования программного обеспечения;
- планы обеспечения качества, стандарты и графики;
- защитная и тестовая информация.

Структура процесса документирования в соответствии с СТБ ISO/IEC 12207



В соответствии с *ISO/IEC 12207* процесс документирования включает в себя:

- подготовку процесса документирования:
 - разработку и реализацию плана обозначения документов, выпускаемых в процессах жизненного цикла программных средств;
- проектирование и разработку документации:
 - проектирование документов согласно стандартам на документацию;
 - подтверждение источника и соответствия исходных материалов для документов;
 - проверку и редактирование документов согласно стандартам, утверждение компетентными лицами;
- выпуск документации:
 - издание и распространение документов в соответствии с планом;
 - управление документированием в соответствии с процессом управления конфигурацией;
- сопровождение документации:
 - внесение изменений в документацию согласно процессам сопровождения и управления конфигурацией.

В плане обозначения документов должны быть определены:

- заголовок или наименование;
- назначение;
- пользователи документа;
- процедуры и обязанности по подготовке исходных материалов, разработке, проверке, изменению, утверждению, выпуску, хранению, распространению, сопровождению и управлению конфигурацией документов.

Реализация процесса документирования в соответствии со стандартом ISO/IEC 15910:1999

В 1999 году введен в действие международный стандарт ISO/IEC 15910:1999 – Информационная технология – Процесс создания документации пользователя программного средства (аналогичный стандарт ГОСТ Р ИСО/МЭК 15910-2002 действует с 2003 г.).

Стандарт ISO/IEC 15910 определяет одну из реализаций процесса документирования, описанного в ISO/IEC 12207, и предоставляет пользователю метод применения данного процесса при создании конкретного программного средства. Стандарт содержит структуру комплексного плана разработки документации, однако не определяет состав требований к стилю оформления документов, устанавливая только их диапазоны.

Стандарт ISO/IEC 15910 должен использоваться при создании пользовательской документации всех видов:

- печатной документации (руководство пользователя, краткие справочные карты);
- диалоговой (оперативной) информации;
- справочного текста (Help);
- системы диалоговой документации.

Основные определения стандарта ISO/IEC 15910:1999

Стандарт *ISO/IEC 15910* использует следующие основные определения.

Аудитория (*audience*): категория пользователей, предъявляющих к документации одинаковые или аналогичные требования, определяющие содержание, структуру и назначение данной документации.

Документатор (*documenter*): сторона, создающая документацию.

Справочный текст (*help text*): текст, автоматически выбираемый в зависимости от контекста, в котором он вызывается, облегчающий и упрощающий при эксплуатации ПС поиск содержащихся в издании объектов. Справочный текст контекстно зависим.

Диалоговая документация (*on-line documentation*): информация, доступная пользователю при эксплуатации ПС, которая необязательно привязана к конкретному контексту.

Система диалоговой информации или справочная система (*on-line documentation system or help system*): часть программы или отдельная программа, запрашиваемая пользователем и позволяющая ему просматривать части диалоговой документации или справочного текста.

Продукт (*программный продукт, product*): полный набор компьютерных программ, процедур и соответствующих им документации и данных, предназначенный для поставки пользователю.

Тестирование на практичность (*usability testing*): формальный процесс оценки соответствия документации установленным требованиям.

4. Программная документация

Существует четыре основных типа документации на ПО:

- архитектурная/проектная – обзор программного обеспечения, включающий описание рабочей среды и принципов, которые должны быть использованы при создании ПО;
- техническая – документация на код, алгоритмы, интерфейсы, API;
- пользовательская – руководства для конечных пользователей, администраторов системы и другого персонала;
- маркетинговая.

Программную документацию по отношению к пользователю делят на:

- внутренняя (технологическая) – используется в процессе разработки ПО и недоступна пользователю;
- внешняя (пользовательская) – всевозможные руководства для пользователя.

Документы, описывающие процесс разработки программного обеспечения, определяют требования, которым должно удовлетворять программное обеспечение, определяют проект программного обеспечения, определяют, как его контролируют и как обеспечивают его качество. Документация разработки, также включает в себя подробное техническое описание программного обеспечения (программную логику, программные взаимосвязи, форматы и хранение данных и т. д.).

Стандарт документирования

ГОСТ при разработке программных продуктов дает возможность:

- унифицировать программные изделия для взаимного обмена и применения ранее разработанных программ в новых разработках;
- снизить трудоемкость и повысить эффективность разработки, сопровождения, изготовления и эксплуатации программных продуктов;
- автоматизировать изготовление и хранение программной документации.
- основу отечественной нормативной базы в области документирования ПО составляет **комплекс стандартов Единой системы программной документации (ЕСПД)**.
- стандарты ЕСПД в основном охватывают ту часть документации, которая создается в процессе разработки ПО, и связаны, по большей части, с документированием функциональных характеристик ПО.

Значение ЕСПД

- стандарты ЕСПД вносят элемент упорядочения в процесс документирования ПС;
- предусмотренный стандартами ЕСПД состав программных документов не является «жестким»: стандарты позволяют вносить в комплект документации на ПС дополнительные виды;
- стандарты ЕСПД позволяют мобильно изменять структуры и содержание установленных видов программной документации исходя из требований заказчика и пользователя.

Недостатки ЕСПД

- ориентацию на единственную, «каскадную» модель жизненного цикла ПО;
- отсутствие четких рекомендаций по документированию характеристик качества ПО;
- отсутствие системной увязки с другими действующими отечественными системами стандартов по ЖЦ и документированию продукции в целом;
- нечетко выраженный подход к документированию ПО как товарной продукции;
- отсутствие рекомендаций по самодокументированию ПО;
- отсутствие рекомендаций по составу, содержанию и оформлению перспективных документов на ПО, согласованных с рекомендациями международных и региональных стандартов.

Группы стандарта ЕСПД

Код группы	Наименование группы
0	Общие положения
1	Основополагающие стандарты
2	Правила выполнения документации разработки
3	Правила выполнения документации изготовления
4	Правила выполнения документации сопровождения
5	Правила выполнения эксплуатационной документации
6	Правила обращения программной документации

Перечень документов ЕСПД

- ГОСТ 19.001-77 ЕСПД. Общие положения.
- ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.
- ГОСТ 19.102-77 ЕСПД. Стадии разработки.
- ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов.
- ГОСТ 19.104-78 ЕСПД. Основные надписи.

- ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам.
- ГОСТ 19.106-78 ЕСПД. Требования к программным документам, выполненным печатным способом.
- ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.
- ГОСТ 19.202-78 ЕСПД. Спецификация. Требования к содержанию и оформлению.
- ГОСТ 19.301-79 ЕСПД. Порядок и методика испытаний
- ГОСТ 19.401-78 ЕСПД. Текст программы. Требования к содержанию и оформлению.
- ГОСТ 19.402-78 ЕСПД. Описание программы.
- ГОСТ 19.404-79 ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.
- ГОСТ 19.501-78 ЕСПД. Формуляр. Требования к содержанию и оформлению.
- ГОСТ 19.502-78 ЕСПД. Описание применения. Требования к содержанию и оформлению.
- ГОСТ 19.503-79 ЕСПД. Руководство системного программиста. Требования к содержанию и оформлению.
- ГОСТ 19.504-79 ЕСПД. Руководство программиста.
- ГОСТ 19.505-79 ЕСПД. Руководство оператора.
- ГОСТ 19.506-79 ЕСПД. Описание языка.
- ГОСТ 19.508-79 ЕСПД. Руководство по техническому обслуживанию. Требования к содержанию и оформлению.
- ГОСТ 19.604-78 ЕСПД. Правила внесения изменений в программные документы, выполняемые печатным способом.
- ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.
- ГОСТ 19.781-90. Обеспечение систем обработки информации программное.