

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ
«Международный Колледж Бизнеса и Дизайна»
(АНО ПОО «Международный Колледж Бизнеса и Дизайна»)**

УТВЕРЖДАЮ
Директор АНО ПОО «МКБид»
Н.Н.Репин

Н.Н.Репин 2023 г.

КОНСПЕКТ ЛЕКЦИЙ

по учебной дисциплине ОП 09 Стандартизация, сертификация и техническое
документоведение

программы подготовки специалистов среднего звена
по специальности: 09.02.07 «Информационные системы и программирование»

Тема: термины и определения стандартизации и сертификации. Цели и задачи стандартизации. Категории стандартов. Методика разработки и утверждения стандартов.

Цель: изучить основные понятия стандартизации

Мы поговорим о мониторинге и оценке ПО. Чем отличается и зачем нужно мониторить и

Современные контракты и планы на создание сложных программных средств (ПС) для информационных систем подготавливаются и оцениваются часто неквалифицированно, на основе неформализованных представлений заказчиков и разработчиков о требуемых функциях и их характеристиках качества.

Т.е. мы делаем, то что требует заказчик, а не следуем правилам, как нужно правильно проектировать и разрабатывать

Во многих случаях нужное качество программных средств зависит от интуиции, вкусов и квалификации их заказчиков и пользователей.

Значительные системные ошибки при определении требуемых характеристик качества, оценке трудоемкости, стоимости и длительности создания ПС являются достаточно массовыми и типичными.

Многие созданные программные продукты не способны полностью выполнять требуемые функциональные задачи с гарантированным качеством и их приходится долго и иногда безуспешно дорабатывать для достижения необходимого качества и надежности функционирования, затрачивая дополнительно большие средства и время.

В результате программные средства, *не соответствуют исходном и первоначальным требованиям заказчика к характеристикам качества, не укладываются в согласованные графики и бюджет разработки.*

Стандартизация — это процесс установления и применения стандартов, под которыми понимается «образец, эталон, модель, принимаемые за исходные для сопоставления с ними других подобных объектов».

Стандарт как нормативно-технический документ устанавливает комплекс норм, правил, требований к объекту стандартизации и утверждается компетентным органом.

Стандарт-это уже документ по стандартизации, в котором описаны нормы и правила и требования к продукту. Содержит в себе показатель изменчивости качества.

Применение стандартов способствует улучшению качества создаваемого изделия (в данном случае документа).

Документирование — это процесс создания и оформления документа. Государственный стандарт определяет документирование как «запись информации на различных носителях по установленным правилам»*.

СТАНДАРТИЗАЦИЯ. – это установление и применение правил с целью упорядочения деятельности в определённой области на пользу и при участии всех заинтересованных сторон, в частности, для достижения всеобщей оптимальной экономии при соблюдении условий эксплуатации и требований безопасности.

Стандартизация, основанная на объединённых достижениях науки, техники и передового опыта, определяет основу не только настоящего, но и будущего развития промышленности.

Из определения следует, что стандартизация – это плановая деятельность по установлению обязательных правил, норм и требований, выполнение которых обеспечивает экономически оптимальное качество продукции, повышение производительности общественного труда и эффективности использования материальных ценностей при соблюдении требований безопасности.

СТАНДАРТ.– нормативно-технический документ по стандартизации, устанавливающий комплекс норм, правил, требований к объекту стандартизации и утверждённый компетентным органом.

Стандарты содержат показатели, которые гарантируют возможность повышения качества продукции и экономичности её производства, а также повышения уровня её взаимозаменяемости.

ЦЕЛИ И ЗАДАЧИ СТАНДАРТИЗАЦИИ.

Главная цель Государственной системы стандартизации (ГСС) – с помощью стандартов, устанавливающих показатели, нормы и требования, соответствующие передовому уровню отечественной и зарубежной науки, техники и производства, содействовать обеспечению пропорционального развития всех отраслей страны.

Цели Следить за тем, чтобы все отрасли развивались одинаково не нарушая нормы требования и показатели стандарта, соответствовали государственному и мировому уровню..

задачи Госстандарта является:

- 1) разработка мер по повышению эффективности стандартизации
- 2) улучшение качества выпускаемой продукции и экономичности её производства путём внедрения систем стандартов при комплексной
- 3) развития межотраслевой унификации

4) создания общетехнических систем стандартов, обеспечения единства и достоверности измерений в стране и др.

КАТЕГОРИИ СТАНДАРТОВ

В зависимости от сферы действия ГСС (Государственной системы стандартизации)

предусматривает следующие категории стандартов:

- 1) государственные (ГОСТ),
- 2) отраслевые (ОСТ),
- 3) республиканские (РСТ)
- 4) стандарты предприятий (СТП).

1. Государственные стандарты обязательны для всех предприятий, организаций и учреждений страны в пределах сферы их действия.

Государственные стандарты устанавливают требования к продукции массового и крупносерийного производства широкого и межотраслевого производства, к изделиям, прошедшим государственную аттестацию, экспортным товарам; они устанавливают также общие нормы, термины и т. п.

2. Отраслевые стандарты используют все предприятия и организации данной отрасли (например, IT –чистота и правильность написания кода), а также другие предприятия и организации, разрабатывающие, изготавливающие и применяющие изделия, которые относятся к номенклатуре, закреплённой за соответствующим министерством.

Отраслевые стандарты устанавливают требования к продукции, не относящейся к объектам государственной стандартизации, к технологической оснастке, инструменту, специфическим для отрасли.

В сфере IT: есть протоколы передачи данных TCP-IP протоколы нижнего уровня для передачи информации, есть HTTP протокол передачи гипертекста.

Так же относятся стандарты для оборудования возьмем кабель Ethernet (10mb/s- скорость, кабель-толстый, витая пара, оптика)- стандарт 802.3

3. Республиканские стандарты обязательны для предприятий республиканского и местного подчинения данной республики независимо от их ведомственной принадлежности.

Республиканские стандарты обязательны для предприятий республиканского и местного подчинения данной республики независимо от их ведомственной принадлежности.

У нас есть свои стандарты и мы им следуем у себя в республике, не зависимо, что есть гос. Стандарт.

4. Стандарты предприятий (объединений) действуют только на предприятии, утвердившем данный стандарт.

Стандарты предприятий (объединений) распространяются на нормы, правила, методы, составные части изделий и другие объекты, имеющие применение только на данном предприятии; *Стандарты предприятий могут также устанавливать ограничения по применяемой номенклатуре деталей, составных частей, материалов, предусмотренные государственными, отраслевыми или республиканскими стандартами.*

МЕТОДИКА РАЗРАБОТКИ И УТВЕРЖДЕНИЯ СТАНДАРТОВ.

Целесообразность разработки каждого стандарта обосновывается потребностями народного хозяйства и ожидаемым техническим и экономическим эффектом. Для этого предварительно подбирают и анализируют литературные и производственные данные, устанавливают тенденции развития и перспективные потребности промышленности по стандартизуемым объектам или параметрам.

Т.е. стандарт разрабатывается тогда, когда изучены потребительские особенности, ожидаемые технические характеристики уже знаем тенденции развития в определенной сфере.

Например, для покупателя телевизора важны размеры экрана, четкость изображения, гарантийный срок, внешний вид и его ремонтпригодность, т.е. возможность быстрого обнаружения повреждений и замены неисправных элементов.

Для завода-изготовителя, кроме указанного, важное значение имеют совершенство конструкции и технологичность составных частей телевизора, определяющих трудоемкость и экономичность его производства

Обязательным этапом является анализ зарубежного опыта и достигнутого там уровня качественных показателей стандартизуемых объектов.

Номенклатура показателей качества должна быть достаточной, чтобы всесторонне и полно характеризовать изделие не только с точки зрения изготовителя, но и с точки зрения потребителя.

ГСС устанавливает шесть стадий разработки стандартов:

- 1) организация разработки стандарта, составление и утверждение технического задания;
- 2) разработка проекта стандарта и рассылка его на отзыв;
- 3) анализ отзывов и разработка окончательной редакции проекта стандарта;
- 4) подготовка, согласование и представление стандарта на утверждение;
- 5) рассмотрение, утверждение и регистрация стандарта;
- 6) издание стандарта и информации о нем.

Контрольные вопросы:

1. Что такое стандартизация?
2. Чем стандартизация отличается от стандарта?
3. Что следует из определения стандартизации? Дайте определение
4. Какая главная цель государственной системы стандартизации?
5. Назовите задачи госстандарта?
6. Назовите и опишите категории стандартов?
7. Назовите 6 стадий разработки стандартов?
8. Какая методика разработки и утверждения стандартов?

Список использованных источников:

- 1) Орлов, С.А. Технологии разработки программного обеспечения: учебник / С.А. Орлов. – СПб: Питер, 2002. – 464 с.
- 2) Липаев, В.В. Управление разработкой программных средств: Методы, стандарты, технология / В.В. Липаев. – М.: Финансы и статистика, 1993.
- 3) Липаев, В.В. Тестирование программ / В.В. Липаев. – М.: Радио и связь, 1986.
- 4) Липаев, В.В., Технология сборочного программирования / В.В. Липаев, Б.А. Позин, А.А. Штрик. – М.: Радио и связь, 1992.
- 5) Сертификация продукции. Международные стандарты и руководства ИСО/МЭК в области сертификации и управления качеством. – М.: Изд-во стандартов, 1990.
- 6) Лифиц, И.М. Стандартизация, метрология и сертификация /И.М. Лифиц. –М.: Юрайт-издат, 2004. – 335 с.
- 7) Сертификация сложных технических систем /Л.Н. Александровская [и др.]. – М.: Логос, 2001. – 312 с.
- 8) Якушев, А.И., Взаимозаменяемость, стандартизация и технические измерения / А.И Якушев, Л.Н. Воронцов, Н.М. Федотов. – М.: Машиностроение, 1986. – 352 с.

ЛЕКЦИЯ 3

Тема: жизненный цикл программных средств. Нормативные документы. Процессы ЖЦ в соответствии со стандартом ISO 12207

Цель: изучить основные процессы продиктованные стандартом ISO 12207

Программное средство – это набор компьютерных программ, процедур и связанных с ними документации и данных.

Это наш инструментарий с его мануалами(документацией)

Жизненный цикл (ЖЦ) ПС – это период времени, который начинается с момента принятия решения о необходимости создания ПС и заканчивается в момент его полного изъятия из эксплуатации.

Процесс – это совокупность взаимосвязанных действий, преобразующих некоторые входные данные в выходные.

Основной нормативный документ, регламентирующий состав процессов ЖЦ ПС, – международный стандарт ISO/IEC 12207. Стандарт определяет структуру ЖЦ, содержащую процессы, действия и задачи, выполняемые в процессе создания.

Стандарты для ПС:

1. ISO (International Organization of Standardization) – Международная организация по стандартизации
2. IEC (International Electro-technical Commission) – Международная организация по электротехнике
3. ISO/IEC 12207 – стандарт, принятый в 1995 году для разработки ПС.

В соответствии со стандартом ISO 12207 все процессы ЖЦ разделены на следующие:

1. Основные процессы;
2. Вспомогательные процессы, обеспечивающие выполнение основных процессов;
3. Организационные процессы.

1) К ОСНОВНЫМ ПРОЦЕССАМ ОТНОСЯТСЯ:

1. приобретение;

то когда мы захотели купить какую либо услугу

2. поставка;

поставщик услуг следит за работоспособностью своего продукта

3. разработка;

разработчик проектирует систему, выбирает средства для разработки, программирует и тестирует модули разработки

4. эксплуатация;

проводится тестирование общего ПО

5. сопровождение.

Все что касается разработки и поддержки ПО в работоспособном виде или ее усовершенствовании

1) Приобретение Действия заказчика, приобретающего ПС:

1.1 инициирование приобретения;

1.2 подготовка заявочных предложений;

1.3 подготовка и корректировка договора;

1.4 надзор за деятельностью поставщика;

1.5 приемка и завершение работы.

2) Поставка -Действия и задачи поставщика, который снабжает заказчика программным продуктом или услугой:

2.1 инициирование поставки;

2.2 подготовка ответа на заявочное предложение;

2.3 подготовка договора;

2.4 планирование;

2.5 выполнение и контроль;

2.6 проверка и оценка;

2.7 поставка и завершение работы.

3) Разработка -Действия и задачи, выполняемые разработчиком. Охватывает работы по созданию ПС и его компонентов в соответствии с заданными средствами, включая оформление проекта и эксплуатационной документации, подготовку материалов, необходимых для проверки работоспособности и соответствующего качества программных продуктов, материалов, необходимых для организации обучения персонала и т.д.

Разработка включает следующие действия:

- 3.1 подготовительную работу;
- 3.2 анализ требований к системе;
- 3.3 проектирование архитектуры системы;
- 3.4 анализ требований к ПС;
- 3.5 проектирование архитектуры ПС;
- 3.6 детальное проектирование ПС;
- 3.7 кодирование и тестирование ПС;
- 3.8 интеграцию ПС;
- 3.9 квалификационное тестирование;
- 3.10 интеграцию системы;
- 3.11 установку ПС;
- 3.12 приемку ПС.

4) **Эксплуатация** -Охватывает действия и задачи оператора организации, эксплуатирующей ПС.

Процесс эксплуатации включает в себя следующие действия:

- 4.1 подготовительная работа;
- 4.2 эксплуатационное тестирование;
- 4.3 эксплуатация системы;
- 4.4 поддержка пользователей.

5) **Сопровождение**- Предусматривает действия, выполняемые сопровождающей организацией. Это внесение изменений в ПС в целях исправления ошибок, повышения производительности или адаптации к изменившимся условиям работы. Изменения, вносимые в ПС, не должны нарушать его целостность.

Процесс сопровождения включает в себя следующие действия:

- 5.1 подготовительная работа;
- 5.2 анализ проблем и запросов на модификацию ПС;
- 5.3 модификация ПС;
- 5.4 проверка и приемка;
- 5.5 перенос ПС в другую среду;
- 5.6 снятие ПС с эксплуатации.

2) **К ВСПОМОГАТЕЛЬНЫМ ПРОЦЕССАМ ОТНОСЯТСЯ:**

- 1) документирование;
- 2) управление конфигурацией;
- 3) обеспечение качества;
- 4) верификация;
- 5) аттестация;
- 6) совместная оценка;

- 7) аудит;
- 8) разрешение проблем.

3) **ОРГАНИЗАЦИОННЫЕ ПРОЦЕССЫ:**

- 1) управление;
- 2) создание инфраструктуры;
- 3) усовершенствование;
- 4) обучение.

СТАДИИ ЖИЗНЕННОГО ЦИКЛА ИНФОРМАЦИОННОЙ СИСТЕМЫ:

1) Предпроектное обследование:

1. сбор материалов для проектирования, при этом выделяют формулирование требований, с изучения объекта автоматизации, даются предварительные выводы пред проектного варианта ИС;
2. анализ материалов и разработка документации, обязательно дается технико экономическое обоснование с техническим заданием на проектирование ИС.

2) Проектирование:

ISO 9001:2000 – устанавливает детальные требования для систем управления качеством, достаточные в случае необходимости продемонстрировать способность предприятия обеспечить соответствие качества продукции и услуг требованиям заказчика;

2.1 предварительное проектирование:

1. выбор проектных решений по аспектам разработки ИС;
2. описание реальных компонент ИС;
3. оформление и утверждение технического проекта (ТП).

2.2 детальное проектирование:

1. выбор или разработка математических методов или алгоритмов программ;
2. корректировка структур БД;
3. создание документации на доставку и установку программных продуктов;
4. выбор комплекса технических средств с документацией на ее установку.

2.3 разработка техно-рабочего проекта ИС (ТРП).

2.4 разработка методологии реализации функций управления с помощью ИС и описанием регламента действий аппарата управления.

3) Разработка ИС:

1. получение и установка технических и программных средств;
2. тестирование и доводка программного комплекса;
3. разработка инструкций по эксплуатации программно-технических средств.

4) Ввод ИС в эксплуатацию:

1. ввод технических средств;
2. ввод программных средств;
3. обучение и сертификация персонала;
4. опытная эксплуатация;
5. сдача и подписание актов приемки-сдачи работ.

ISO 9004:2000 – содержит руководство по внедрению и применению широко развитой системы управления качеством, чтобы достичь постоянного улучшения деловой деятельности и результатов предприятия.

5) Эксплуатация ИС:

1. повседневная эксплуатация;
2. общее сопровождение всего проекта.

После того ,как мы сдали наше ПО в эксплуатацию идет сопровождение и поддержка

Контрольные вопросы:

9. Что такое ЖЦ ПО?
10. Что такое программное средство?
11. Что такое процесс?
12. Стандарты для ПО?
13. В соответствии со стандартным процессом ЖЦ разделены
14. Что относится к основным процессам?
15. Что относится к вспомогательным процессам?
16. Что относится к организационным процессам?
17. Что такое сопровождение?

Список использованных источников:

- 9) Орлов, С.А. Технологии разработки программного обеспечения: учебник / С.А. Орлов. – СПб: Питер, 2002. – 464 с.
- 10) Липаев, В.В. Управление разработкой программных средств: Методы, стандарты, технология / В.В. Липаев. – М.: Финансы и статистика, 1993.
- 11) Липаев, В.В. Тестирование программ / В.В. Липаев. – М.: Радио и связь, 1986.
- 12) Липаев, В.В., Технология сборочного программирования / В.В. Липаев, Б.А. Позин, А.А. Штрик. – М.: Радио и связь, 1992.
- 13) Сертификация продукции. Международные стандарты и руководства ИСО/МЭК в области сертификации и управления качеством. – М.: Изд-во стандартов, 1990.
- 14) Лифиц, И.М. Стандартизация, метрология и сертификация / И.М. Лифиц. – М.: Юрайт-издат, 2004. – 335 с.
- 15) Сертификация сложных технических систем /Л.Н. Александровская [и др.]. – М.: Логос, 2001. – 312 с.
- 16) Якушев, А.И., Взаимозаменяемость, стандартизация и технические измерения / А.И. Якушев, Л.Н. Воронцов, Н.М. Федотов. – М.: Машиностроение, 1986. – 352 с.

ЛЕКЦИЯ 4

Тема: стандарты регламентирующих ЖЦ ПО/ Основные процессы ЖЦ ISO/IEC 12207 Стандарты жизненного цикла ПО

Информационные системы должны удовлетворять интересам бизнеса, а также быть легко модифицируемыми и недорогими. Плохо спроектированная система, в конечном счете, требует больших затрат и времени для ее содержания и обновления.

Одним из базовых понятий методологии проектирования ИС является понятие жизненного цикла ее программного обеспечения (ЖЦ ПО).

Жизненный цикл ПО – это непрерывный процесс, который начинается с момента принятия решения о необходимости его создания и заканчивается в момент его полного изъятия из эксплуатации.

Существует целый ряд стандартов, регламентирующих ЖЦ ПО, а в некоторых случаях и процессы разработки.

стандарты регламентирующих ЖЦ ПО можно выделить следующие:

• ГОСТ 34.601-90 - распространяется на автоматизированные системы и устанавливает стадии и этапы их создания. Кроме того, в стандарте содержится описание содержания работ на каждом этапе. Стадии и этапы работы, закрепленные в стандарте, в большей степени соответствуют *каскадной модели жизненного цикла*.

• ISO/IEC 12207:1995 - стандарт на процессы и организацию жизненного цикла. Распространяется на все виды заказного ПО. *Стандарт не содержит описания фаз, стадий и этапов.*

• Custom Development Method (методика Oracle) по разработке прикладных информационных систем - технологический материал, детализированный до уровня заготовок проектных документов, рассчитанных на использование в проектах с

применением Oracle. Применяется CDM для классической модели ЖЦ (предусмотрены все работы/задачи и этапы), а также для технологий "быстрой разработки" (Fast Track) или "облегченного подхода", рекомендуемых в случае малых проектов.

• **Rational Unified Process (RUP)** предлагает итеративную модель разработки, включающую четыре фазы: начало, исследование, построение и внедрение. Каждая фаза может быть разбита на этапы (итерации), в результате которых выпускается версия для внутреннего или внешнего использования. Прохождение через четыре основные фазы называется циклом разработки, каждый цикл завершается генерацией версии системы. Если после этого работа над проектом не прекращается, то полученный продукт продолжает развиваться и снова минует те же фазы. Суть работы в рамках RUP - это создание и сопровождение моделей на базе UML.

• **Microsoft Solution Framework (MSF)** сходна с RUP, так же включает четыре фазы: анализ, проектирование, разработка, стабилизация, является итерационной, предполагает использование объектно-ориентированного моделирования. MSF в сравнении с RUP в большей степени ориентирована на разработку бизнес-приложений.

• **Extreme Programming (XP)**. Экстремальное программирование (самая новая среди рассматриваемых методологий) сформировалось в 1996 году. В основе

Основные процессы жизненного цикла ISO/IEC 12207 делятся на 3 группы:

В соответствии с базовым международным стандартом ISO/IEC 12207 *все процессы ЖЦ ПО делятся на три группы:*

1. Основные процессы:

- 1.1 приобретение;
- 1.2 поставка;
- 1.3 разработка;
- 1.4 эксплуатация;
- 1.5 сопровождение.

2. Вспомогательные процессы:

Вспомогательные процессы предназначены для поддержки выполнения основных процессов, обеспечения качества проекта, организации верификации, проверки и тестирования ПО.

- 2.1 документирование;
- 2.2 управление конфигурацией;
- 2.3 обеспечение качества;
- 2.4 разрешение проблем;

2.5 аудит;

2.6 аттестация;

2.7 совместная оценка;

2.8 верификация.

3. **Организационные процессы:**

Организационные процессы определяют действия и задачи, выполняемые как заказчиком, так и разработчиком проекта для управления своими процессами

3.1 создание инфраструктуры;

3.2 управление;

3.3 обучение;

3.4 усовершенствование.

В таблице 1 приведены ориентировочные описания основных процессов ЖЦ.

Для поддержки практического применения стандарта ISO/IEC 12207 разработан ряд технологических документов:

• **Руководство для ISO/IEC 12207 (ISO/IEC TR 15271:1998 Information technology - Guide for ISO/IEC 12207)**

ISO – International Organization of Standardization – Международная организация по стандартизации;

IEC – International Electrotechnical Commission – Международная комиссия по электротехнике) определяет структуру ЖЦ, содержащую процессы, действия и задачи, которые должны быть выполнены во время создания ПО

• **Руководство по применению ISO/IEC 12207 к управлению проектами (ISO/IEC TR 16326:1999 Software engineering - Guide for the application of ISO/IEC 12207 to project management).**

Таблица 1. Содержание основных процессов ЖЦ ПО ИС (ISO/IEC 12207)

Процесс (исполнитель процесса)	Действия	Вход	Результат
Приобретение (заказчик)	Инициирование	Решение о начале работ по внедрению ИС	Технико-экономическое обоснование внедрения ИС
	Подготовка заявочных предложений	Результаты обследования деятельности заказчика	Техническое задание на ИС
	Подготовка договора	Результаты анализа рынка ИС/ тендера	Договор на поставку/ разработку
	Контроль деятельности поставщика	План поставки/ разработки	Акты приемки этапов работы
	Приемка ИС	Комплексный тест ИС	Акт приемно-сдаточных испытаний
Процесс (исполнитель процесса)	Действия	Вход	Результат
	Инициирование	Техническое задание на ИС	Решение об участии в разработке

Поставка (разработчик ИС)	Подготовка заявочных предложений	Решение	Коммерческие предложения/ конкурсная заявка
	Подготовка договора	руководства об участии в разработке	Договор на поставку/ разработку
	Контроль деятельности поставщика	Результаты тендера	План управления проектом
	Приемка ИС	Техническое задание на ИС	Реализация/ корректировка
		План управления проектом	
Процесс (исполнитель процесса)	Действия	Вход	Результат
Разработка (разработчик ИС)	Подготовка	Техническое задание на ИС	Используемая модель ЖЦ, стандарты разработки
	Анализ требований к ИС	Техническое задание на ИС, модель ЖЦ	План работ
	Проектирование архитектуры ИС	Подсистемы ИС	Состав подсистем, компоненты оборудования
	Разработка требований к ПО	Спецификации требования к компонентам ПО	Спецификации требования к компонентам ПО
	Проектирование архитектуры ПО	Архитектура ПО	Состав компонентов ПО, интерфейсы с БД, план интеграции ПО
	Детальное проектирование ПО	Материалы детального проектирования ПО	Проект БД, спецификации интерфейсов между компонентами ПО, требования к тестам
	Кодирование и тестирование ПО	План интеграции ПО, тесты	Тексты модулей ПО, акты автономного тестирования
	Интеграция ПО и квалификационное тестирование ПО	Архитектура ИС, ПО, документация на ИС, тесты	Оценка соответствия комплекса ПО требованиям ТЗ
	Интеграция ИС и квалификационное тестирование ИС		Оценка соответствия ПО, БД, технического комплекса и комплекта документации требованиям ТЗ

В общем ЖЦ ПО ИС гоаорит (ISO/IEC 12207),, что должна быть:

1. **Эксплуатация** включает в себя работы по внедрению компонентов ПО в эксплуатацию, в том числе конфигурирование базы данных и рабочих мест пользователей, обеспечение эксплуатационной документацией, проведение обучения персонала и т.д., и непосредственно эксплуатацию, в том числе локализацию проблем и устранение причин их возникновения.

2. **Управление проектом** связано с вопросами планирования и организации работ, создания коллективов разработчиков и контроля за сроками и качеством выполняемых работ. Техническое и организационное обеспечение проекта включает выбор методов и инструментальных средств для реализации проекта, определение методов описания промежуточных состояний разработки, разработку методов и средств испытаний ПО.

3. **Верификация** – это процесс определения того, отвечает ли текущее состояние разработки, достигнутое на данном этапе, требованиям этого этапа. Проверка позволяет оценить соответствие параметров разработки исходным требованиям. Проверка частично совпадает с тестированием, которое связано с идентификацией различий между действительными и ожидаемыми результатами и оценкой соответствия характеристик ПО исходным требованиям.

4. Управление конфигурацией является одним из вспомогательных процессов, поддерживающих основные процессы жизненного цикла ПО, *прежде всего процессы разработки и сопровождения ПО*. При создании проектов сложных ИС, состоящих из многих компонентов, каждый из которых может иметь разновидности или версии, возникают проблемы учета их связей и функций, создания унифицированной структуры и обеспечения развития всей системы. Управление конфигурацией позволяет организовать внесение изменений в ПО на всех стадиях ЖЦ. Общие принципы и рекомендации конфигурационного учета, планирования и управления конфигурациями ПО отражены в проекте стандарта **ISO 12207-2**.

Каждый процесс характеризуется определенными задачами и методами их решения, исходными данными, полученными на предыдущем этапе, и результатами. Результатами анализа, в частности, являются функциональные модели, информационные модели и соответствующие им диаграммы. ЖЦ ПО носит итерационный характер: результаты очередного этапа часто вызывают изменения в проектных решениях, выработанных на более ранних этапах.

Контрольные вопросы:

1. Что такое модель ЖЦ ПО?
2. Назовите все стандарты регламентирующих ЖЦ ПО?
3. Что за стандарт *XP*?
4. Расскажите в чем суть стандарта *MSF*?
5. Расскажите в чем суть стандарта *RUP*?
6. Расскажите в чем суть стандарта методики Oracle?
7. Расскажите в чем суть стандарта *ISO/IEC*?
8. *ГОСТ* что такое?
9. Какие стандарты нужны для практического применения?
10. Назовите 3 основных процесса ЖЦ? Дайте оному определение.

ЛЕКЦИЯ 5

Тема: Модели ЖЦ в соответствии со стандартом ISO 12207 и 9004:2000. Каскадная модель.

Цель: изучить основные модели продиктованные стандартом ISO 12207. ISO 9004:2000

Модели жизненного цикла ПС

Под моделью ЖЦ ПС понимается структура, определяющая последовательность выполнения и взаимосвязь процессов, действий и задач на протяжении ЖЦ.

соответствуют, регламентированным и детализированным в стандартах ISO 9001:2000, ISO 12207 основных компонентах профиля стандартов жизненного цикла сложных ПС.

Модель ЖЦ продиктована стандартом ISO 12207 и зависит от:

1. спецификации;
2. масштабов;
3. условий.

Модель ЖЦ определяет характер процессов его создания, который представляет собой совокупность упорядоченных во времени, взаимосвязанных и объединенных в стадии работ, выполнение которых необходимо и достаточно для создания ПС, соответствующего заданным требованиям.

Модель ЖЦ определяет все стадии от идеи до поддержки и снятия с эксплуатации ПП, описывает какие действия должны происходить на каждом этапе, какой документацией должно сопровождаться разработка.

Выбор модели жизненного цикла зависит от специфики, масштаба, сложности проекта и набора условий, в которых АИС создается и функционирует.

Модель ЖЦ АИС включает:

- стадии;
- результаты выполнения работ на каждой стадии;
- ключевые события или точки завершения работ и принятия решений.

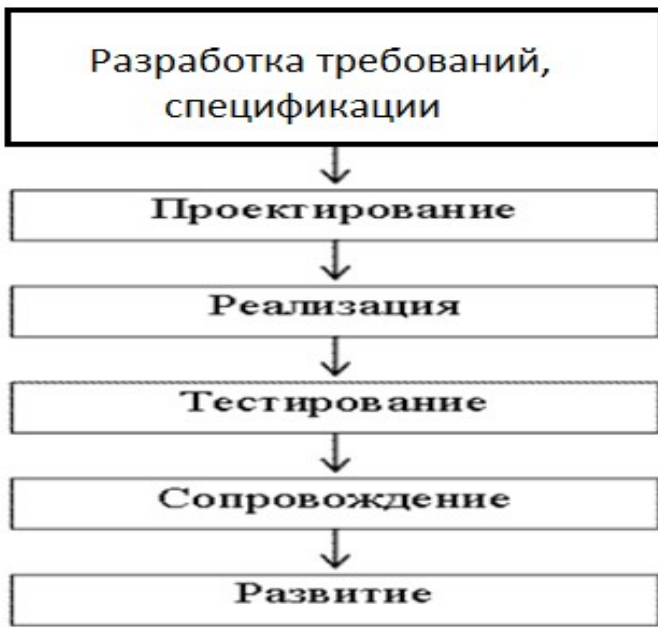
В соответствии с известными моделями ЖЦ ПО определяют модели ЖЦ АИС — каскадную, итерационную, спиральную.

КАСКАДНАЯ МОДЕЛЬ ЖЦ

В модели водопада, называемой также "каскадная модель жизненного цикла" или "каскадная модель жизненного цикла с обратными связями", сопровождение ПО выделяется в отдельную фазу жизненного цикла.

Каскадная стратегия (однократный проход, водопадная или классическая модель) подразумевает линейную последовательность выполнения стадий создания информационной системы (рис.3.1). Другими словами, переход с одной стадии на следующую происходит только после того, как будет полностью завершена работа на текущей.

Данная модель применяется при разработке информационных систем, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования.

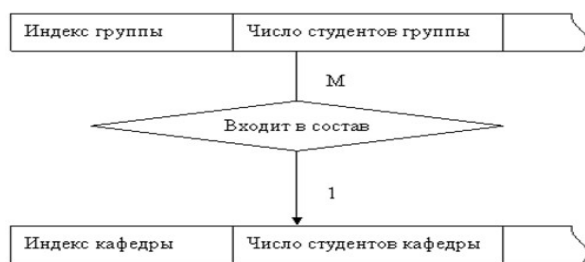


Этап	Результат
Выявление информационных потребностей конечных пользователей (предпроектное обследование, разработка ТЗ, частных ТЗ)	
Концептуальные проект (эскизное проектирование)	
Архитектура системы (техническое проектирование)	
Логическое проектирование (техническое проектирование)	
Комплексная отладка(разработка рабочей документации)	
сопровождение	



1. Функциональный граф ПО - граф, узлы которого обозначают данные и процессы

будущей системы. Дуги используются для обозначения входных/выходных данных для процесса



2.1 Эскизное проектирование- Для реализации концептуальной модели проектировщик вынужден выделять из функционального графа данные и строить для них схему БД, а также выделять процессы и разрабатывать для них спецификацию и кодировать. Это является источником большинства

ошибок проектирования. Данные функционального графа структурируются в виде инфологической схемы БД. Наши ER-модели

2.2. Спецификация процессов - входные и выходные данные процессов, а также алгоритмическая связь между ними. Для описания используют UML

3. Выбор архитектуры

- выбор модели доступа к данным (файл-сервер, сервер-БД, сервер-приложение, доступ к данным по Internet/Intarnet)
- выбор комплекса технических средств (выбор «железа»)
- выбор общесистемных пакетов
- выбор способа тиражирования данных

4. Логическое проектирование Выполняется отражение концептуального проекта в СУБД-ориентированную среду с помощью выбранных оболочек программирования. Сущности преобразуются в таблицы, а на основе спецификации задач разрабатываются тексты программ

5. Отладка Результаты проектирования БД и приложений объединяются. В итоге разрабатывается пилотный проект системы

6. Сопровождение Выявление ошибок и их устранение, модернизация.

Контрольные вопросы:

11. Что такое модель ЖЦ АИС?
12. От чего зависит выбор модели ЖЦ?
13. Что включает в себя каждый этап ЖЦ?
14. Что такое каскадная стратегия?
15. Опишите этапы каскадной стратегии
16. На какие две фазы делится каскадная стратегия?
17. Что такое эскизное проектирование?

ЛЕКЦИЯ 6

Тема: Спиральная модель в соответствии со стандартом ISO 12207. Процесс разработки, документирования АИС по спиральной схеме. Выбор архитектуры подсистем.

Модели жизненного цикла информационной системы:

1. каскадная модель 1970-80гг
2. инкрементная модель, RAD
3. спиральная модель 1986г

Следует отметить, что большинство спиральных моделей было разработано ранее принятия

международного стандарта ISO/IEC 12207:1995 [3]. В связи с этим необходимо выполнять адаптацию этих моделей с учетом положений действующих национальных аналогов данного стандарта [9] и его новой редакции ISO/IEC 12207:2008 [4]

Спиральная стратегия (эволюционная или итерационная модель) подразумевает разработку в виде последовательности версий, но в начале проекта определены не все требования. Требования уточняются в результате разработки версий.

Спиральная модель представляет шаблон процесса разработки ПО, который сочетает идеи итеративной и каскадной моделей.

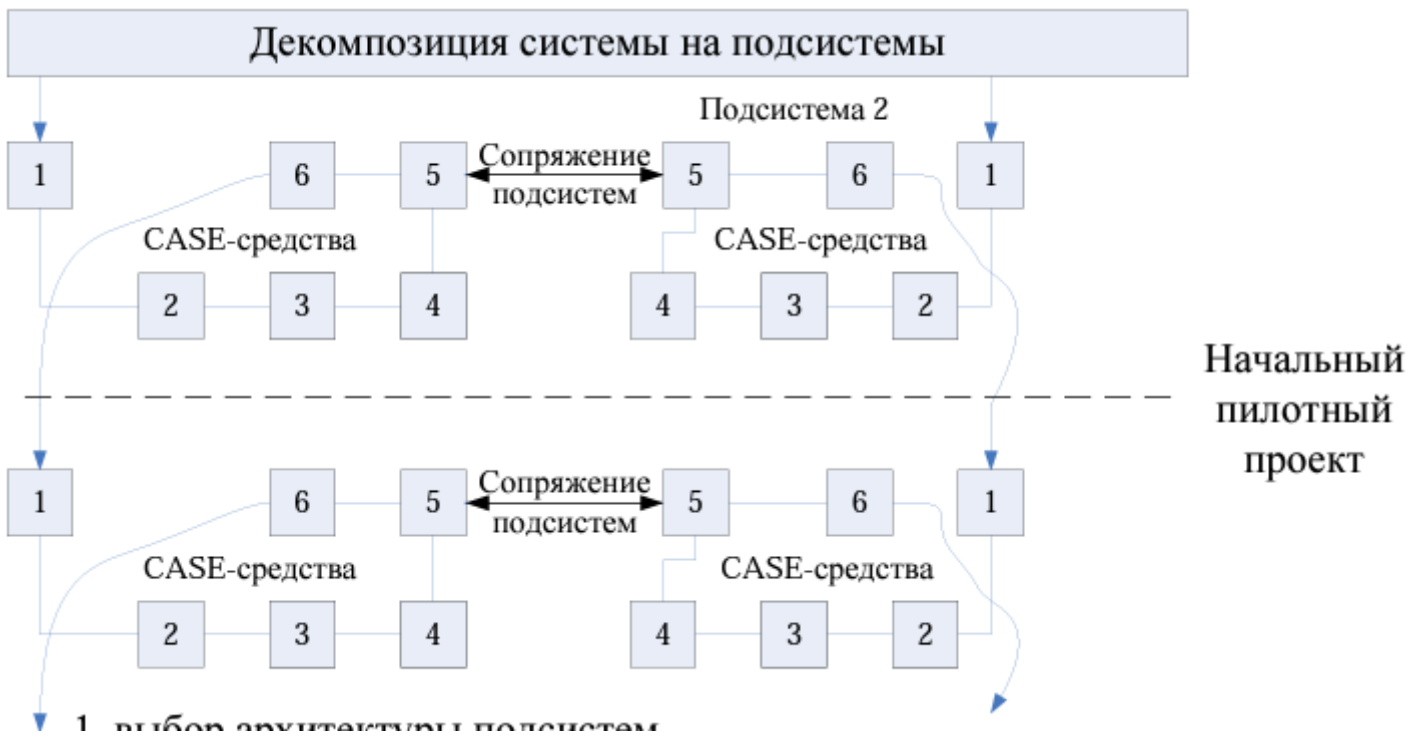


Рис. 3.3. Спиральная стратегия

Анализ рисков (анализ вероятности того, что произойдут определенные нежелательные события и отрицательно повлияют на достижение целей проекта)

Главная особенность спиральной модели – концентрация на возможных рисках. Для их оценки даже выделена соответствующая стадия.

Основная проблема спирального цикла – определение момента перехода на следующий этап. Для ее решения необходимо ввести временные ограничения на каждый из этапов жизненного цикла. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена. План составляется на основе статистических данных, полученных в предыдущих проектах, и личного опыта разработчиков



выбор архитектуры подсистем зависит :

1. выбор архитектуры подсистемы
2. выявление информационных потребностей конечных пользователей системы
3. концептуальное проектирование
4. логическое проектирование
5. отладка подсистем
6. сопровождение подсистемы

Начальный пилотный проект . Каждый этап реализуется CASE-средствами.

это наши бизнес модели, ГЪД диаграммы, блок схемы

Содержание этапов совпадает с аналогичными в каскадной модели, но в отличие от нее, этапы реализуются с помощью CASE-средств – использование этих средств позволяет существенно снизить время реализации витка спирали проектирования подсистем (в этом состоит основное преимущество спиральной модели), но это профессиональные средства, непредназначенные для конечных пользователей.

С помощью CASE-средств можно быстро сгенерировать проект хотя бы на уровне экранных форм и показать его конечному пользователю, который высказывает свои предложения и замечания, и на следующем витке реализуются они. Когда спецификация на уровне экранных форм будет согласована, то проектировщик может начинать детальную реализацию. Описанная схема разработки называют визуальным проектированием. Витков может быть много, разделение на этапы условно. Работы могут выполняться параллельно или в комплексной итерации на любом витке спирали.

Основные механизмы используемые подсистемами при документировании:

1. Применение общих справочников системы.
2. Глобальные настройки системы и индивидуальные настройки пользователя.
3. Учет часовых поясов.
4. Настройка автоматических уведомлений и назначений.
5. Диагностические опросники.
6. Динамические приоритеты.
7. Двухуровневый каталог услуг.
8. Механизм расчета параметров уровня сервиса.
9. Сценарии и правила обработки.
10. Возможность использования шаблонов документов.
11. Процедуры согласования.
12. Механизм замещения отсутствующих сотрудников.
13. Цветовая маркировка.
14. Настройка матрицы переходов состояний документов.
15. Хранение файлов в информационных объектах системы.
16. Работа с метриками и показателями процессов.
17. История обработки документа.
18. Интеграция с другими подсистемами и системами

Контрольные вопросы:

1. суть спиральной модели
2. что такое спиральная модель ПП?
3. Основные механизмы подсистем при документировании?
4. От чего зависит выбор архитектуры подсистемы?
5. Основная проблема спирального цикла?
6. Для чего используют case-средства?

ЛЕКЦИЯ 7

Тема: формирование требований к характеристикам и качеству ПП. Внешнее проектирование.

Цель: изучить внешнее проектирование и понять как устанавливаются требования к разработке

В данной лекции мы поговорим о том какие цели, задачи заказчик должен ставить к проекту, что он должен определить для начала работ по проекту.

Формирование требований к функционированию программных продуктов:

1. - системная эффективность целевого применения программных продуктов определяется степенью удовлетворения потребностей заказчика и пользователей;

Нравится заказчику – значит эффективно в применении на производстве

2. - каждое требование должно отражать отдельно распознаваемую, измеряемую сущность программного продукта;

Требование минималистичный стиль- каждый элемент дизайна должен выполняться в данном стиле. Есть стандарт минимализма и по его пунктам сравниваем насколько хорошо выполнено требование.

3. - для контроля качества и корректности на значимость задач, следует сопоставлять конкретные требования и сформулированные цели разработки программного продукта;

Контроль Качество осуществляется по средствам контроля целей, цель- поставлена, цель- выполнена

4. - для обеспечения качества программного продукта необходим набор требуемых характеристик, свойств, их мер и значений качества для определенных потребителей программного продукта;

Обеспечение Качества мы сравниваем меры и характеристики свойства объектов со значениями требований

5. следует учитывать, что требования обычно иницируются заказчиком с объемом функций, превышающим тот, который можно реализовать при выделенных ресурсах;

т.е. заказчик хочет за 100\$ полноценный интернет магазин, за 3 дня с расширенным функционалом, а это не соответствует действительности.

6. - ограниченные ресурсы для реализации требований функциональной пригодности, могут негативно отражаться на конструктивных характеристиках продукта.

2 разработчика никак не смогут выполнить заказ за 3 дня, качественно и со всем функционалом.

Анализ и разработка требований к ПС(роль заказчика):

Здесь мы определяемся, какую роль заказчик будет выполнять в проекте

1. проекты, управляемые пользователем;

заказчик все стадии ЖЦ проекта распределяет и координирует ресурсами сам.

2. проекты, утверждаемые пользователем;

заказчик, только документально либо в диалоговой переписке утверждает либо отклоняет решения принятые тимлидером

3. проекты, не зависимые от пользователя.

Все решения принимает руководитель проекта, заказчик только говорит, какие ф-и он хочет видеть в своем ПО.

В процессе разработки требований необходимо решить следующие задачи:

1. Выявить наличие информации, необходимой для выполнения планируемых функций.

Какая информация нам нужна для разработки от заказчика и отдельных отделов разработки

2. Определить трудоемкость и стоимость предстоящей работы.

Рассчитать стоимость работ и сколько понадобится кадровых единиц для выполнения работ по проекту

3. Обеспечить полноту и точность определения функций, подлежащих выполнению ПС, и их взаимосвязь.

Документируем, какие ф-и на выходе должен выполнять наш продукт, как между собой они будут связаны. *Печать документ не сможет осуществиться, пока не появится сам документ.*

4. Выявить пространственно-временные ограничения, налагаемые на систему, а также средства системы, которые в будущем могут претерпеть изменения.

Ставим временные рамки для эксплуатации нашего продукта, указываем когда возможно понадобится обновления или переход к примеру с РНР 7.0 на РНР 7.2

Результатом по выработке требований является соответствующий документ, который должен быть:

1. достаточным для идентификации целей ПС, его среды, преимуществ и недостатков ПС для пользователя, состава и конфигурации ресурсов для его работы;

изучая документ по выработке требований, мы должны точно знать следующее:

-сформулированы точно цели проекта

-выбрана среда для разработки и четко сформулирована область применения ПС

-какие достоинства и недостатки будут при использовании системы

-определена команда для разработки, ее состав и количество участвующих

2. достаточно полным, чтобы в последующем при разработке исключить серьезные модификации к пересмотру требований;

изучая документ по выработке требований, он настолько должен быть полно описан, *чтобы в середине разработки нам не потребовалось отклоняться от плана за планируемых действий или срочно модифицировать продукт, если вышло какое либо обновление.*

Проект должен длительный срок самостоятельно работать без существенных изменений.

3. достаточным для просмотра и утверждения администрацией на основе его реализованности в соответствии с выбранными критериями.

Высшее руководство, когда прочтет документ, должен понять и сравнить реализовались его требования или нет.

Определение целей создания ПС

Цели проекта – это цели, которые должны быть достигнуты в процессе проектирования.

Это то исходя из чего мы решили вообще строить свое проект. Вообще какие цели должны быть достигнуты при проектировании ПП.

Например: повышение эффективности работы, за счёт автоматизации внутренних процессов производства печати документа.

В общем случае цели могут быть сгруппированы в 10 самостоятельных категорий:

ISO 9126

В каждой из подгрупп мы описываем свои требования, которые мы хотим достичь.

1. Универсальность.

(насколько универсально наше ПО оно годится и для печати фото и для печати документов)

2. Человеческие факторы.

(должен влиять чел. Фактор на работу системы или это должен быть искусственный интеллект)

3. Адаптируемость.

(наше ПО будет работать и на ПК и на смартфонах)

4. Сопровождаемость.

(как долго наша фирма будет поддерживать ПО, какой срок, какие услуги предоставлять)

5. Надежность. свойства комплекса программ обеспечивать достаточно низкую вероятность потери работоспособности отказа, в процессе функционирования программного продукта в реальном времени. Основные атрибуты надежности могут быть объективно измерены и сопоставлены с требованиями

(дает верный результат по заданным критериям, нет скрытых ошибок(битые ссылки или есть в макете функционал, а на программном уровне не реализован))

6. Безопасность. ИЕС 61508:3 комплексов программ характеризуется величиной ущерба, возможного при проявлении дестабилизирующих факторов и реализации конкретных

угроз – рисков, а также средним временем между проявлениями непредумышленных угроз, нарушающих безопасность.

(похожа на надежность, но учитываются в безопасности только те угрозы, которые несут материальные потери)

Случайные катастрофический отказ в надежности отразился на безопасности, что в данный момент мы потеряли потенциального зарегистрированного клиента.

7. Документация.

(полнота документации и на каких этапах и каким стандартам должна придерживаться)

8. Стоимость.

(должны быть указаны расценки на выполнение функционала, чтобы заказчик смог переделится и с его бюджетом и с тем функционалом, который может позволить его бюджет)

9. Календарный план.

(устанавливаем сроки сдачи нам результатов по Проекту)

10. Производительность (эффективность).

(насколько должна повыситься наша производительность сайта к примеру, измеряем с помощью goole speed)

Сформулированные цели ПС, рассматриваемые с точки зрения пользователя, обычно включают следующую информацию:

1. Краткое описание.
2. Определение пользователя.
3. Подробное описание функциональных задач.
4. Документация.
5. Эффективность.
6. Совместимость.
7. Конфигурация.
8. Безопасность. IEC 61508:3
9. Обслуживание.
10. Установка.
11. Надежность.

Внешнее проектирование – процесс описания планируемого поведения разработанного ПС с точки зрения потенциального пользователя.

Внешнее проектирование предназначено для пользователя, что он в первую очередь ожидает от проекта, что хочет получить на выходе.

Описывается, как по его словам должен работать проект. Зашел нажал кнопочку и что в результате должно появиться.

К примеру: зашел на сайт сначала, должно появиться окно регистрации/авторизации далее главная страница с информацией сайта и главным меню...

Целью этого процесса является конкретизация внешних взаимодействий будущего ПС без детализации внутреннего устройства.

Получить детальную информацию по будущему функционалу

Внешний проект представляет собой внешние спецификации ПС, предназначенные для каждой группы специалистов: пользователей и разработчиков.

Для пользователей это будет пользовательская документация, для разработчиков это будет техническая документация.

Предварительный внешний проект содержит основные компоненты разрабатываемого ПС.

Т.е из каких главных модулей будет состоять.

Детальный внешний проект каждой функции пользователя включает:

1. описание входных данных;

это текст, картинки, формулы, числовые данные

2. описание выходных данных;

статистику по какому-либо признаку или шаблон заполненного документа для печати

3. преобразование системы;

расширяемость модулей системы будет возможна или нет, при дополнительном функционале, нам нужно будет нужно переписывать ПО или просто дописать один модуль и подключить его.

4. характеристика надежности;

случайные катастрофические отказы системы, отказы ситуаций, прерывания цикла работоспособности на время, автоматическое обслуживание опасных отказов(рестарт),

5. эффективность; ISO 9126 :

5.1 временная время отклика программы

5.2 пропускная способность число заданий которые может реализовать на данном ЭВМ в заданном интервале времени

5.3 используемость ресурсов степень загрузки доступных вычислительных ресурсов в течение заданного времени при выполнении функций комплекса программ в установленных условиях

6. замечания по программированию

читабельность кода, стандарты написания кода

Контрольные вопросы:

7. перечислите формирование требований к ПП?
8. Анализ и разработка требований к ПС заказчиком?
9. Какие задачи нужно решить в процессе разработки требований?
10. Что должен одержать в себе документ по выработке т требований?
11. Что такое цель проекта?
12. Назовите 10 категорий целей проекта?
13. Сформулированные цели с точки зрения заказчика, какую содержат информацию?
14. Что такое внешнее проектирование?
15. Что такое внешний проект, на кого рассчитан?
16. Что включает в себя детальный внешний проект?

ЛЕКЦИЯ 8

Тема: проектирование и кодирование программных модулей внешней спецификации.

Цель: изучить внешнее проектирование модулей изучить шаги проектирования и кодирования модулей.

(Внешнее проектирование модулей, проектирование и кодирование модулей, стиль программирования)

как создаются модули, как разбиваются на под модули декомпозицию ф-й.

Принцип внешнего проектирования модулей разрабатываются внешние взаимосвязи модулей, которые представляют собой внешнюю спецификацию каждого модуля. Внешняя спецификация модуля не должна содержать никакой информации о внутреннем устройстве модуля, об особенностях реализованного в нем алгоритма. Кроме того, недопустимо, чтобы спецификация содержала какие-либо ссылки на вызываемые модули или контексты, в которых этот модуль используется.

Т.е. внешнее проектирование модулей- это более детальный процесс внешнего проектирования.

Внешнее проектирование модуля-включает в себя полное описание внешнего взаимодействия модуля с другими модулями программы(как передаются данные, в каком формате они должны поступить к другому модулю, что модуль выполняет, какие ф-и до мельчайших подробностей, с какими модулями у него есть внешняя взаимосвязь.)

Внешняя спецификация модуля содержит следующую информацию:

1. Имя модуля.

Указывается имя, с помощью которого можно обратиться к модулю. Для модуля, имеющего несколько входов, составляются отдельные спецификации.

2. Функцию, назначение модуля (без подробностей).

Определяется, что делает модуль, когда он вызван, а также его назначение. Этот элемент спецификации не должен содержать сведения о том, как функция реализуется.

3. Список параметров.

Определяются число и порядок параметров, передаваемых модулю.

4. Входные параметры

Описываются все данные, возвращаемые модулем. Описывается взаимосвязь между входными и выходными данными,

т.е. какие выходные данные на основе каких входных данных получаются. Определяются выходные данные, возвращаемые в вызывающий модуль в случае ошибочных входных данных.

5. Выходные параметры

Дается описание всех внешних для программы или системы событий, происходящих при работе модуля, таких, как прием запроса, выдача сообщений об ошибках

После разработки внешних спецификаций модулей приступают к проектированию модуля и собственно программированию (кодированию) внутренней логики каждого модуля.

Этот процесс должен быть тщательно спланирован и состоять из следующих шагов.

Шаги проектирование и кодирование модуля:

1. Выбор языка программирования.

Существенное влияние на выбор языка оказывают его возможности обеспечивать надежный процесс получения программ, наличие и специфические особенности компилятора

2. Проектирование внешних специфик модуля.

Это процесс определения внешних характеристик каждого модуля.

3. Проверка правильности внешних спецификаций модуля.

Правильность спецификаций каждого модуля должна быть проверена сравнением их с информацией о взаимосвязях, полученной при проектировании структуры программы и в результате последующего обсуждения всеми программистами, разрабатывающими вызывающие модули.

4. Выбор алгоритма и структуры данных.

К настоящему времени разработано значительное количество алгоритмов и соответствующих структур данных. Следует использовать опыт предыдущих разработок, отчеты, выбрать из имеющихся алгоритмов и структур данных необходимые.

5. Оформление начала и конца будущего модуля в соответствии с требованиями выбранного языка программирования.

Предусматривается оформление модуля в соответствии с требованиями принятого языка программирования.

6. Объявление всех данных, используемых в качестве параметров.

Записываются соответствующие операторы объявления.

7. Объявление оставшихся данных.

Записываются операторы объявления всех оставшихся данных, которые должны быть использованы в модуле.

8. Детализация логики программы.

В результате нескольких итераций осуществляется последовательная детализация логики модуля, начиная с достаточно высокого уровня абстракции и заканчивая готовым текстом программы.

На этом шаге используются методы пошаговой детализации и структурного программирования.

9. Окончательное оформление текста программы.

Текст модуля проверяется еще раз. При этом вставляются дополнительные комментарии, поясняющие текст программы,

10. Проверка правильности программы.

Вручную проверяется правильность модуля - правильность его внутренней логики. Проверка правильности основывается на различных способах чтения текста программы. Проверка может осуществляться как в форме статического чтения программы, так и в форме динамического чтения.

11. Компиляция модуля.

Этот шаг отмечает переход проектирования к тестированию модуля. Работа над созданием модуля завершена. После компиляции на основе полученной информации проверяется правильность интерпретации компилятором намерений программиста по объявленным данным.

Пошаговая детализация – процесс разложения функции модуля на подфункции.

Применяется при декомпозиции модуля.

Структурное программирование – методология разработки программного обеспечения, в основе которой лежит представление программы в виде иерархической структуры блоков. Служит для организации проектирования и кодирования

программ таким образом, чтобы предотвратить большинство логических ошибок и обнаружить те, которые допущены.

Структурное программирование:

1. Проектирование сверху вниз.
2. Модульное программирование.
3. Структурное кодирование.

Стиль программирования – набор приемов и методов программирования, позволяющих создавать корректные, эффективные и доступные для чтения и понимания программы.

Существует набор рекомендаций:

1. Требование простоты, ясности и удобочитаемости программы.
Использовать постоянно рефакторинг кода, избегать дублирования
2. Использование программистом особенностей языка программирования.
3. Стремление программиста повысить эффективность программы не путем ее улучшения до завершения отладки, а в результате тщательного анализа структур данных, используемых ресурсов и выбора алгоритма реализации.

Контрольные вопросы:

17. Опишите принцип внешнего проектирования модулей?
18. Какую информацию содержат в себе внешние спецификации модулей?
19. Перечислите и опишите шаги проектирования и кодирования модулей?
20. Что такое пошаговая детализация?
21. Что такое структурное программирование?
22. Виды структурного программирования?
23. Что такое Стиль программирования?
24. Назовите 3 рекомендацию по кодированию модулей?

Список использованных источников:

- 17) Орлов, С.А. Технологии разработки программного обеспечения: учебник / С.А. Орлов. – СПб: Питер, 2002. – 464 с.
- 18) Липаев, В.В. Управление разработкой программных средств: Методы, стандарты, технология / В.В. Липаев. – М.: Финансы и статистика, 1993.
- 19) Липаев, В.В. Тестирование программ / В.В. Липаев. – М.: Радио и связь, 1986.
- 20) Липаев, В.В., Технология сборочного программирования / В.В. Липаев, Б.А. Позин, А.А. Штрик. – М.: Радио и связь, 1992.
- 21) Сертификация продукции. Международные стандарты и руководства ИСО/МЭК в области сертификации и управления качеством. – М.: Изд-во стандартов, 1990.
- 22) Лифиц, И.М. Стандартизация, метрология и сертификация /И.М. Лифиц. –М.: Юрайт-издат, 2004. – 335 с.
- 23) Сертификация сложных технических систем /Л.Н. Александровская [и др.]. – М.: Логос, 2001. – 312 с.
- 24) Якушев, А.И., Взаимозаменяемость, стандартизация и технические измерения / А.И Якушев, Л.Н. Воронцов, Н.М. Федотов. – М.: Машиностроение, 1986. – 352 с.

ЛЕКЦИЯ 9-10

Тема: общие положения о тестировании и отладки программы ИСО/МЭК/ИИЭР 29119-2. Принципы и методы тестирования. Взаимосвязи между общим подпроцессом тестирования, уровнями тестирования и типами тестирования

Тестирование – этап ЖЦ (40-60% от всего ЖЦ).

Цель тестирования – процесс многократного выполнения программы с целью обнаружения ошибок. Результатом тестирования являются исходные данные для отладки.

Отладка – процесс, позволяющий получить программу, функционирующую с требуемыми характеристиками в заданной области изменения входных данных.

Процесс отладки включает:

1. Действия, направленные на выявление ошибок (тестирование).
2. Диагностику и локализацию ошибок (определение характера и местонахождения ошибок).
3. Внесение исправлений в программу с целью удаления ошибок.

Существуют различные формы тестирования:

1. Доказательство. попытка найти ошибки в программе безотносительно к внешней для программы среде. Большинство методов доказательства предполагает формулировку утверждений о поведении программы и доказательство математических теорем о правильности программы.

Доказательства могут рассматриваться как форма тестирования, хотя они и не предполагают прямого выполнения программы.

2. Контроль. попытка найти ошибки, выполняя программу в тестовой, или моделируемой, среде.

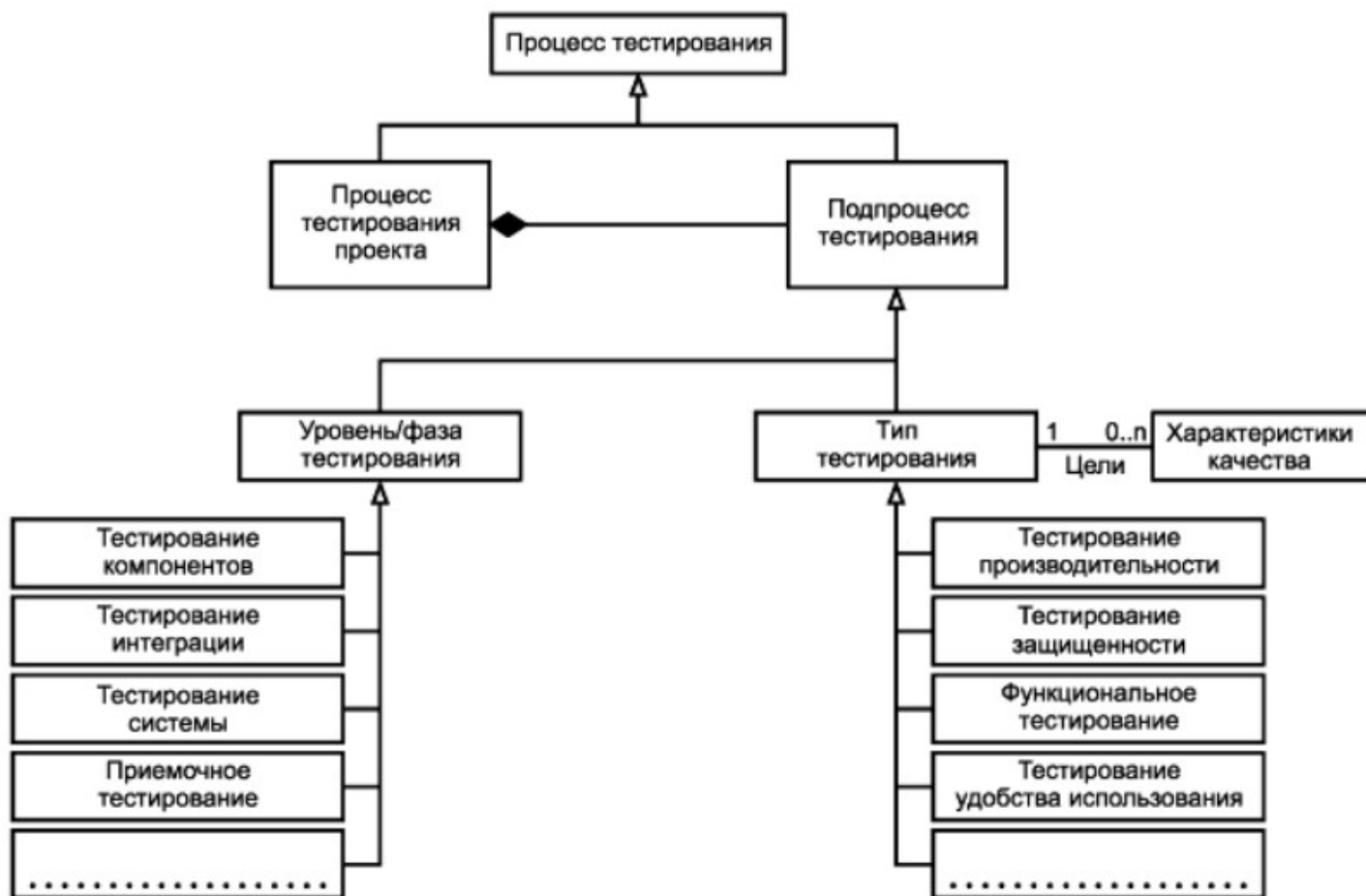
3. Испытания. попытка найти ошибки, выполняя программу в заданной реальной среде.

4. Аттестация. авторитетное подтверждение правильности программы. При тестировании с целью аттестации выполняется сравнение с некоторым заранее определенным стандартом.

Отладка не является разновидностью тестирования. Хотя слова «отладка» и «тестирование» часто используются как синонимы, но под ними подразумеваются разные виды деятельности.

Процессы тестирования, описанные в ИСО/МЭК/ИИЭР 29119-2, включают в себя процесс менеджмента тестирования и могут быть применены к тестированию во всех жизненных циклах разработки и менеджменте исследовательского тестирования;

Взаимосвязи между общим подпроцессом тестирования, уровнями тестирования и типами тестирования, как правильно оформлять в документах иерархию тестов.



Существует два метода проектирования тестовых наборов данных:

1. **тестирование программы как «белого ящика»** ISTQB, т.е. стратегия тестирования, которая управляет логикой программы. Когда разработчик видит и может проанализировать и проверить весь код.

В основе лежат тесты, как результат анализа.

это тестирование, которое основано на анализе внутренней структуры (кода) системы. Его проведение предполагает знание программирования, поэтому, как правило, осуществляется разработчиками. Методика применяется в основном при юнит- и интеграционном тестировании.

Данный вид теста предполагает не только анализ кода, но выявления причин неполадок с записью в соответствующий чек лист тестировщика.

Результаты которые документируются по данному виду тестирования: White box:

1. Анализируется реализация программы;
2. В программе определяются возможные маршруты;
3. Выбираются такие входные данные, чтобы программа выполнила выбранные пути. Это называется сенсбилизацией путей. Заранее определяются ожидаемые результаты для входных данных;

4. Тесты выполняются:

5. Результаты анализируются:

тестирование на основе структуры не ограничено использованием на уровне компонентов, а может использоваться на всех уровнях, например при покрытии пункта меню, как части тестирования системы. (ГОСТ 56920)

2. тестирование программы как «черного ящика», т.е. тестирование по входу и выходу. это процесс тестирования системы и её поведения вне зависимости от её внутренней структуры, архитектуры и реализации. Происходит сквозное тестирование всей системы

При тестировании по стратегии чёрного ящика исследуются все отдельные компоненты, например, интерфейс пользователя и UX, веб-сервер или сервер приложения, база данных, зависимости и интегрированные системы

СТРУКТУРНОЕ ТЕСТИРОВАНИЕ(часть белого ящика)

Структурное тестирование- Детальное изучение текста программы и построение тестовых наборов входных данных осуществляются на основе одного из четырех.

Критерии построения тестовых наборов:

- 1) Покрытие операторов.
- 2) Покрытие узлов ветвлений (решений).
- 3) Покрытие условий.
- 4) Комбинаторное покрытие условий.

ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ.(часть черного ящика)

Существует три метода:

1) Метод эквивалентного разбиения:

- Выделение классов эквивалентности.

- Построение тестов.

2) Метод анализа граничных значений – исследование ситуаций, возникающих на и вблизи границ эквивалентного разбиения.

Если это 0, то мы рассмотрим диапазон от -5 до 0 и от 0 до +5

3) Метод функциональных диаграмм:

- В спецификации программы выделяются причины и следствия.
- Каждой причине и следствию присваивается уникальный номер.
- Анализируются семантическое содержание спецификации, которое преобразуется в граф, связывающий причины и следствия.

- Диаграмма снабжается примечаниями.
- По функциональной диаграмме строится таблица решений.
- Столбцы решений преобразуются в тесты.

Контрольные вопросы:

25. Что такое тестирование?
26. Какие методы тестирования вы знаете?
27. Какие методы проектирования тестовых наборов знаете?
28. Что такое структурное тестирование?
29. Что такое функциональное тестирование?

25)

ЛЕКЦИЯ 11

Тема: Признаки критического модуля. Отладка ПО. Методы и принципы отладки. Критерии завершения тестирования ГОСТ Р ИСО/МЭК 12119-2000

Цель: изучить общие положения о тестировании и отладки программы. Изучить основные признаки критического модуля

МЕТОДИКА ТЕСТИРОВАНИЯ ПС.

Методики тестирования:

1. Тестирование элементов (модулей), проверяющее результаты кодирования ПС.

Проверяют результаты модулей, соответствуют действительности или нет

2. Тестирование интеграции, ориентированное на выявление ошибок этапов проектирования ПС.

Проверяют на правильность проектирования ПО, допущены ли были при проектировании системы ошибки.

3. Тестирование правильности, проверяющее корректность этапа анализа требований к ПС.

Вводим требования , которые мы получили на этапе анализа и вводим в систему и сравниваем соответствие результаты.

Хотели, чтобы система распознавала лица, сканируем изображение и сморим распознает или нет и какие ошибки выдает.

4. Системное тестирование, выявляющее дефекты этапа системного анализа ПС.

Проверяем на несоответствия Программного оборудования. К примеру будет ли работать система при малом объеме оперативной памяти.

Тестирование модулей подвергаются:

1. Интерфейс.
2. Внутренние структуры данных.
3. Независимые пути.

4. Пути обработки ошибок.
5. Граничные условия.

При тестировании путей обнаруживаются следующие категории ошибок:

1. Ошибочные вычисления.
2. Некорректные сравнения.
3. Неправильный поток управления.

Для тестирования модуля необходимы дополнительные средства:

- Драйвер – управляющая программа, которая принимает исходные данные и ожидаемые результаты тестовых вариантов, запускает в работу тестируемый модуль, получает из модуля реальные результаты и формирует донесение о тестировании.
- Заглушка – замещает модули, которые вызываются тестируемым модулем.

ТЕСТИРОВАНИЕ КОМПЛЕКСА ПРОГРАММ

Цель сборки и тестирования комплекса программ (интеграций) – взять модули, протестированные как элементы, и построить программную структуру, требуемую проектом.

Проверить правильную ли систему мы спроектировали и реализовали зависимости от поставленных задач в начале проекта .

соответствует ли система требованиям заказчика, все ф-и выполняет или нет.

Два способа комбинирования модулей в систему:

• Монолитное тестирование – каждый модуль тестируется независимо друг от друга, последовательно или параллельно, затем модули собираются в программу за «один раз». Преимущество: можно распараллелить процесс, тем самым убыстрить его выполнение.

• Пошаговое тестирование – постепенно тестируем либо нисходящее (можно показать программу сразу), либо восходящее тестирование (система не существует до тех пор, пока не будет добавлен последний модуль)

Постепенно тестируем либо готовое элемент либо тестируем его в процессе разработки.

Признаки критического модуля:

это то когда нужно созданный модуль разделять или исправлять ошибки

1. **Реализует несколько требований к системе.**

Требование и высокая скорость считывания и обработка файлов, но при этом без использования SSD накопитель и с маленькой ОП.

2. **Имеет высокий уровень управления.**

Сильно связанный с другими модулями, может вывести всю систему из работоспособного состояния

3. Имеет высокую сложность или склонность к ошибкам.

Не понятно как работает его внутренние ф-и на чем они завязаны или есть вероятность, что при большом количестве вводимых данных наш модуль выйдет из строя.

4. Имеет определенные требования к производительности обработки.
Тестирование функций. Тестирование системы

Предъявлено много требований одному модулю

5. Тестирование удобства использования.

Не удобно его подключать к системе или занимает длительный процесс времени(написание доп.скрипта)

6. Тестирование на предельных объемах.

Есть задачи, есть предметная область, должны прогнать модуль на определенном объеме входных данных и отследить, как себя ведет модуль.

7. Тестирование на предельных нагрузках.

8. Удобства эксплуатации.

9. Защита.

10. Производительность.

11. Требования к памяти.

12. Конфигурация оборудования.

Должен иметь определенный набор аппаратного и программного обеспечения и прошивок и сопроводительной документации по их использованию

13. Совместимость.

14. Удобства установки.

15. Надежность.

16. Восстановление.

17. Удобства обслуживания.

18. Документация.

Приемо-сдаточные испытания

Критерии завершения тестирования: ГОСТ Р ИСО/МЭК 12119-2000

1. Выбирается какое-то количество заранее установленных ошибок или время проверки.

2. При проведении тестирования тесты должны стать неудачными.

3. По графику.

Цель Тестирование правильности– подтвердить, что функции, описанные в спецификации требований, соответствуют ожиданиям заказчика.

элементов информации, вырабатываемых в процессе разработки ПС:

проверяем каждый пункт заданным требования и его описанию, есть он в наличии или нет, если нет, то добавляем его.

1. Спецификация требований к ПС.
2. Руководство пользователя.
3. Системная спецификация.

Представление требований системе

4. План программного проекта.
5. Спецификация проектирования.
6. Листинги исходных текстов программ.
7. План и методика тестирования.
8. Руководство по работе и инсталляции.
9. .exe код выполняемой программы.
10. Описание БД.
11. Руководство пользователя по настройке.
12. Документ сопровождения.
13. Стандарты и методики разработки программного средства.

ОТЛАДКА

Отладка -тап разработки компьютерной программы, на котором обнаруживают, локализируют и устраняют ошибки.

Чтобы понять, где возникла ошибка, приходится: узнавать текущие значения переменных; выяснять, по какому пути выполнялась программа

Отладка – совокупность действий:

1. Разработка тестовых данных.
2. Динамическое и статическое тестирование.
3. Диагностика, локализация причин отклонения, корректировки, изменения.

Два исхода отладки:

• Причина ошибки найдена, уничтожена.

• Не найдена.

Методы отладки

1. Аналитические – используют методы дедукции и индукции.

2. Экспериментальные – методы «грубой силы»: трассировка, отладочная печать, трассировка переменных, потоков управления.

Принципы отладки:

1. Использовать средства отладки как вспомогательные.

2. Избегать экспериментирования.

3. Думать.

4. Остерегаться ошибок при процессе корректировки.

5. Все исправления следует документировать.

Контрольные вопросы:

30. Что такое отладка?
31. Какую совокупность действий включает в себя отладка?
32. Методы отладки?
33. Какие исходы отладки вы знаете?
34. Какие элементы информации вырабатываются в процессе разработки ПС?
35. Какие критерии завершения тестирования вы знаете?
36. Назовите признаки критического модуля?
37. Назовите методики тестирования?
Что подвергается тестированию модулей?

Список использованных источников:

- 26) Орлов, С.А. Технологии разработки программного обеспечения: учебник / С.А. Орлов. – СПб: Питер, 2002. – 464 с.
- 27) Липаев, В.В. Управление разработкой программных средств: Методы, стандарты, технология / В.В. Липаев. – М.: Финансы и статистика, 1993.
- 28) Липаев, В.В. Тестирование программ / В.В. Липаев. – М.: Радио и связь, 1986.
- 29) Липаев, В.В., Технология сборочного программирования / В.В. Липаев, Б.А. Позин, А.А. Штрик. – М.: Радио и связь, 1992.
- 30) Сертификация продукции. Международные стандарты и руководства ИСО/МЭК в области сертификации и управления качеством.
- 31) Лифиц, И.М. Стандартизация, метрология и сертификация /И.М. Лифиц. –М.: Юрайт-издат, 2004. – 335 с.
- 32) Сертификация сложных технических систем /Л.Н. Александровская [и др.]. – М.: Логос, 2001. – 312 с.
- 33) Якушев, А.И., Взаимозаменяемость, стандартизация и технические измерения / А.И Якушев, Л.Н. Воронцов, Н.М. Федотов. – М.: Машиностроение,

ЛЕКЦИЯ 12

Тема: Оценка качества готового программного продукта по стандарту ISO 14598

После того, как мы прошли этапы

1. формирования требований
2. проектирование и кодирования
3. отладка и тестирование
4. этап оценка качества

Методологии и процессам оценивания характеристик качества ПС и их компонентов на различных этапах ЖЦ посвящен базовый международный стандарт ISO 14598:1–6:1998–2000.



Рис.5.4. Общая схема процессов оценивания характеристик качества ПС

1)В первой части изложена концепция и методология применения стандарта и его частей. Отмечается и иллюстрируется глубокая связь его положений со стандартами ISO/IEC 12207 и ISO 9126:1-4. Описана общая схема процессов оценивания характеристик качества программ (П.7 – П.10 на рис.5.4).

В этом стандарте, так же как и в ISO 9126:1-4, классифицируются характеристики качества:

- для пользователей;
- внешние и внутренние метрики

- -схема взаимосвязи этих метрик качества и их атрибутов с процессами оценивания.

Для каждой характеристики и атрибута качества рекомендуется формировать шкалу мер с выделением требуемых, допустимых и неудовлетворительных значений.

2)Во второй части изложена концепция планирования и управления процессами оценивания качества ПС, а также их связь с процессами управления ЖЦ по ISO/IEC 12207.

Представлены общие рекомендации по организации, технологии, управлению, инструментальному оснащению и проведению испытаний качества сложных комплексов программ. Для управления проектами предложены методы поддержки процессов выполнения плана оценивания качества и анализа результатов его реализации, а также сохранения и использования отчетов о планировании и управлении качеством.

Приводится шаблон плана проведения оценивания характеристик качества программных продуктов.

3)4)Третья, четвертая части стандарта содержат рекомендации для реализации процессов оценивания программного продукта с позиции разных потребителей результатов соответственно:

1. разработчиков – оценивание внутренних и внешних характеристик качества;

2. оперативных пользователей – измерение внешних метрик и метрик в использовании;

3. заказчиков и испытателей – определение метрик в использовании.

Результаты оценки характеристик качества предлагается отражать с позиций:

1. процессов ЖЦ;
2. продуктов и компонентов;
3. функционирования и применения ПС.

Требования к процессам оценивания рекомендуется структурировать на главные (функциональные), организационные, проектные,

а также выделять внутренние и внешние метрики качества и их измерение, ориентируясь на субхарактеристики и их атрибуты в соответствующей части стандарта ISO 9126:1–4.

Рекомендуемая технология выполнения оценивания качества программ включает четыре шага:

• формализацию общих требований к организации оценивания и к характеристикам качества проекта ПС;

• специфицирование требований к субхарактеристикам и атрибутам выделяемых характеристик качества и к методам их оценивания;

• планирование и проектирование процессов оценивания выделенных атрибутов качества;

• реализацию процессов оценивания выделенных атрибутов характеристик качества ПС, анализ результатов и обобщение в отчете выводов о качестве программного продукта.

Реализация процессов оценивания должна быть достаточно автономной и независимой от специалистов и процессов создания ПС и его компонентов, однако, в соответствии со стандартом ISO/IEC 12207.

характеристики и атрибуты качества рекомендуется использовать в терминах и понятиях стандарта ISO 9126:1–4, а также применять шкалы количественных и/или бальных оценок свойств или уровней качества с несколькими градациями.

5)В пятой части приводится шаблон типового отчета о результатах выполненных испытаний качества.

б)Шестая часть посвящена оформлению документации при оценке качества программных модулей и компонентов. Приведен типовой фрагмент рекомендуемой структуры и содержание подразделов для описания и оценки качества модулей.

В целом стандарт ISO 14598 предлагается применять для оценки качества ПС на различных этапах ЖЦ: на предприятиях подразделениям и специалистам по тестированию комплексов программ и их компонентов, поставщикам, потребителям, пользователям и сертификационным организациям.

Контрольные вопросы:

38. Что такое отладка?
39. Какую совокупность действий включает в себя отладка?
40. Методы отладки?
41. Какие исходы отладки вы знаете?
42. Какие элементы информации вырабатываются в процессе разработки ПС?
43. Какие критерии завершения тестирования вы знаете?
44. Назовите признаки критического модуля?
45. Назовите методики тестирования?
Что подвергается тестированию модулей?

ЛЕКЦИЯ 13

Тема: Сопровождение ПС. Конфигурационное управление. Сертификация. Стандарты сертификации

Цель: изучить принцип сертификации и конфигурационного управления ПС

СОПРОВОЖДЕНИЕ

Сопровождение ПС -Процесс модификации ПС, обусловленный необходимостью устранения выявленных в нем ошибок, и/или изменение функциональных возможностей.

Это то когда наш продукт уже какое-то время используется заказчиком и его нужно поддерживать в работоспособном состоянии оговоренный срок с заказчиком

Что бы осуществлять сопровождение не только после выпуска ПС, но и при его разработки, находить баги и устранять их вовремя, вносить изменения и обновления в версии разработки для этого используют КОНФИГУРАЦИОННОЕ УПРАВЛЕНИЕ

Конфигурационное управление Процесс применения административных и технических процедур на всем протяжении жизненного цикла для:

1. идентификации, определения и базирования единиц ПС в информационных системах;
2. управления модификацией и выпуском версий ПС;
3. фиксирования и сообщения о состоянии версии ПС;
4. для управления и контролирования хранения обращения и поставок ПС.

Все то, что может показать состояние на данный момент нашего ПО

Цель конфигурации : – обеспечить управляемое и контролируемое развитие структуры ПС.

Обеспечить управляемую и вирируемую структуру ПС , чтобы мы в любой момент могли откатить версию ПО на более ранний срок, если произошла глобальная ошибка.

СЕРТИФИКАЦИЯ ПС

Цели сертификации ПС:

1) Основная – защита интересов пользователей:

- Контроль качества.
- Обеспечение высоких потребительских свойств.
- Повышение эффективности затрат.

2) Формальная – выдача сертификата:

- Полнота, точность эталонных данных.
- Адекватные показатели качества ПС.
- Методологии интерпретации данных.

Сертификат соответствия – документ, издаваемый в соответствии с правилами системы сертификации, удостоверяющий, что обеспечивается необходимая уверенность в том, что должным образом идентифицированная продукция, процесс или услуга соответствуют конкретным стандартам или другим нормативным документам.

Задачи сертификации: распределение экономической и юридической ответственности между испытателями, разработчиками и заказчиками за качество сертифицированной продукции и за возможный ущерб при ее несоответствии документированному и объявленному качеству.

Т.е. кто будет нести ответственность, включая поломки, за нанесенный ущерб.

Сертификация бывает 2 видов:

- 1) Обязательная.
- 2) Добровольная.

Организационная структура системы сертификации

- 1) Госстандарт РФ.
- 2) Ведомственные органы те же функции, но в ограниченном объеме и для конкретных классов продукции.
- 3) Испытательные лаборатории сертификации (ИЛС)

Стандарты сертификации

- 1) ISO - 0002 : 1983 - общие термины и определения в области стандартизации и смежных видах деятельности.
- 2) ISO - 0025 : 1983 – общие требования к оценке технической компетентности.
- 3) ИЛС ISO - 0038 : 1983 – общие требования к приемке

- 4) ИЛС ISO - 0043 : 1983 - организация и проведение проверки на компетентность
- 5) ISO - 0045 : 1983 - руководящее положение по представлению результатов испытаний.
- 6) ISO - 0049 : 1983 – руководящее положение по разработке, руководство по качеству для испытаний лаборатории.
- 7) ISO - 0054 : 1983 – общие требования к приемке органов аккредитации.
- 8) ISO - 0055 : 1983 – система аккредитации ИЛС, общие требования к испытательной деятельности

Контрольные вопросы:

18. Какие цели сертификации вы знаете?
19. Назовите основные стандарты сертификации
20. Виды сертификации?
21. Назовите организационную структуру сертификации?
22. Что такое сопровождение?
23. Цель конфигурации?
24. Что такое конфигурационное управление?
25. Что такое сертификат соответствия?

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 830 структура SRS

ЛЕКЦИЯ 13

Тема: Общая характеристика ЕСПД

Основная и большая часть комплекса ЕСПД была разработана в 70-е и 80-е годы. Но уже тогда у программистов была масса претензий к этим стандартам.

Что-то требовалось дублировать в документации много раз (как, казалось – неоправданно), а многое не было предусмотрено, как, например, отражение специфики документирования программ, работающих с интегрированной базой данных.

Сейчас этот комплекс представляет собой систему межгосударственных стандартов стран СНГ (ГОСТ),

Стандарты ЕСПД в основном охватывают ту часть документации, которая создается в процессе разработки ПС, и связаны, по большей части, с документированием функциональных характеристик ПС.

Впрочем, это относится и ко всем другим стандартам в области ПС (ГОСТ серии 34, Международному стандарту ISO/IEC и др.).

Эти стандарты становятся обязательными на контрактной основе – то есть при ссылке на них в договоре на разработку (поставку) ПС.

Говоря о состоянии ЕСПД в целом, можно констатировать, что большая часть стандартов ЕСПД морально устарела.

Единая система программной документации (ЕСПД) — комплекс государственных стандартов, устанавливающих взаимосвязанные правила разработки, оформления и обращения программ и программной документации.

В стандартах ЕСПД устанавливают требования, регламентирующие разработку, сопровождение, изготовление и эксплуатацию программ, что обеспечивает возможность:

1. унификации программных изделий для взаимного обмена программами и применения ранее разработанных программ в новых разработках;
приведение к единообразной системе или форме
2. снижения трудоемкости и повышения эффективности разработки, сопровождения, изготовления и эксплуатации программных изделий;
3. автоматизации изготовления и хранения программной документации.

Сопровождение программы включает анализ функционирования, развитие и совершенствование программы, а также внесение изменений в нее с целью устранения ошибок.

недостатков ЕСПД можно отнести:

1. ориентацию на единственную каскадную модель ЖЦ ПС;
2. отсутствие четких рекомендаций по документированию характеристик качества ПС;
3. отсутствие системной связности с другими действующими отечественными системами стандартов по ЖЦ и документированию продукции в целом, например, ЕСКД;
4. нечетко выраженный подход к документированию ПС как товарной продукции;
5. отсутствие рекомендаций по само документированию ПС, например, в виде экранных меню и средств оперативной помощи пользователю (Help);
6. отсутствие рекомендаций по составу, содержанию и оформлению перспективных документов на ПС, согласованных с рекомендациями международных и региональных стандартов.

ЕСПД нуждается в полном пересмотре на основе стандарта ISO/IEC 12207 на процессы ЖЦ ПС.

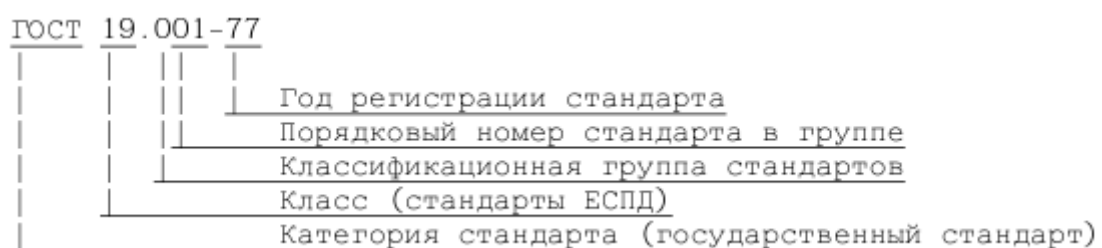
Рассмотрим кратко, какие бывают стандарты (фокусируясь на ИТ-области).

1. международные. Отличительный признак — принят международной организацией.

Пример такой организации — ISO (международная организация стандартизации). Пример её стандарта: ISO 2382-12:1988 (Периферийное оборудование). Распространены совместные стандарты ISO и международной электротехнической комиссии(IEC, в по русски — МЭК): например, ISO/IEC 12207:2008 (жизненный цикл ПО);

2. региональные. Отличительный признак — принят региональной комиссией по стандартизации.

К примеру, многие советские ГОСТы сейчас являются региональным стандартом, т.к. приняты межгосударственным советом, куда входят некоторые бывшие советские республики. Этим советом принимаются и новые стандарты — и они тоже получают обозначение ГОСТ. Пример: ГОСТ 12.4.240-2013;



3. стандарты общественных объединений; К примеру, той же МЭК: IEC 60255;

4. национальные стандарты.

Для России национальными стандартами можно считать стандарты — “ГОСТ Р”. Могут быть трех типов:

1. точные копии международных или региональных. Обозначаются неотличимо от “самописных” (национальных, написанных самостоятельно);

2. копии международных или региональных с дополнениями. Обозначаются добавлением к шифру отечественного стандарта шифра международного, который был взят за основу. Например: ГОСТ Р ИСО/МЭК 12207;

3. собственно, национальные стандарты. Например, ГОСТ Р 34.11-94.

Системы обозначений на каждом уровне и в каждой организации свои, для каждого случая придется разбираться отдельно.

Контрольные вопросы:

26. типы стандартов для России ?
27. Что такое АСПД?
28. Недостатки еспд?
29. Что устанавливают ЕСПД?
30. Какие стандарты бывают для области IT&

ЛЕКЦИЯ 15

Тема: Виды программных документов разрабатываемое на разных стадиях ЖЦ ПО. ГОСТ 19.101-77

ГОСТ 19.101-77 определяет следующие виды программных документов:

1. Спецификация - состав программы и документации на нее;
2. Ведомость держателей подлинников - перечень предприятий, на которых хранят подлинники программных документов;
3. Текст программы - запись программы с необходимыми комментариями;
4. Описание программы - сведения о логической структуре и функционировании программы;
5. Программа и методика испытаний - требования, подлежащие проверке при испытании программы, а также порядок и методы их контроля;
6. Техническое задание - назначение и область применения программы, технические, технико-экономические и специальные требования, предъявляемые к программе, необходимые стадии и сроки разработки, виды испытаний;
7. Пояснительная записка - схема алгоритма, общее описание алгоритма и (или) функционирования программы, а также обоснование принятых технических и технико-экономических решений;
8. Эксплуатационные документы - сведения для обеспечения функционирования и эксплуатации программы:
 - 8.1 Ведомость эксплуатационных документов - перечень эксплуатационных документов на программу;
 - 8.2 Формуляр - основные характеристики программы, комплектность и сведения об эксплуатации программы;
 - 8.3 Описание применения - сведения о назначении программы, области применения, применяемых методах, классе решаемых задач, ограничениях для применения, минимальной конфигурации технических средств;
 - 8.4 Руководство системного программиста - сведения для проверки, обеспечения функционирования и настройки программы на условия конкретного применения;
 - 8.5 Руководство программиста - сведения для эксплуатации программы;
 - 8.6 Руководство оператора - сведения для обеспечения процедуры общения оператора с вычислительной системой в процессе выполнения программы;
 - 8.7 Описание языка - описание синтаксиса и семантики языка;

8.8 Руководство по техническому обслуживанию - сведения для применения тестовых и диагностических программ при обслуживании технических средств.

Виды программных документов, разрабатываемых на разных стадиях:

Код вида документа	Вид документа	Стадия разработки			
		Эскизный проект	Технический проект	Рабочий проект	
				компонент	комплекс
-	Спецификация	-	-	●	●
05	Ведомость держателей подлинников	-	-	-	●
12	Текст программы	-	-	●	□
13	Описание программы	-	-	□	□
20	Ведомость эксплуатационных документов	-	-	□	□
30	Формуляр	-	-	□	□
31	Описание применения	-	-	□	□
32	Руководство системного программиста	-	-	□	□
33	Руководство программиста	-	-	□	□
34	Руководство оператора	-	-	□	□
35	Описание языка	-	-	□	□
46	Руководство по техническому обслуживанию	-	-	□	□
51	Программа и методика испытаний	-	-	□	□
81	Пояснительная записка	□	□	-	-
90-99	Прочие документы	□	□	□	□

Условные обозначения:

● - документ обязательный;

● - документ обязательный для компонентов, имеющих самостоятельное применение;

- - необходимость составления документа определяется на этапе разработки и утверждения технического задания;
- - документ не составляют.

Контрольные вопросы:

1. Назовите основные виды документов
2. Что такое пояснительная записка?
3. Что такое ТЗ?
4. Что такое руководство программиста?

ЛЕКЦИЯ 16

Тема: Структура программного проекта. Гост 19.101-77. Структура SRS
<http://asutpseta.narod.ru/download/download5.htm>

СТРУКТУРА ПРОГРАММНОГО ПРОЕКТА

(по ГОСТ 19.101-77)

(-) Спецификация

- (05) Ведомость держателей подлинников** (необязательна)
- (12) Текст программы**
- (13) Описание программы**
- (51) Программа и методика испытаний**
- (-) Техническое задание**
- (81) Пояснительная записка**
- (-) Эксплуатационные документы**
 - (20) Ведомость эксплуатационных документов
 - (30) Формуляр
 - (31) Описание применения
 - (32) Руководство системного программиста
 - (33) Руководство программиста
 - (34) Руководство оператора
 - (35) Описание языка (необязательно)
 - (46) Руководство по техническому обслуживанию

(по ГОСТ 19.103-77)

	A . V . XXXXX - XX	XX	XX - X
Общая часть обозначения программы и программных документов на нее			
Номер редакции документа			
Код вида документа			
Номер документа данного вида			
Номер части документа			

Структура SRS:

1. Введение -

Во введении объясняется значение SRS в целом, его возможности для вашей команды и его структура.

1.1. Цель

Здесь объясните цель и структуру документации по программному обеспечению SRS: типы требований, которые будут рассмотрены, а также персонал, который будет ее использовать. Этот раздел должен быть коротким: достаточно 1-2 абзацев.

1.2. Целевая аудитория

Вы можете углубиться и объяснить, как заинтересованные стороны и команды будут работать с SRS, а также участвовать в ее разработке. Обычно это владельцы продукта, инвесторы, бизнес-аналитики, разработчики, иногда тестировщики и операционный персонал. Вся структура определяется вашим подходом к разработке программного обеспечения и организационной структурой команды.

1.3. Использование по назначению

Опишите, в каких ситуациях ваша команда будет использовать SRS. Обычно его используют в следующих случаях:

- проектирование и мозговой штурм новых функций
- планирование продолжительности проекта, спринтов, оценка затрат
- оценка рисков
- мониторинг и измерение успеха команды
- конфликтные ситуации, когда вовлеченные стороны имеют разное видение качественно выполненного продукта.

1.4. Объем

В этой части рассматривается объем продукта, поэтому вам необходимо дать краткий обзор системы — ее основное назначение, функции и положение. Это сравнимо с тем, как вы объясняете продукт на собрании заинтересованных сторон, за исключением того, что вам разрешено более глубоко вникать в технические особенности.

В этом разделе должны быть описаны:

- Все типы пользователей, которые, как ожидается, будут взаимодействовать с системой
- Все основные части архитектуры

1.5 Определения и сокращения

Вышеупомянутые компоненты составляют определение. Определения предоставляют информацию о функции, базовых технологиях, целевых лицах, бизнес-объектах (пользователях, клиентах, посредниках) и заинтересованных сторонах. Вы можете использовать аббревиатуру для более быстрого написания SRS, если хотите. Документ будет доступен для чтения до тех пор, пока он включен в таблицу определений.

В вашем документе команда часто использует определенные слова. Устранение возможных недоразумений, подключение новых разработчиков и разрешение конфликтных ситуаций станет проще, если вы проясните значение этих слов.

2. Общее описание

Во второй части вы описываете читателям основные функции продукта, целевых пользователей и возможности системы. Это описание концентрируется только на ключевых функциях и архитектуре программного обеспечения, не вдаваясь в подробности о надстройках и соединениях.

2.1 Потребности пользователей

Эта часть является вопросом выбора, поэтому некоторые организации предпочитают не включать ее в свою техническую документацию SRS. Мы считаем, что лучше прямо сейчас перечислить проблемы, которые вы хотите решить с помощью вашего функционала. Это пригодится позже при мозговом штурме и мониторинге функций. Вы можете вернуться к

этому разделу в любое время в процессе разработки продукта и посмотреть, не отклонилась ли команда взаимодействия с пользователем с намеченного пути.

Потребности относятся к проблемам, которые пользователи смогут решить с помощью системы. Вы можете разделить эти потребности на подкатегории, если имеете дело с сильно сегментированной аудиторией. Старайтесь не вдаваться в подробности о потребностях каждого пользователя. Вам нужно оставить место для интерпретации на тот случай, если проблема окажется более серьезной, чем вы думали изначально.

2.2 Допущения и зависимости

Предположения — это предположения команды о продукте и его возможностях, которые будут правильными в 99% ситуаций. Естественно предположить, например, что платформа, помогающая водителям ориентироваться в ночное время, будет использоваться преимущественно в ночном режиме.

Каково значение предположений? Они позволяют в первую очередь сосредоточиться на наиболее важных функциях приложения. Это предположение помогает понять, что дизайнеры должны разработать интерфейс, подходящий для видения в темноте, для помощника вождения в ночное время. Некоторые пользователи, безусловно, могут открыть приложение в течение дня, но это далеко не так, поэтому вам не нужно сразу включать связанные элементы в прототип.

3. Особенности системы и требования

В этой части подробно рассматриваются характеристики продукта и критерии исполнения. Поскольку предыдущие два раздела посвящены продукту в целом, здесь вы найдете более подробное описание.

3.1 Функциональные требования

Функциональные требования указаны в списке функций, которые будут выполняться в системе. Эти критерии касаются вопроса «что будет создано?» а не "как" и "когда".

Функциональные требования начинаются с описания требуемой функциональности в зависимости от того, насколько она важна для приложения. Если вы хотите сначала поработать над этим, вы можете начать с дизайна, но затем вам следует перейти к разработке. Функциональные требования не содержат подробностей о стеках технологий, поскольку они могут меняться по ходу проекта. Вместо того, чтобы концентрироваться на внутренней логике, функциональные требования сосредотачиваются на функциональности конечного пользователя.

3.2 Требования к внешнему интерфейсу

Функциональные требования составляют значительную часть спецификации системных требований. Чтобы охватить все необходимые функции системы, вам понадобится 4-5 страниц информации. Некоторые команды разбивают их по темам, чтобы документ было легче читать.

Как правило, компоненты проектирования SRS рассматриваются отдельно от серверной части и бизнес-логики. Это имеет смысл, поскольку дизайнеры, а не разработчики, занимаются большей частью этой области, а также потому, что именно здесь начинается процесс разработки продукта.

В зависимости от проекта требования к внешнему интерфейсу могут состоять из четырех типов:

- Интерфейс пользователя
- Программный интерфейс
- Аппаратный интерфейс
- Интерфейс связи

Требования к внешнему интерфейсу описывают элементы страницы, которые будут видны конечному клиенту. Они могут включать в себя список страниц, элементы дизайна, ключевые

стилистические темы, даже художественные элементы и многое другое, если они необходимы для продукта.

3.3 Системные требования

Системные требования продукта определяют условия, при которых он может использоваться. Обычно они относятся к аппаратным спецификациям и функциям. Требования к оборудованию SRS часто определяются минимальным и максимальным диапазонами, а также порогом оптимальной производительности продукта.

Создание системных требований перед началом создания продукта может показаться сложным, но это необходимо. Разработчики должны придерживаться требований к оборудованию, чтобы им не пришлось перезапускать проект позже. Мобильные приложения (с множеством переменных, которые необходимо учитывать) и приложения, требующие высокой реактивности (игры, любой продукт с VR/AR или IoT), особенно уязвимы.

3.4 Нефункциональные требования

Для многих организаций эта часть SRS является самой сложной. Если функциональные требования касаются вопроса о том, что создавать, то нефункциональные стандарты определяют, как это сделать. Они устанавливают критерии того, насколько эффективно должна работать система. В эту область включены пороговые значения производительности, безопасности и удобства использования.

Настоящая ценность заключается в том, что трудно определить нефункциональные требования. Дать определение таким фразам, как «параллелизм» или «переносимость», сложно, поскольку они могут иметь различное толкование для всех вовлеченных сторон. В результате мы выступаем за присвоение каждому нефункциональному требованию оценки. Вы можете пересмотреть требования вашего проекта в любое время, чтобы увидеть, удовлетворяет ли текущая система первоначальным ожиданиям.

Требования к Visure Платформа ALM:

Visure — одна из самых надежных платформ управления жизненным циклом приложений, специализирующаяся на [управление требованиями](#) для организаций всех размеров по всему миру. В число основных партнеров Visure входят критически важные для бизнеса и безопасности компании. Компания интегрирует все процессы управления жизненным циклом приложений, включая управление рисками, отслеживание проблем и дефектов, управление прослеживаемостью, управление изменениями и различные другие области, такие как анализ качества, управление версиями требований и мощные отчеты.

Если вы ищете инструмент управления требованиями, который поможет вам с функциональными и нефункциональными требованиями, ознакомьтесь с требованиями Visure. С помощью этой платформы вы можете легко создавать, управлять и отслеживать все требования вашего проекта в одном месте.

Вывод:

Спецификация требований — это документ, в котором излагаются конкретные потребности проекта или системы. Спецификация требований важна, потому что она служит основой для всей будущей работы над проектом. Спецификация требований к программному обеспечению (SRS) отличается от спецификации бизнес-требований (BRS), хотя они и связаны между собой. SRS фокусируется на том, что система должна делать, в то время как BRS фокусируется на том, зачем нужна система и как она будет использоваться. Структура документа с требованиями к программному обеспечению может варьироваться, но всегда должна включать разделы, посвященные назначению, области применения, функциям, функциям, ограничениям, предположениям и зависимостям. Платформа ALM для требований Visure помогает с легкостью создавать SRS и управлять ими. Запросите бесплатную 30-дневную пробную версию на платформе Visure Requirements ALM, чтобы узнать, как наш инструмент может помочь вашим проектам работать более гладко.

Контрольные вопросы:

31. Назовите виды воздействий?
 32. Назовите какие бывают результаты воздействий?
 33. Назовите какие бывают продолжительности испытаний?
 34. Назовите какие бывают условия и места испытаний?
 35. Назовите какие бывают этапы разработки?
 36. Назовите какие бывают уровни проведения испытаний?
- 4.

ЛЕКЦИЯ 17

Тема: организация испытаний и оценка испытания ПП. Категории испытаний.

Цель: изучить спецификацию требований к ПО.

Испытания продукции - один из наиболее важных и интересных этапов постановки продукции на производство.

Характеристики свойств объекта при испытаниях *могут оцениваться*, если целью испытаний является получение количественных или качественных характеристик, а *могут контролироваться*, если целью испытаний является установление соответствия характеристик объекта заданным требованиям.

В этом случае испытания сводятся к контролю. Поэтому ряд видов испытаний являются контрольными, в процессе которых выполняется процесс контроля качества продукции.

Цель испытания – определение степени соответствия созданного комплекса программ

Виды испытаний:

1. Испытание опытного образца ПС на полное соответствие требованиям ТЗ.
2. Испытание рабочей версии ПС, адаптируемое к конкретным условиям применения.
3. Испытание версии (модернизированного ПС при сопровождении)

Категории испытаний

Назначение испытаний	Исследовательские	Испытания, проводимые <u>для изучения определенных характеристик свойств объекта</u>
	Контрольные	Испытания, проводимые <u>для контроля качества объекта</u>
	Сравнительные	Испытания аналогичных по характеристикам или

		одинаковых объектов, проводимые в идентичных условиях для <i>сравнения характеристик их свойств</i>
	Определительные	<u>Испытания, проводимые для определения значений характеристик объекта с заданными значениями показателей точности и (или) достоверности</u>
Уровень проведения испытаний	Государственные	Испытания установленных важнейших видов продукции, проводимые головной организацией по государственным испытаниям, или приемочные испытания, проводимые <u>государственной комиссией или испытательной организацией</u> , которой предоставлено право их проведения
	Межведомственные	Испытания продукции, <u>проводимые комиссией из представителей нескольких заинтересованных министерств для приемки составных частей объекта, разрабатываемого совместно несколькими ведомствами</u>
	Ведомственные	Испытания, <u>проводимые комиссией из представителей заинтересованного министерства или ведомства</u>
Этапы разработки продукции	Доводочные	Исследовательские испытания, <u>проводимые при разработке продукции с целью оценки влияния вносимых в нее изменений для достижения заданных значений показателей ее качества</u>
	Предварительные	<u>Контрольные испытания опытных образцов и (или) опытных партий продукции с целью определения возможности их предъявления на приемочные испытания</u>
	Приемочные	<u>Контрольные испытания опытных образцов, опытных партий продукции или изделия единичного производства, проводимые соответственно с целью решения вопроса о целесообразности постановки этой продукции на производство</u>
Испытания готовой продукции	Квалификационные	<u>Контрольные испытания первой промышленной партии, проводимые с целью оценки готовности предприятия к выпуску продукции данного типа в заданном объеме</u>
	Предъявительские	Контрольные испытания продукции, проводимые <u>службой технического контроля предприятия-изготовителя перед предъявлением ее для приемки заказчика, потребителя или других органов приемки</u>
	Приемо-сдаточные	Контрольные испытания продукции при <u>приемочном контроле</u>
	Периодические	<u>Контрольные испытания выпускаемой продукции, проводимые в объемах и в сроки, установленные нормативно-технической документацией, с целью контроля стабильности качества продукции и</u>

		возможности продолжения ее выпуска
	Инспекционные	Контрольные испытания установленных видов выпускаемой продукции, <u>проводимые в выборочном порядке с целью контроля стабильности качества продукции специально уполномоченными организациями</u>
	Типовые	Контрольные испытания выпускаемой продукции, <u>проводимые с целью оценки эффективности и целесообразности вносимых изменений в конструкцию, рецептуру или технологический процесс</u>
	Аттестационные	Испытания, <u>проводимые для оценки уровня качества продукции</u> при ее аттестации по категориям качества
	Сертификационные	Контрольные <u>испытания продукции, проводимые с целью установления соответствия характеристик свойств национальным и (или) международным нормативно-техническим документам</u>
Условия и место проведения испытаний	Лабораторные	Испытания объекта, <u>проводимые в лабораторных условиях</u>
	Стендовые	Испытания объекта, <u>проводимые на испытательном оборудовании</u>
	Полигонные	Испытания объекта, <u>проводимые на испытательном полигоне</u>
	Натурные	Испытания объекта <u>в условиях, соответствующих условиям его использования по прямому назначению с непосредственным оцениванием или контролем определяемых характеристик свойств объекта</u>
	Испытания с использованием моделей	Испытания с использованием моделей включают <u>проведение расчетов на математических или физико-математических моделях объекта испытаний и (или) воздействий на него в сочетании с натурными испытаниями объекта и его составных частей (опытно-теоретический метод испытаний), а также применение физической модели объекта испытаний или его составных частей</u>
	Эксплуатационные	Испытания объекта, <u>проводимые при эксплуатации</u>
Продолжительность испытаний	Нормальные	Испытания, методы и условия проведения которых обеспечивают получение необходимого объема информации о характеристиках свойств объекта в такой же <u>интервал времени, как и в предусмотренных условиях эксплуатации.</u>
	Ускоренные	Испытания, методы и условия проведения которых обеспечивают <u>получение необходимой информации о характеристиках свойств объекта в более короткий срок, чем при нормальных испытаниях</u>

	Сокращенные	Испытания, проводимые по <u>сокращенной программе</u>
Вид воздействия	Механические	Испытания на <u>воздействие механических факторов</u>
	Климатические	Испытания на <u>воздействие климатических факторов</u>
	Термические	Испытания на <u>воздействие термических факторов</u>
	Радиационные	Испытания на <u>воздействие радиационных факторов</u>
	Электрические	Испытания на <u>воздействие электрического напряжения, тока или ноля</u>
	Электромагнитные	Испытания на <u>воздействие электромагнитных полей</u>
	Магнитные	Испытания на <u>воздействие магнитного поля</u>
	Химические	Испытания на <u>воздействие специальных сред</u>
	Биологические	Испытания на <u>воздействие биологических факторов</u>
Результат воздействия	Неразрушающие	Испытания с <u>применением неразрушающих методов контроля</u>
	Разрушающие	Испытания с <u>применением разрушающие методов контроля</u>
	Испытания на прочность	Испытания, <u>проводимые для определения значений воздействующих факторов</u> , вызывающих выход значений характеристик свойств объекта <u>за установленные пределы или его разрушение</u>
	Испытания на устойчивость	Испытания, проводимые <u>для контроля способности изделия выполнять свои функции и сохранять значения параметров в пределах установленных норм</u> во время действия на него <u>определенных факторов</u>
Определяемые характеристики и объекта	Функциональные	Испытания, проводимые с <u>целью определения значений показателей назначения объекта</u>
	Испытания на надежность	Испытания, <u>проводимые для определения показателей надежности в заданных условиях</u>
	Испытания на безопасность	Испытания, <u>проводимые для определения показателей безопасности продукции</u>
	Испытания на транспортабельность	Испытания, <u>проводимые для определения сохраняемости изделия после транспортировки</u>
	Граничные испытания	Испытания, проводимые <u>для определения зависимостей между предельно допустимыми значениями параметров объекта и режимом эксплуатации</u>
	Технологические испытания	Испытания, <u>проводимые при изготовлении продукции с целью оценки ее технологичности</u>

Контрольные вопросы:

37. Назовите виды воздействий?
38. Назовите какие бывают результаты воздействий?
39. Назовите какие бывают продолжительности испытаний?
40. Назовите какие бывают условия и места испытаний?
41. Назовите какие бывают этапы разработки?
42. Назовите какие бывают уровни проведения испытаний?

Список использованных источников:

5. Государственный стандарт Российской Федерации. Информационная технология. Сопровождение программных средств
6. Основы маркетинга учебное пособие Суркова Е.В.
7. Рекомендуемая стандартом IEEE 830 структура SRS