**1. Жизненный цикл программных средств, его этапы. Основные модели жизненного цикла.** Под жизненным циклом по понимают весь процесс его разработки и использования начиная от момента возникновения идеи создания по,заканчивая прекращением всех видов его использования. Планирование, проектирование,реализация,ввод в действие,использование и сопровождение. Каскадная, Поэтапная модель с промежуточным контролем, Спиральная

**2.Жизненный цикл программных средств. Стандарты в области организации жизненного цикла программных средств.** Под жизненным циклом по понимают весь процесс его разработки и использования начиная от момента возникновения идеи создания по,заканчивая прекращением всех видов его использования. Международный стандарт ISO/IEC 12207 (Системная и программная инженерия - Процессы жизненного цикла ПС).

Базовый стандарт Беларуси СТБ ИСО/МЭК 12207-2003( Информационная технология - Процессы жизненного цикла ПС)

Базовый стандарт России ГОСТ Р ИСО/МЭК 12207-99

**3. Основные понятия в области жизненного цикла программных средств. Компоненты жизненного цикла. Иерархическая структура жизненного цикла.**

***Аттестация*** *-* подтверждение экспертизой и представлением объективных доказательств того, что конкретные требования к конкретным объектам полностью реализованы.

***Аудит*** *-* проверка, выполняемая компетентным органом (лицом) с целью обеспечения независимой оценки степени соответствия программных продуктов или процессов установленным требованиям.

***Верификация*** *-* подтверждение экспертизой и представлением объективных доказательств того, что конкретные требования полностью реализованы.

***Квалификационное испытание -*** испытание (тестирование), проводимое разработчиком, при необходимости санкционированное заказчиком, для демонстрации того, что программный продукт удовлетворяет установленным требованиям и готов к использованию в заданных условиях эксплуатации.

***Квалификационное требование-***набор критериев или условий, которые должны быть удовлетворены для того, чтобы квалифицировать программный продукт на соответствие установленным требованиям и готовность к использованию в заданных условиях эксплуатации.

***Обеспечение качества*** *-* все запланированные и систематически выполняемые в рамках системы качества работы; при необходимости объективные доказательства, обеспечивающие уверенность в том, что объект

будет полностью соответствовать установленным требованиям качества.

***Тестируемость-***степень, до которой могут быть запланированы

объективность и реализуемость тестирования, проверяющего соответствие требованию.

***Техническое задание-*** документ, используемый заказчиком в качестве

средства для описания и определения задач, выполняемых при реализации договора.

В соответствии со стандартом ЖЦ ПС и систем имеет ***трехуровневую иерархическую структуру*:** основу составляют ***процессы,*** которые разделены на ***работы,*** *которые* разделены на ***задачи***.

Процессы делятся на ***три группы***:

■основные;

■вспомогательные;

■организационные.

**Основные компоненты жизненного цикла:**

Процессы

Работы

Задачи

**4. СТБ ИСО/МЭК 12207-2003. Процессы жизненного цикла программных средств и их классификация.**

**Процессы делятся на *три группы*:**

**■основные;**

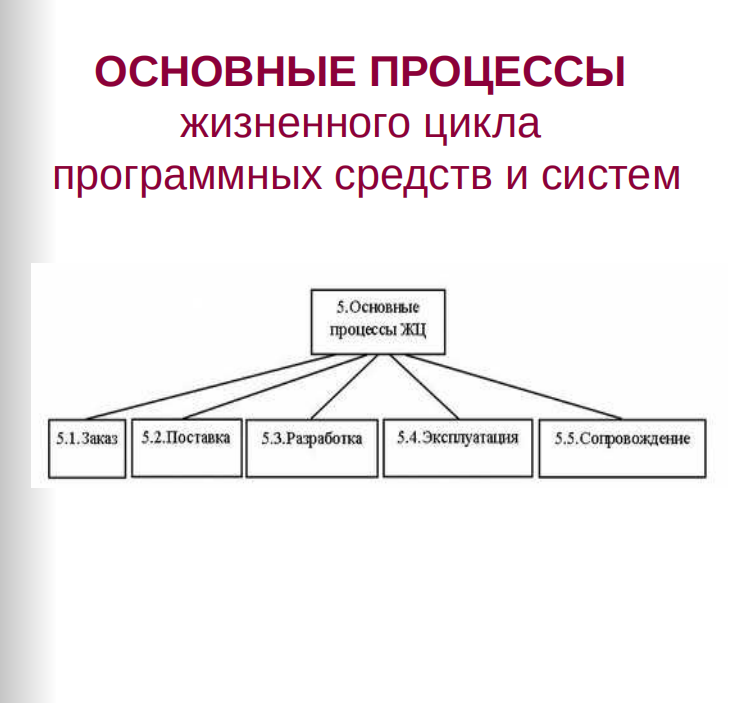
**■вспомогательные;**

**■организационные.**



**5. СТБ ИСО/МЭК 12207-2003. Основные процессы жизненного цикла**

**■Основные процессы жизненного цикла –** это процессы,которые реализуются под управлением основных сторон, участвующих в ЖЦ ПС.Основными сторонами являются заказчик, поставщик, разработчик, оператор и персонал сопровождения программных продуктов.

****

**■ *Процесс заказа*** определяет работы и задачи заказчика. Процесс заказа состоит из определения потребностей заказчика в системе, программном продукте или программной услуге, подготовки и выпуска заявки на подряд, выбора поставщика и управления процессом заказа до завершения приемки системы, программного продукта или программной услуги.

■ ***Процесс разработки***

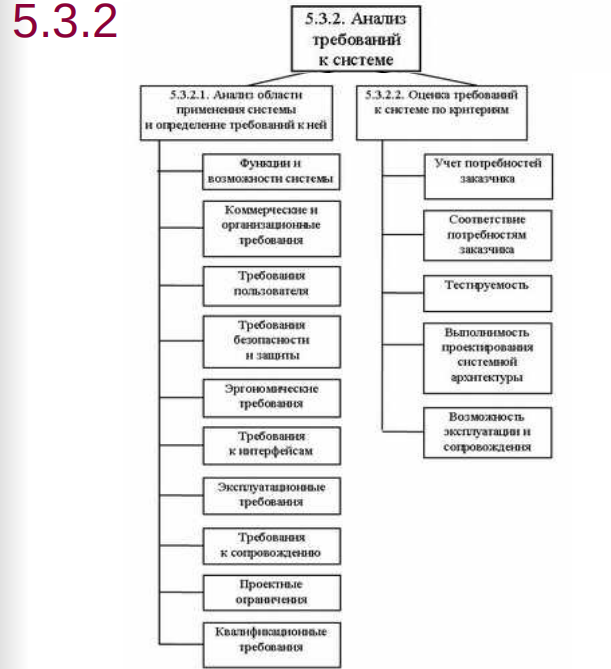
■ ***Процесс поставки*** определяет работы и задачи поставщика. Процесс поставки начинается с решения о подготовке предложения в ответ на заявку на подряд, присланную заказчиком, или с подписания договора с заказчиком на поставку системы, ПП или программной услуги. Затем определяются процедуры и ресурсы, необходимые для управления и обеспечения проекта, включая разработку проектных планов и их выполнение.

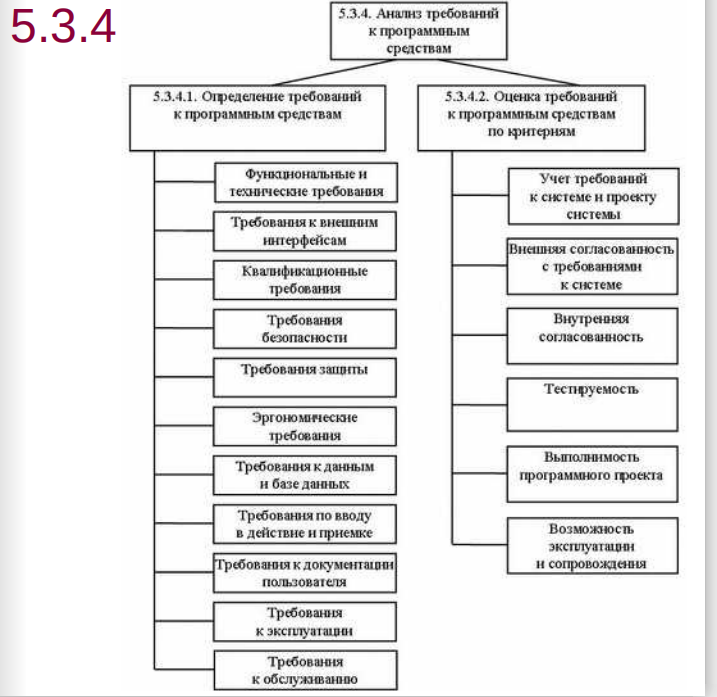
■***Процесс эксплуатации*** определяет работы и задачи оператора. Данный процесс включает эксплуатацию программного продукта и поддержку пользователей в процессе эксплуатации.

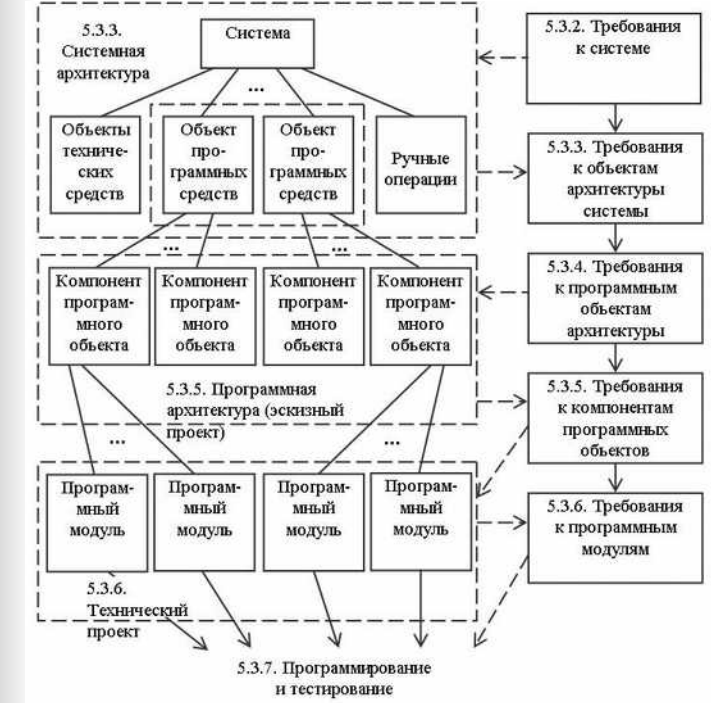
■ ***Процесс сопровождения*** определяет работы и задачи персонала сопровождения и реализуется при модификациях программного продукта. Цель процесса – изменение существующего ПП при сохранении его целостности. Процесс охватывает вопросы переносимости и снятия ПП с эксплуатации.

**6. СТБ ИСО/МЭК 12207-2003. Процесс разработки и его структура.**









**7. СТБ ИСО/МЭК 12207-2003. Вспомогательные процессы жизненного**

**■Вспомогательные процессы жизненного цикла –** это процессы,

являющиеся целенаправленными составными частями других процессов.

Их основное назначение – обеспечить успешную реализацию и качество выполнения программного проекта. Вспомогательный процесс инициируется и используется другим процессом.



**8. СТБ ИСО/МЭК 12207-2003. Организационные процессы жизненного цикла.**

**■Организационные процессы жизненного цикла –** это

процессы, предназначенные для создания в некоторой организации и совершенствования организационных структур, охватывающих процессы ЖЦ и соответствующий персонал. Обычно организационные процессы являются типовым



**9. СТБ ИСО/МЭК 12207-2003. Процесс верификации и его структура.**

Базовый стандарт Беларуси СТБ ИСО/МЭК 12207-2003 (Информационная технология – Процессы жизненного цикла ПС)

**Процесс верификации** - является процессом определения того, что программные продукты функционируют в полном соответствии с требованиями и условиями, реализованными в предшествующих работах.

Структура процесса верификации: Подготовка процесса верификации и сама верификация

Договора ● возможности поставщика ● непротиворечивость ● наличие процедур создания/ внесения изменений/ взаимодействия. ■

Процесса ● пригодность и применимость ● соответствие и своевременность ● готовность персонала. ■

Требования ● правильность, непротиворечивость, выполнимость и тестируемость. ■

Результаты тестирования ● правильность, соответствие требованиям и ограничениям ● возможность дальнейшего использования. ■

Исходные коды ● трассируемость и тестируемость ● завершенность, правильность и соответствие ограничениям ● реализация и возможность дальнейшего использования. ■

Документация ● соответствие, полнота и непротиворечивость ● своевременность и соблюдение установленных процедур.

**10. СТБ ИСО/МЭК 12207-2003. Процесс аттестации и его структура.**

Базовый стандарт Беларуси СТБ ИСО/МЭК 12207-2003 (Информационная технология – Процессы жизненного цикла ПС)

**Процесс аттестации** - является процессом определения полноты соответствия установленных требований, созданной системы или программного продукта их функциональному назначению.

Структура процесса аттестации:Подготовка процесса аттестации и сама аттестация

**11. СТБ ИСО/МЭК 12207-2003. Процесс совместного анализа и его структура.**

**Процесс совместного анализа** - является процессом оценки состояний и результатов работ по проекту. Совместные анализы проводятся в течение всего договора и применяются как на уровне управления проектом, так и на уровне его технической реализации. Процесс может выполняться двумя любыми сторонами, участвующими в договоре, когда одна сторона (анализирующая) проверяет другую (анализируемую).

**Структура процесса совместного анализа**: Подготовка процесса совместного анализа, Анализы управления проектом, технические анализы

**12. Инструментальные средства автоматизации жизненного цикла.** Очевидно, что большие размеры и высокая сложность разрабатываемых программных средств при ограничениях на бюджетные и временные затраты проекта могут привести к низкому качеству конечных программных продуктов и системы в целом. В этой связи в последнее время все большее внимание уделяется современным технологиям и инструментальным средствам, обеспечивающим автоматизацию процессов жизненного цикла программных средств (CASE-средствам). Использование таких инструментальных средств позволяет существенно сократить длительность и стоимость разработки систем и ПС при одновременном повышении качества процесса разработки и, как следствие, качества разработанных программных средств.

К современным инструментальным средствам, обеспечивающим эффективную поддержку процессов жизненного цикла программных средств и систем, можно отнести семейство интегрированных продуктов компании ***Telelogic***. Основными в данном семействе являются следующие инструментальные средства.

**13. Качество программных средств. Основные понятия и определения. Показатели качества программных средств.**

**Атрибут (attribute) –** внутренне присущее объекту свойство, которое мо- жет быть распознано количественно или качественно человеком или автомати- зированными средствами. В стандартах серии *SQuaRE* понятие атрибута ис- пользуется аналогично понятию свойства.

**Базовая мера (base measure)** – мера, определенная в терминах атрибута и метода для ее количественной оценки. Базовая мера функционально не зависит от других мер. В стандартах серии *SQuaRE* понятие базовой меры соответству- ет понятию элемента меры качества.

**Внешняя мера качества программного средства (external measure of software quality)** – мера обеспечения программным продуктом поведения си- стемы (включая программное обеспечение (ПО)), удовлетворяющего заданным и подразумеваемым потребностям, при использовании программного продукта в заданных условиях. Например, количество отказов, обнаруженных при тести- ровании – это внешняя мера качества программного средства (ПС), связанная с количеством ошибок в нем. Две меры количества отказов, измеренные в разное время, могут отличаться друг от друга, поскольку тестирование может не найти всех ошибок, а ошибка может вызвать другие отказы в других условиях.

**Внутренняя мера качества программного средства (internal measure of software quality)** – мера удовлетворения набором статических атрибутов про- граммного продукта заданных и подразумеваемых потребностей в данном про- дукте при его использовании в заданных условиях. Статические атрибуты – это атрибуты, связанные с архитектурой, структурой ПС и его компонентами.

Например, меры сложности, а также количество, серьезность и частота найденных при проверке ошибок являются внутренними мерами качества, при- меняемыми по отношению к самому программному продукту.

**Качество в использовании (quality in use)** – степень применимости про- дукта или системы заданными пользователями для удовлетворения их потреб- ностей в достижении заданных целей с результативностью, эффективностью, свободой от риска и удовлетворенностью в заданных контекстах использования.

**Качество программного средства (software quality)** – степень удовле- творения программным продуктом заданных и подразумеваемых потребностей при его использовании в заданных условиях.

**Контекст использования (context in use)** – пользователи, задачи, среда (аппаратная, программная, материалы), а также физическое и социальное окру- жение, в которых продукт используется.

**Мера (measure)** – переменная, которой присваивается значение в резуль- тате измерений. Термин «мера» используется собирательно для указания базо- вых мер, производных мер и указателей.

**Мера качества (quality measure)** – производная мера, определяемая как результат функции измерения двух или более величин элементов меры качества (ЭМК).

**Метод измерения (measurement method)** – логическая последователь- ность действий, описанная с общих позиций и используемая при измерении.

**Модель качества (quality model)** – определенный набор характеристик и связей между ними, который обеспечивает схему для задания требований к ка- честву и оценки качества.

**Подразумеваемые потребности (implied needs)** – потребности, которые не могут быть заданы, но являются реально существующими потребностями. Подразумеваемые потребности включают потребности не заданные, но подра- зумеваемые другими заданными потребностями, и те незаданные потребности, которые рассматриваются как очевидные. Некоторые потребности становятся очевидными при использовании продукта в конкретных условиях.

**Подхарактеристика качества программного средства (software quality subcharacteristic)** – характеристика качества программного средства, входящая в состав другой характеристики качества.

**Правообладатель (stakeholder)** – лицо или организация, имеющие права, долю, претензии или интерес к системе или ее характеристикам, обеспечиваю- щим их потребности и ожидания.

**Программный продукт (software product)** – набор компьютерных про- грамм, процедур, возможно связанных с ними документации и данных. В стан- дартах серии *SQuaRE* качество программных средств (программного обеспече- ния) имеет то же значение, что и качество программного продукта.

**Производная мера (derived measure)** – мера, определяемая как функция двух или более значений базовых мер. Базовая мера, преобразованная с помо- щью математической функции, также может рассматриваться как производная мера.

**Процедура измерения (measurement procedure)** – логическая организа- ция действий, применяемая специфически, используемая при выполнении кон- кретных измерений в соответствии с данным методом измерений. Процедура измерений обычно оформляется в виде документа, в котором она должна быть описана достаточно подробно, чтобы позволить выполнять измерение без до- полнительной информации.

**Свойство для количественной оценки (property to quantify)** – свойство целевого объекта, связанное с элементом меры качества, которое может быть оценено количественно с помощью метода измерений. В стандартах серии *SQuaRE* понятие свойства используется аналогично понятию атрибута.

**Свойство качества (quality property)** – измеримый компонент качества.

**Система (system)** – совокупность взаимодействующих элементов, органи- зованная для достижения одной или более заданных целей.

**Функция измерения (measurement function)** – алгоритм или вычисления, выполняемые для объединения двух или более элементов мер качества.

**Характеристика качества программного средства (software quality characteristic)** – категория свойств программного средства, с помощью которой выражается его качество. Характеристики качества программных средств могут быть определены с помощью подхарактеристик и, в конечном итоге, атрибутов качества ПС.

**Целевой объект (target entity)** – фундаментальный объект интереса для пользователя, о котором собирается информация и который должен быть изме- рен. Целевым объектом могут быть разрабатываемые рабочие продукты, а так- же поведение системы, программного средства или правообладателей, таких как пользователи, операторы, разработчики, тестировщики или персонал со- провождения.

**Элемент меры качества (quality measure element)** – мера, определяемая в терминах свойства и метода измерения для ее количественной оценки, воз- можно включая преобразование с помощью математической функции. Элемент меры качества может являться как базовой, так и производной мерой.

**14. Общие сведения о стандартах в области оценки качества, действующих на территории Республики Беларусь.**

В настоящее время в области оценки качества ПС на территории Респуб- лики Беларусь действуют следующие основные стандарты:

межгосударственный стандарт стран СНГ ***ГОСТ 28806–90. Качество программных средств. Термины и определения*** [3];

межгосударственный стандарт стран СНГ ***ГОСТ 28195–99. Оценка ка- чества программных средств. Общие положения*** [2];

национальный стандарт Беларуси ***СТБ ИСО/МЭК 9126-2003. Информа- ционные технологии. Оценка программной продукции. Характеристики ка- чества и руководства по их применению*** [4];

### национальный стандарт Беларуси **СТБ ISO/IEC 25000-2009. Разработка программного обеспечения. Требования к качеству и оценка программного продукта (SQuaRE). Руководство по SQuaRE** [5];

национальный стандарт Беларуси ***СТБ ISO/IEC 25001-2009. Разработка программного обеспечения. Требования к качеству и оценка программного продукта (SQuaRE). Планирование и управление*** [6].

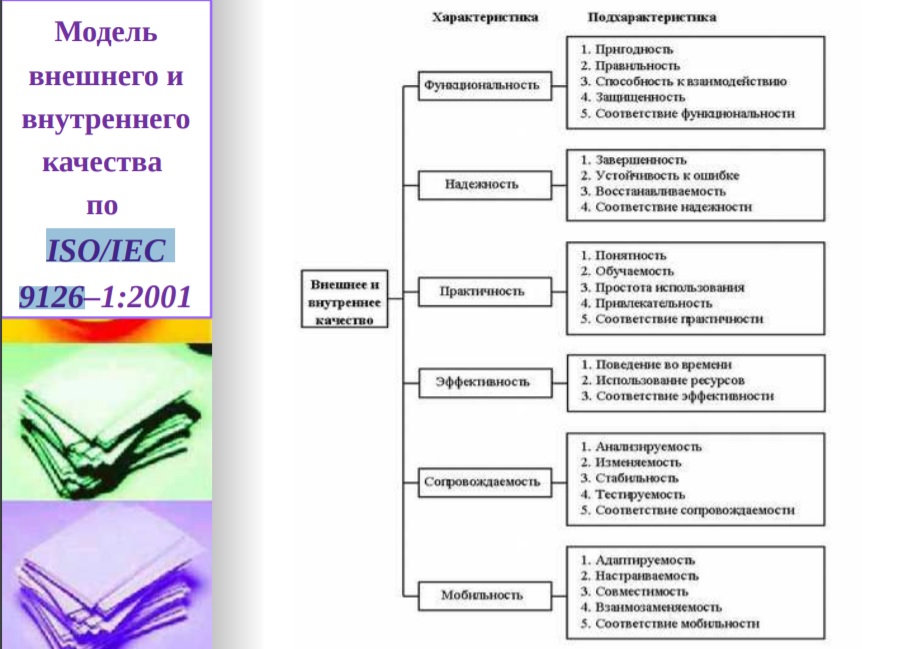
*СТБ ИСО/МЭК 9126-2003* представляет собой аутентичный перевод меж- дународного стандарта *ISO/IEC 9126:1991* [20].

В *ГОСТ 28806–90* приведены основные термины и определения, принятые в области обеспечения качества ПС, определена модель качества ПС.

*В ГОСТ 28195–99* представлены модель качества ПС, метод оценки каче- ства ПС, классификация методов измерений свойств ПС.

*СТБ ISO/IEC 25000-2009* и *СТБ ISO/IEC 25001-2009* представляют собой аутентичные переводы международных стандартов *ISO/IEC 25000:2005* и *ISO/IEC 25001:2007*. Данные стандарты относятся к серии стандартов *SQuaRE*. Информация о данной серии приведена в подразд. 1.6.

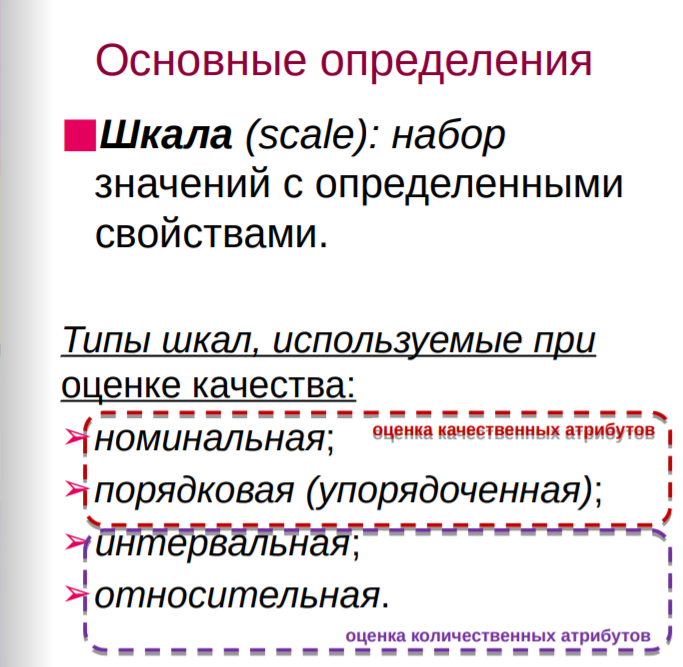
**15. ISO/IEC 9126-1:2001. Иерархическая модель оценки качества программного средства.**



**16. Метрики и их типы. Шкала. Типы шкал. Атрибуты.**

Метрика (quality) ПОmetric): определенный метод и шкала измерения. Метрики могут быть внутренними, внешними или метриками качества в использовании; прямыми или косвенными.

Атрибут (quality) ПОattribute): измеримое физическое или абстрактное свойство продукта.



**17. Свойства и критерии обоснованности метрик качества программных средств.**

Стандарт ISO/IEC 9126-1:2001 классифицирует метрики качества ПС на внутренние, внешние и метрики качества в использовании.

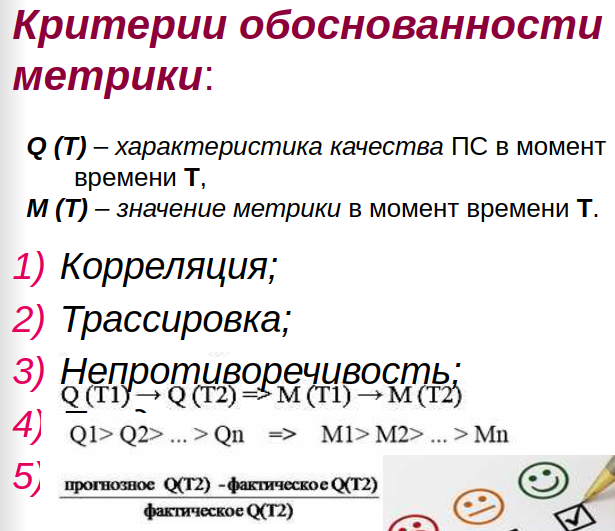
Применение конкретного вида метрик определяется стадией ЖЦ ПС.

Вторая, третья и четвертая части стандарта ISO/IЕС 9216-2-4 посвящены детальному рассмотрению соответственно внешних и внутренних метрик качества программных средств и метрик качества в использовании.

Ожидаемые свойства метрик

1. Надежность
2. Повторяемость
3. Однотипность
4. Применимость
5. Показательность
6. Корректность
7. Значимость

Метрика также должна быть эффективной по отношению к стоимости.



**18. Метрики программных продуктов. Метрики сложности программных**

**средств.**

Метрики программных проектов - это количественные показатели, отражающие их отдельные характеристики.

По виду информации, получаемой при оценке качества ПО, метрики можно разбить на 3 группы :

■ метрики, оценивающие отклонение от нормы характеристик исходных проектных материалов.

■ метрики, позволяющие прогнозировать качество разрабатываемого ПО.

■ метрики, по которым принимается решение о соответствии конечного ПО

заданным требованиям.

Основные направления применения метрик

Выделяют ШЕСТЬ основных направлений :

1. оценки топологической и информационной сложности ПО;

2. оценки надежности ПО;

3. оценки производительности ПО и повышения его эффективности;

4. оценки уровня языковых средств и их применения;

5. оценки трудности восприятия и понимания программных текстов;

6. оценки производительности труда программистов.

**Метрики сложности ПО**

Выделяют 3 основные группы:

1) метрики размера программ

2) метрики сложности потока

управления программ

3) метрики сложности потока

данных программ

**19. Метрики размера программ. Примеры.**

**Метрики размера программ**

Традиционной характеристикой размера программ является количество строк исходного текста (кода)

**(Source Lin):es Of Code, SLOC).**

Выделяют два основных подхода:

■ оценка показателя SLOC по аналогии;

■ оценка показателя SLOC «снизу-вверх» экспертным методом.

**МЕТРИКА ХОЛСТЕДА**

Основу метрики составляют 4 измеряемых характеристики программы:

■ n1 - число уникальных операторов программы, включая символы-разделители, имена процедур и знаки операций (словарь операторов);

■ n2 - число уникальных операндов программы (словарь операндов);

■ N1 - общее число операторов в программе;

■ N2 - общее число операндов в программе.

**Оценки метрики Холстеда**

■словарь программы

n = n1+n2,

■длину программы

N = N1+N2, (1)

■объем программы

V = N\*log2(n) (quality) ПОбит) (2)

**20. Метрики сложности потока управления программ. Примеры.**

Основная цель метрик сложности – выявить наиболее критичные участки ПС,

которые являются потенциальными источниками ошибок и повышенных рисков на всех стадиях его жизненного цикла. Характерно представление программ в виде управляющего ориентированного графа G=(V,E), где V - вершины, соответствующие операторам, E - дуги, соответствующие переходам. Структурная сложность определяется:

■числом взаимодействующих компонент;

■числом связей между компонентами;

■сложностью взаимодействия компонент.

Для оценивания сложности потока управления программы –

ЦИКЛОМАТИЧЕСКАЯ СЛОЖНОСТЬ,(Томас МакКейб, Thomas McCabe)

Данная метрика вычисляется на основе ориентированного графа.

Формула вычисления цикломатической сложности выглядит:

|  |  |
| --- | --- |
| Z(G) = l – v + 2\*p |  |

l – число ребер (дуг) графа программы,

v – число узлов (вершин) графа,

p – число компонентов связности графа.

Число компонентов связности графа можно рассматривать как количество

дуг, которые необходимо добавить для преобразования графа в сильносвязный.

Сильносвязным называется граф, любые две вершины которого взаимно достижимы.

Цикломатическое число

Маккейба

По сути Z(G) определяет число линейнонезависимых контуров в сильносвязном графе. Цикломатическое число Маккейба показывает требуемое (минимальное) число проходов для покрытия всех контуров сильносвязанного графа или количество тестовых прогонов программы, необходимых для исчерпывающего тестирования по критерию «работает каждая ветвь» 

**Метрика подсчета точек пересечения**

Определение: В графе программы, где каждому оператору соответствует вершина, т. е. не исключены линейные участки, при передаче управления от вершины a к b номер оператора a равен min(a,b), а номер оператора b - max(a,b).

Точка пересечения дуг возникает в случае выхода управления за пределы пары вершин (a,b) .

**МЕТРИКА ДЖИЛБА**

Простая, но достаточно эффективная метрика определения сложности

программы – подсчет насыщенности программы условными операторами.

Вводятся две характеристики:

■ CL - абсолютная сложность программы, характеризующаяся количеством операторов условия;

■ cl - относительная сложность программы, характеризующаяся насыщенностью программы операторами условия, т. е. cl определяется как отношение CL к общему числу операторов.

На основе метрики Джилба построена еще одна метрика – характеристика

максимального уровня вложенности оператора (CLI).

**МЕТРИКА ГРАНИЧНЫХ ЗНАЧЕНИЙ**

Пусть G=(V,E) — ориентированный граф программы с единственной

начальной и единственной конечной вершинами. Отрицательная степень вершины – число входящих в вершину дуг. Положительная степень вершины –

число исходящих из вершины дуг.

Тогда набор вершин графа можно разбить на две группы:

■вершины у которых положительная степень <=1 (принимающие вершины)

■вершины у которых положительная степ >=2 (вершины отбора)

Каждая вершина имеет скорректированную сложность:

■Принимающая = 1,

■Конечная = 0.

■Вершина отбора = числу вершин подграфа.

Скорректированные сложности всех вершин графа G суммируются,

образуя абсолютную граничную сложность программы (Sa)

Граничная сложность программы

определяется как: S0=1-(v-1) / Sa ,где

■S0 – относительная граничная сложность программы;

■Sa – абсолютная граничная сложность программы,

■v – общее число вершин графа программы.

**21. Метрики сложности потока данных программ. Примеры.**

**Метрика сложности потока**

**данных**

Характеристика Aup отражает сколько раз модули Up действительно получили доступ к глоб. переменным,число Pup – сколько раз они могли бы получить доступ.

Отношение числа фактических обращений к возможным определяется как :

Rup = Aup / Pup

Формула показывает приближенную вероятность ссылки произвольного модуля на произвольную глобальную переменную. Чем выше эта вероятность, тем выше вероятность «несанкционированного» изменения к.-либо переменной

**Спен** – это число утверждений, содержащих данный идентификатор, между его первым и последним появлением в тексте программы. Идентификатор, появившийся n): раз, имеет спен, равный n):-1.

Спен определяет количество контролирующих утверждений (контрольных точек), вводимых в тело программы при построении трассы программы по этому идентификатору в процессе тестирования и отладки метрика Чепина

Суть метода состоит в оценке информационной прочности отдельно взятого

программного модуля с помощью анализа характера использования переменных из списка ввода-вывода.

**метрика Чепина** Все множество переменных, составляющих список вводавывода разбивается на 4 функциональные группы:

■ Р – вводимые переменные для расчетов и для обеспечения вывода.

■ М – модифицируемые или создаваемые внутри программы переменные.

■ C – переменные, участвующие в управлении работой программного модуля (quality) ПОуправляющие переменные).

■ T – не используемые в программе (quality) ПО«паразитные») переменные.

Т.к. каждая переменная может выполнять одновременно несколько функций, необходимо учитывать ее в каждой соответствующей функциональной группе.

Q = P + 2\*M + 3\*C + 0.5\*T

**22. Основные понятия сертификации программных средств и систем качества.**

В руководстве ИСО/МЭК 2-1983 Общие термины и определения в области стандартизации и смежных видов деятельности - *сертификация соответствия* определена как действие третьей стороны, доказывающее, что обеспечивается необходимая уверенность в том, что должным образом идентифицированная продукция, процесс или услуга соответствует конкретному стандарту или другому нормативному документу. В понятие нормативные документы включены документы, содержащие правила, общие принципы или характеристики, касающиеся различных видов деятельности или их результатов, стандарты, технические требования, инструкции и регламенты по применению.

Результатом положительных испытаний является сертификат соответствия - документ, изданный в соответствии с правилами системы сертификации, удостоверяющий, что обеспечивается необходимая уверенность в том, что должным образом идентифицированная продукция, процесс или услуга соответствует конкретным стандартам или другим нормативным документам.

Основные понятия, цели и виды сертификации программных средств:

1)стандартное определение понятий и компонентов процессов сертификации;

2)распределение потребностей в сертификации продукции среди производителей и заказчиков программных средств;

3) задачи и эффективность обязательной и добровольной сертификации программных средств.

**23. Организация сертификации программных средств.**

Решение о выдаче сертификата на основе соответствия действующим и/или специально разработанным документам:

1)международным и государственным стандартам на технологии создания ПС, их системы обеспечения качества и конкретную продукцию;

2)стандартам на сопровождающую программный продукт документацию с учетом необходимости и достаточности номенклатуры документов, семантической полноты и однозначности понимания содержания документов;

3)нормативным и эксплуатационным документам на конкретный программный продукт - техническим условиям, техническим описаниям, спецификациям требований и другим регламентирующим документам по согласованному выбору заказчика, разработчика и испытателя;

4)действующим международным и национальным стандартам на тестирование, испытания, аттестацию программ, требования которых не ниже требований, регламентируемых отечественными документами.

Сертификационные испытания являются наиболее формализованным и регламентированным этапом тестирования, как продуктов, так и процессов их создания, поддерживаемым значительным числом документов. При сертификации обычно руководствуются следующими основными исходными требованиями заказчика:

1)утвержденным заказчиком и согласованным с разработчиком техническим заданием и/или спецификацией требований к продукту, а также утвержденным комплектом эксплуатационной документации на комплекс программ и его компоненты, а также на систему обеспечения их качества;

2)действующими международными, государственными и ведомственными стандартами на проектирование и испытания программ, а также на техническую документацию производства и продукции;

3)программой испытаний по всем требованиям технического задания и положениям эксплуатационной документации;

4)методиками испытаний по каждому разделу требований технического задания и документации.

**24. Оценка соответствия в Республике Беларусь.**

Оценка соответствия в Беларуси выполняется на основе Закона Республики Беларусь № 269-З «Об оценке соответствия требованиям технических нор-мативных правовых актов в области технического нормирования и стандартизации» [55]. Оценка соответствия осуществляется в целях:

·обеспечения защиты жизни, здоровья и наследственности человека, имущества и охраны окружающей среды;

·повышения конкурентоспособности продукции (услуг);

·создания благоприятных условий для обеспечения свободного перемещения продукции на внутреннем и внешнем рынках.

Принципами оценки соответствия являются:

·гармонизация с международными и межгосударственными(региональными) подходами в области оценки соответствия;

·обеспечение идентичности правил и процедур подтверждения соответствия продукции отечественного и иностранного производства;

· соблюдение требований конфиденциальности сведений, полученных при выполнении работ по оценке соответствия.

Основными объектами оценки соответствия являются:

·продукция;

·процессы жизненного цикла продукции (разработка, производство, эксплуатация, хранение, перевозка, реализация и утилизация);

·оказание услуг;

·система управления качеством;

·система управления окружающей средой;

·юридические лица;

·персонал.

Субъектами оценки соответствия являются:

государство в лице уполномоченных государственных органов;

·аккредитованные органы по сертификации;

·аккредитованные испытательные лаборатории (центры);

·заявители на аккредитацию;

·заявители на подтверждение соответствия.

Законом [55] предусмотрено два вида оценки соответствия: аккредитация и подтверждение соответствия .К документам об оценке соответствия относятся:

·аттестат аккредитации;

·сертификат соответствия;

·декларация о соответствии;

·сертификат компетентности.