



Жизненный цикл программного обеспечения

Жизненный цикл (<u>life cycle</u>) можно представить как ряд событий, происходящих с системой в процессе ее создания и использования.

Под жизненным циклом ПО (software life cycle) понимают весь период его разработки и эксплуатации (использования), начиная от момента возникновения замысла ПО и кончая прекращением всех видов его использования.

Процесс создания ИС делится на этапы.



Обычно выделяют следующие **этапы ЖЦ ПО**:

- формирование требований к системе (планирование),
- проектирование,
- реализация (кодирование,
- тестирование,
- ввод в действие,
- эксплуатация и сопровождение.

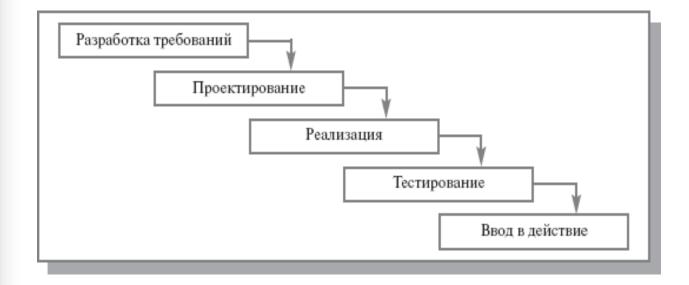


Модели жизненного цикла:

- Каскадная модель.
- Поэтапная модель с промежуточным контролем.
- Спиральная модель.



Рис.1. Каскадная модель ЖЦ ИС





Каскадная модель ЖЦ ИС

Положительные стороны *каскадного подхода*:

- на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
- выполняемые в логической последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты.

Недостаток *каскадного подхода:*

 реальный процесс создания системы никогда полностью не укладывается в такую жесткую схему.

В результате реальный процесс создания ИС оказывается соответствующим <u>поэтапной модели с промежуточным контролем</u>.



Рис. 2. Поэтапная модель с промежуточным контролем

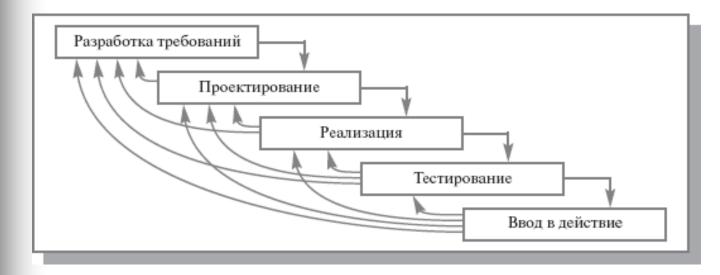
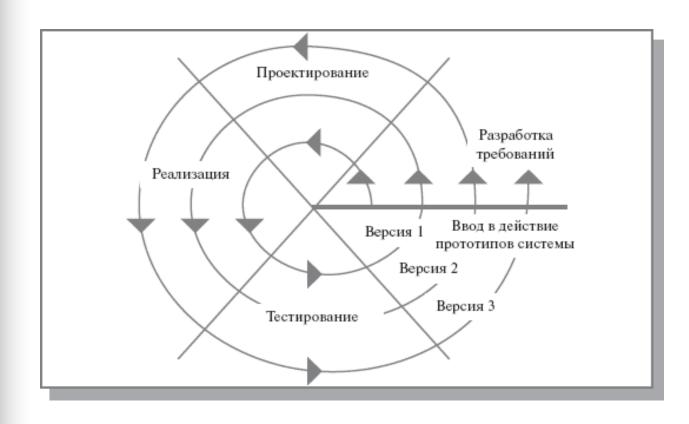




Рис.3. Спиральная модель ЖЦ ИС





Спиральная модель ЖЦ ИС

Положительные стороны <u>спиральной модели ЖЦ</u>:

- Каждый виток спирали соответствует созданию работоспособного фрагмента или версии системы.
- позволяет уточнить требования, цели и характеристики проекта, определить качество разработки, спланировать работы следующего витка спирали.
- углубляются и последовательно конкретизируются детали проекта и в результате выбирается обоснованный вариант.

Недостаток <u>спиральной модели</u>:

 проблема в определении момента перехода на следующий этап.





Стандарты регламентируют ЖЦ ПО, а в некоторых случаях и процессы его разработки.

Среди наиболее известных стандартов:

- **FOCT** xxxxx-yy
- ISO/IEC xxxxx:yyyy
- Custom Development Method (методика Oracle)
- Rational Unified Process (RUP)
- Microsoft Solution Framework (MSF)
- Extreme Programming (XP).



Базовые стандарты в области жизненного цикла ПО

- 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 РоссииГОСТ Р ИСО/МЭК 12207–99



Стандарты в области ЖЦ ПО

ОСНОВНЫЕ ТЕРМИНЫ



Ammecmaция (validation):

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

В процессе проектирования и разработки аттестация связана с экспертизой продукта в целях определения его <u>соответствия потребностям</u> (требованиям) <u>пользователя</u>.

Ay∂um (audit):

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



Базовая линия (baseline):

официально принятая версия элемента конфигурации, независимая от среды, формально обозначенная и зафиксированная в конкретный момент времени жизненного цикла элемента конфигурации.

Beрификация (verification):

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

В процессе проектирования и разработки верификация связана с экспертизой результатов данной работы для определения их <u>соответствия установленным к данной работе требованиям</u>.



Квалификационное испытание (qualification testing):

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 (qualification requirement):

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.



Контекст использования (context of use):

пользователи, задания, среда (аппаратное обеспечение, программное обеспечение и материалы), а также физические и социальные среды, в которых используется данное программное средство.

Обеспечение качества (quality assurance):

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.



 СИСТЕМА УПРАВЛЕНИЯ КАЧЕСТВОМ (система менеджмента качества, система качества, quality management system, quality system):

часть общей системы управления, включающая организационную структуру, планирование, ответственность, методы, процедуры, процессы, ресурсы, необходимые для обеспечения качества продукции и (или) услуг.

Как правило, система управления качеством является частью системы управления предприятием или организацией.



Тестируемость (testability):

степень, до которой могут быть запланированы объективность и реализуемость тестирования, проверяющего соответствие требованию.

Тестовое покрытие (test coverage):

степень, до которой с помощью контрольных примеров проверяют требования к системе или программному продукту.

Техническое задание (statement of work):

документ, используемый заказчиком в качестве средства для описания и определения задач, выполняемых при реализации договора.

Стандарт **СТБ ИСО/МЭК 12207-2003**

ЖЦ ПО имеет *трехуровневую* иерархическую структуру





Трехуровневая иерархическая структура жизненного цикла

основу составляют **процессы**,

которые разделены на работы,

которые разделены на задачи.



Графическое представление стандартов ЖЦ ПС





Процессы делятся на *три группы*:

- основные;
- вспомогательные;
- организационные.





Процессы ЖЦ ПС





Основные процессы жизненного цикла

– это процессы, которые реализуются под управлением основных сторон, участвующих в ЖЦ ПС.



Вспомогательные процессы жизненного цикла

 – это процессы, являющиеся целенаправленными составными частями других процессов.



Организационные процессы жизненного цикла

- это процессы, предназначенные для создания в некоторой организации и совершенствования организационных структур, охватывающих процессы ЖЦ и соответствующий персонал.





Основные понятия

В большинстве случаев (при отсутствии спецификации) критерием качества служит то, насколько пользователи удовлетворены ПП и сопутствующими услугами.

Качество (quality) ПО — это совокупность его черт и характеристик, которые влияют на его способность удовлетворять заданные потребности пользователей.



Стандартизация качества программного обеспечения в Республике Беларусь

- стандарт СССР ГОСТ 28806-90.
 Качество программных средств.
 Термины и определения;
- межгосударственный стандарт стран СНГ ГОСТ 28195–99.
 Оценка качества программных средств. Общие положения;
- национальный стандарт Беларуси
 СТБ ИСО/МЭК 9126—2003.
 Информационные технологии. Оценка программной продукции. Характеристики качества и руководства по их применению.



Терминология по стандартам:

ГОСТ 28806-90,

СТБ ИСО/МЭК 9126-2003,

ISO/IEC 9126-1-4:2001-2004,

ISO/IEC 14598-1:1999

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ



Ampuбym (attribute):

измеримое физическое или абстрактное *свойство* продукта.

Измерение (measurement):

использование метрики для присвоения *атрибуту* продукта значения (числа или категории) из *шкалы*.

Индикатор (indicator):

мера, которая может использоваться для оценки или прогнозирования другой меры.

Mepa (measure):

число или категория, присваиваемая атрибуту продукта путем измерения.



Mempuka (metric):
 определенный метод и шкала измерения.

Метрики могут быть внутренними, внешними или метриками качества в использовании.



Метрики могут прямыми или косвенными.

- Мера косвенная (indirect measure):
 мера атрибута, которая получена из
 мер одного или большего числа
 других атрибутов.
- **Мера прямая** (direct measure): мера атрибута, которая не зависит от меры любого другого атрибута.



- Внешнее качество (external quality): <u>степень</u>, в которой продукт удовлетворяет установленные и подразумеваемые потребности при использовании в заданных условиях.
- Внешняя мера (external measure): косвенная мера продукта, полученная из мер поведения системы, частью которой он является.
- Внутреннее качество (internal quality): полный набор атрибутов продукта, определяющих его способность удовлетворять установленные и подразумеваемые потребности при использовании в заданных условиях.
- Внутренняя мера (internal measure): собственная мера продукта, прямая или косвенная.



■ Качество в использовании (quality in use): степень, в которой ПП, используемый заданными пользователями, удовлетворяет их потребности в достижении заданных целей с результативностью, продуктивностью, безопасностью и удовлетворенностью в заданном контексте использования.

■ КОНТЕКСТ ИСПОЛЬЗОВАНИЯ (context of use): пользователи, задания, среда (аппаратное обеспечение, ПО и материалы), а также физические и социальные среды, в которых используется данный ПП.



- Критерий оценки качества (software quality assessment criteria): совокупность принятых в установленном порядке правил и условий, с помощью которых устанавливается приемлемость общего качества ПП.
- Оценка качества (quality evaluation): систематическое исследование степени, в которой продукт способен к выполнению указанных требований.
- *Модель качества* (quality model): набор характеристик и связей между ними, обеспечивающий основу для определения требований к качеству и для оценки качества.
- Показатель качества (software quality feature)
 признак, определяющий свойство ПП, которое может быть соотнесено с некоторой характеристикой качества.



- Подразумеваемые потребности (implied needs): потребности, которые не были установлены, но являются действительными потребностями при использовании ПП в конкретных условиях.
- Уровень качества функционирования (уровень пригодности, level of performance): степень удовлетворения потребности, представленная конкретным набором значений характеристик качества.
- Характеристика качества (software quality characteristic): набор свойств ПП, с помощью которых описывается и оценивается его качество.



Основные определения

■ Шкала (scale):

набор значений с определенными свойствами.

Типы шкал, используемые при оценке качества:

- номинальная;
- порядковая (упорядоченная);
- интервальная;
- относительная.



Классификация методов определения показателей качества

- по способам получения информации о показателе качества:
 - измерительный;
 - регистрационный;
 - органолептический;
 - расчетный;
- по источникам получения информации о показателе качества:
 - экспертный;
 - социологический;
 - традиционный.



Оценка качества ПП (по ГОСТ) -

совокупность операций, включающих:

- выбор номенклатуры характеристик качества оцениваемого программного средства,
- определение значений этих характеристик
- и сравнение их с базовыми значениями.



Показатель качества продукции –

количественная характеристика одного (или нескольких) из свойств продукции, составляющих ее качество, которая рассматривается применительно к определенным условиям ее создания и эксплуатации или потребления.





 Стандарты регламентируют выполнение оценки качества на основе иерархической модели качества.

 Модель описывает совокупность свойств ПП, отражающих его качество, в виде многоуровневой структуры.



Иерархическая модель качества

 Характеристики на первом (верхнем) уровне соответствуют основным свойствам ПП.

 Далее характеристики каждого уровня оцениваются посредством характеристик последующих уровней (подхарактеристиками).



Метод интегральной оценки качества ПП

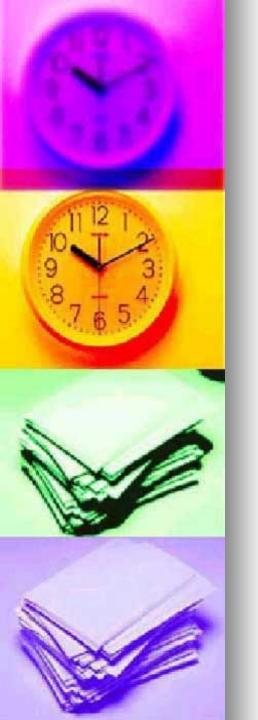
В основе метода предлагается четырехуровневая иерархическая модель качества:

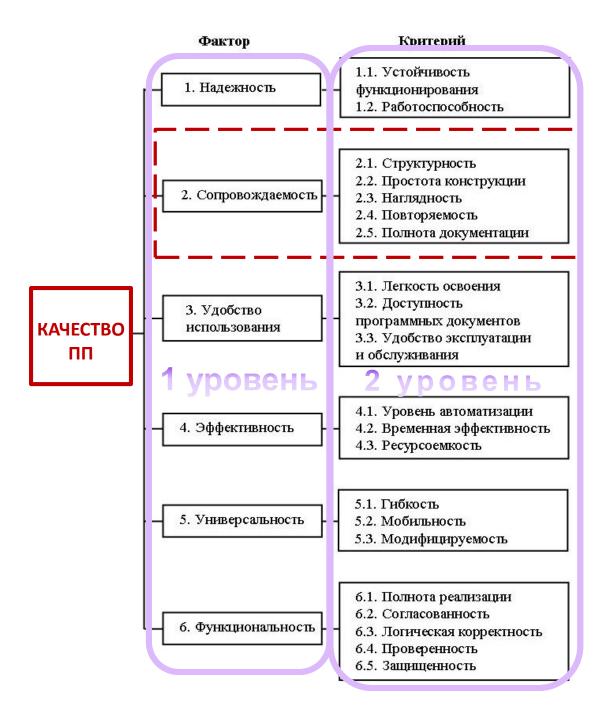
- факторы качества (характеристики качества);
- критерии качества (подхарактеристики качества);
- Метрики (аналог);
- оценочные элементы, или единичные показатели (отсутствует).



Основные характеристики качества:

- Функциональность (Functionality)
- Надежность (Reliability)
- Удобство использования (практичность, Usability)
- Эффективность (Efficiency)
- Сопровождаемость (Maintainability)
- Мобильность (Portability).

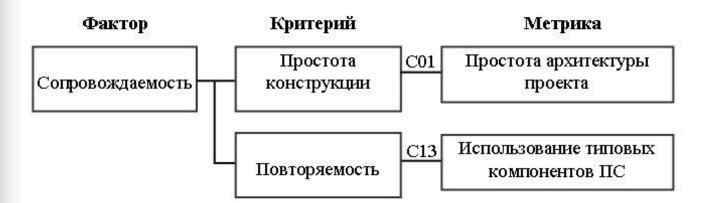






фаза анализа

1 уровень 2 уровень 3 уровень



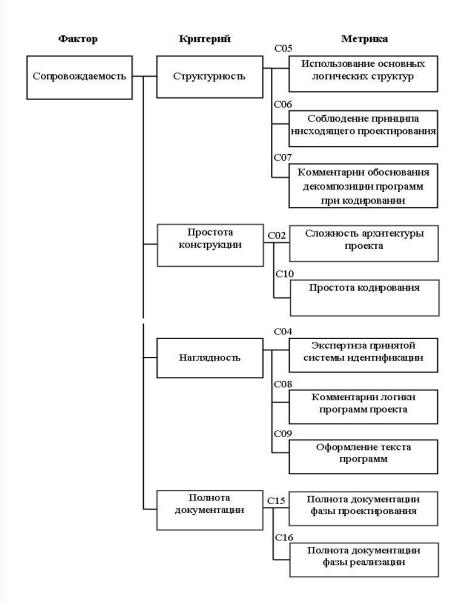


фаза проектирования



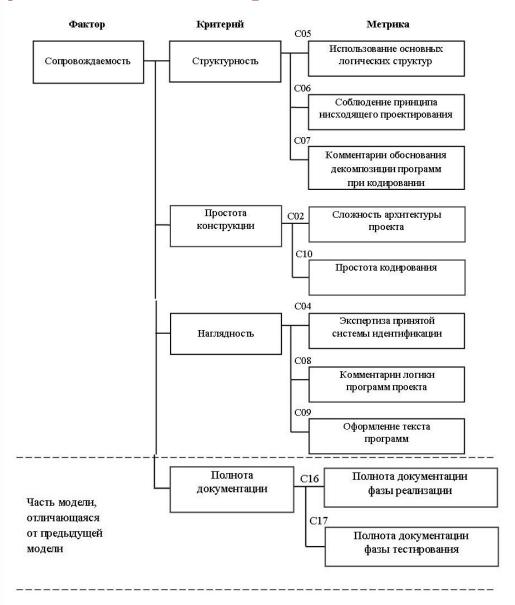


фаза реализации



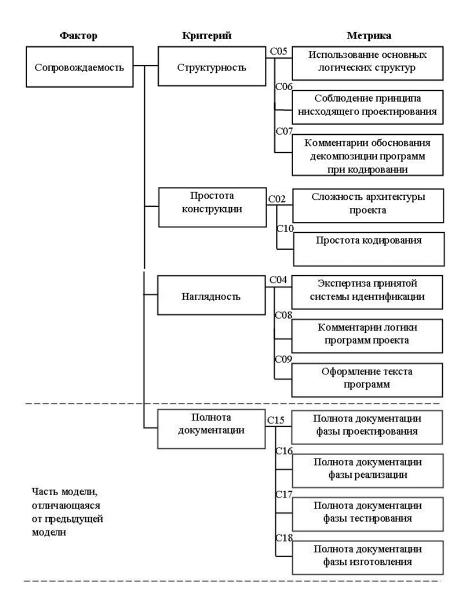


фаза тестирования



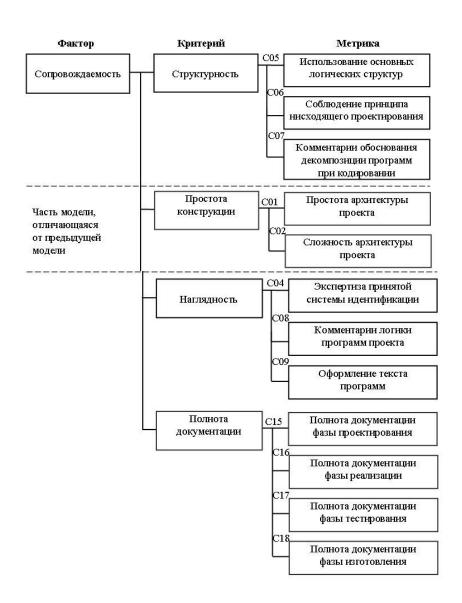


Фаза изготовления





Фаза сопровождения





Оценочные элементы метрик

4 уровень

Для выбора оценочных элементов ГОСТ 28195-99 предлагает перечень таблиц, содержащих наименование элемента, метод оценки и применяемость элемента для различных подклассов.



Код оценочного элемента

Пять символов: **SNNPP**

■ Первый символ (буква) **S** указывает на принадлежность элемента фактору (в ГОСТ 28195—99 приняты обозначения:

H – Надежность; **C** – Сопровождаемость;

У – Удобство использования; **Э** – Эффективность;

 Γ — Универсальность; K — Функциональность).

- Два следующих символа N N номер метрики,
 которой принадлежит оценочный элемент.
- Четвертый и пятый символы Р Р порядковый номер данного оценочного элемента в метрике.



Пример оценочных элементов фактора «Сопровождаемость»

Код элемента	Наименование	Метод оценки
C0101	Наличие модульной схемы программы	Экспертный
C0102	Оценка программы по числу уникальных модулей	Экспертный
C0201	Наличие ограничений на размеры модуля	Экспертный
C0301	Наличие проверки корректности передаваемых данных	Экспертный
C0302	Оценка простоты программы по числу точек входа и выхода W: $\mathbf{W} = \frac{1}{(\mathbf{D} + 1) \cdot (\mathbf{F} + 1)},$ где D – общее число точек входа в программу; F – общее число точек выхода из программы	Измерительный + Расчетный
C0303	Осуществляется ли передача результатов работы модуля через вызывающий его модуль	Экспертный
C0304	Осуществляется ли контроль за правильностью данных, поступающих в вызывающий модуль от вызываемого	Экспертный
C0305	Наличие требований к независимости модулей программы от типов и форматов выходных данных	Экспертный



FOCT 28195-99

метод интегральной оценки качества:

Выбор номенклатуры показателей качества для конкретного продукта осуществляется с учетом:

- назначения ПП,
- требований области применения,
- зависимости от принадлежности ПП к тому или иному подклассу (по классификатору продукции, ОКП).



- Абсолютные и относительные значения і-го фактора качества определяются отношением суммы показателей N критериев (абсолютных или относительных), которые относятся к этому фактору, с учетом их коэффициентов к сумме всех весовых коэффициентов этих критериев.
- Общая оценка качества в целом формируется экспертами по набору полученных значений факторов качества.



Модель процесса оценки по СТБ ИСО/МЭК 9126–2003

Процесс оценки может применяться после любой подходящей работы ЖЦ для каждого компонента ПП.

Данная модель отражает основные стадии и этапы, требуемые для оценки качества ПП.

Недостатки метода:

отсутствие рекомендуемых вариантов метрик и представление метода лишь в общем виде (в виде модели).





Основа регламентирования характеристик качества ПС в мире

международный стандартISO/IEC 9126:1991



Стандарты серии **ISO/IEC** в области оценки качества ПС

международный стандарт

ISO/IEC 9126:1991 -

Информационная технология – Оценка программного продукта

Характеристики качества и руководства по их применению

■ ISO/IEC 9126-1-4

■ ISO/IEC 14598-1-6



- Первая часть стандарта ISO/IEC 9126—1:2001 по существу является пересмотренной редакцией стандарта ISO/IEC 9126:1991. В данной части определены два верхних уровня (характеристики и подхарактеристики) иерархической модели качества, приведены общие требования к метрикам качества, даны рекомендации по их выбору. При этом сохранена та же номенклатура из шести базовых характеристик качества ПС.
- В отличие от *ISO/IEC 9126:1991* подхарактеристики второго уровня стали нормативными, а не рекомендуемыми, определены две части модели качества (модель внутреннего и внешнего качества и модель качества в использовании) и исключен процесс оценки качества (он теперь содержится в стандарте *ISO/IEC 14598*).
- *Во второй части стандарта ISO/IEC TR 9126—2:2003* определяются метрики количественного измерения внешнего качества ПС. *Внешние метрики*.
- *В третьей части стандарта ISO/IEC TR 9126—3:2003* определяются метрики количественного измерения внутреннего качества ПС. *Внутренние метрики*.
- В четвертой части стандарта ISO/IEC TR 9126-4:2004 определяются метрики количественного измерения качества в использовании. Метрики качества в использовании.



- *В первой части стандарта ISO/IEC 14598—1:1999* приведен обзор остальных частей, определена связь *ISO/IEC 14598* со стандартами *ISO/IEC 9126—1—4* и *ISO/IEC 12207:1995*. В данной части содержатся общие требования к спецификации и оценке качества, разъясняются концепции оценки.
- *Вторая часть стандарта ISO/IEC 14598–2:2000* содержит требования и руководство по поддержке оценки. Эта часть стандарта предназначена для применения на уровне организации или ее подразделений.
- *Третья часть стандарта ISO/IEC 14598–3:2000* предназначена для организаций разработчиков ПС. Данная часть ориентирована на выполнение оценки ПП, используя собственный технический персонал.
- *Четвертая часть стандарта ISO/IEC 14598–4:1999* предназначена для организаций, которые планируют приобретать готовый или разрабатываемый программный продукт. В ней определена связь работ процесса заказа из стандарта *ISO/IEC 12207:1995* с работами, выполняемыми при оценке ПП.
- Пятая часть стандарта ISO/IEC 14598—5:1998 предназначена для использования оценщиком, выполняющим независимую оценку программного продукта. Как правило, персонал оценки работает в независимой организации. В данной части приводятся концепции оценки и требования к процессу оценки. Рассмотрена структура отчета об оценке. Приводятся рекомендации по выбору уровней ранжирования при проведении измерений. Оценка качества программного продукта оценщиком может выполняться по запросу разработчика, заказчика (покупателя) или другой стороны.
- Шестая часть стандарта ISO/IEC 14598—6:2001 предназначена для поддержки оценки программного продукта и содержит руководство по документированию модулей оценки.



Модуль оценки представляет собой полностью укомплектованную информацию, необходимую для проведения процесса оценки некоторой характеристики или подхарактеристики качества.

Модуль содержит спецификацию соответствующей модели качества (характеристика, подхарактеристики, внутренние или внешние метрики качества), методики и процедуры оценки, входные данные, связанные с оценкой, информацию о запланированном применении модели и о ее фактическом применении, структуру типового отчета о результатах выполненной оценки.



Связь качества ПП с его жизненным циклом

- В течение жизненного цикла программного средства его качество изменяется.
- Требуемое качество, определенное в начале ЖЦ, отмичается от фактического качества поставленного продукта.
- Существует несколько точек зрения на качество ПС в течение его ЖЦ.



внутреннее качество –

совокупность характеристик ПП с внутренней точки зрения;

внутреннее качество измеряется с помощью внутренних метрик и оценивается по отношению к требованиям к внутреннему качеству;



оценочное (или прогнозируемое) внешнее качество –

оцененное или предсказанное качество конечного программного продукта на каждой стадии процесса разработки для каждой характеристики качества, основанное на знании внутреннего качества;



внешнее качество –

совокупность характеристик программного продукта с внешней точки зрения;

это качество, измеряемое и оцениваемое на основе внешних метрик при выполнении ПП во время тестирования (испытаний) в моделируемой среде с моделируемыми данными или во время эксплуатации



оценочное (или прогнозируемое) качество в использовании –

оцененное или предсказанное качество конечного программного продукта на каждой стадии процесса разработки для каждой характеристики качества в использовании, основанное на знании внутреннего и внешнего качества;



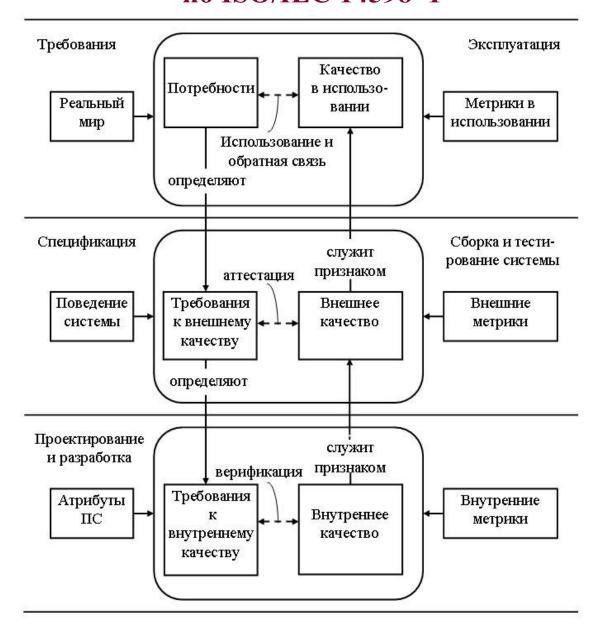
качество в использовании –

качество программного продукта, применяемого в заданной среде и заданном контексте использования, с точки зрения пользователя;

оно оценивается на основе метрик качества в использовании и в первую очередь измеряет степень достижения пользователем своих целей в конкретной среде, а не свойства самого ПП; пользователь оценивает только те атрибуты программного продукта, которые он применяет в своих задачах.



Качество в жизненном цикле программных средств по ISO/IEC 14598-1





Тип оцениваемого промежуточного или конечного программного продукта зависит от цели оценки и стадии ЖЦ ПС



Варианты ранжирования измеренных значений метрик по ISO/IEC 14598–1:1999





На основании оценки и организационных критериев принимается административное решение относительно принятия или отклонения, выпуска или невыпуска программного продукта.







Свойства и критерии обоснованности метрик качества ПС

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/IEC TR 9126–2–4* 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 должна также быть эффективной по отношению к стоимости.



Критерии обоснованности метрики:

- Q (T) характеристика качества ПС в момент времени T,M (T) значение метрики в момент времени T.
- 1) Корреляция;
- 2) Трассировка;
- 3) Непротиворечивость;
- 4) Предсказуемость;
- 5) прогнозное Q(T2) фактическое Q(T2) фактическое Q(T2)



Описание метрик качества:

- 1) название метрики,
- 2) назначение метрики,
- 3) метод применения,
- 4) способ измерения, формула, исходные и вычисляемые данные,
- 5) интерпретация измеренного значения,
- 6) тип шкалы, используемой при измерении метрики,
- 7) тип измеренного значения,
- в) источники входных данных для измерения,
- 9) ссылка на стандарт,
- 10) целевая аудитория.



Совместное использование различных метрик

Для обеспечения возможности совместного использования различных метрик при количественной оценке качества ПП метрики в стандартах *ISO/IEC TR 9126–2–4* представляются в относительных единицах в виде двух формул

$$X = A/B \tag{1}$$

ИЛИ

$$X = 1 - A/B \tag{2}$$

эбѕ

X – значение метрики;

A – абсолютное (измеренное) значение некоторого свойства (атрибута) оцениваемого продукта или документации;

В – базовое значение соответствующего свойства.



 Из двух формул для конкретной метрики выбирается та, которая соответствует критериям трассировки и непротиворечивости.

$$0 \le X \le 1$$



Внутренние метрики качества ПС

- Внутренние метрики функциональности.
- Внутренние метрики надежности.
- Внутренние метрики практичности.
- Внутренние метрики эффективности.
- Внутренние метрики сопровождаемости.
- Внутренние метрики мобильности.



Внешние метрики качества ПС

- Внешние метрики функциональности.
- Внешние метрики надежности.
- Внешние метрики практичности.
- Внешние метрики эффективности.
- Внешние метрики сопровождаемости.
- Внешние метрики мобильности.



Метрики качества ПС в использовании

Метрики результативности.

достигают ли задачи, выполняемые пользователем, заданных целей с точностью и полнотой в заданном контексте использования

Метрики продуктивности.

ресурсы, которые затрачивают пользователи в соответствии с достигнутой результативностью ...

Метрики безопасности.

уровень риска причинения вреда людям, бизнесу, программному обеспечению, имуществу или окружающей среде ...

Метрики удовлетворенности.

отношение пользователя к использованию продукта ...



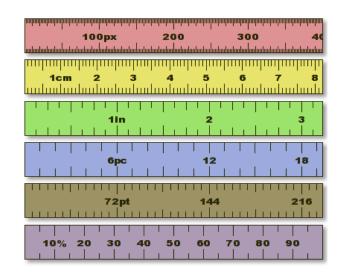


Понятие метрики

Метрики программных проектов

 это количественные показатели, отражающие их отдельные характеристики.

Или, метрики представляют собой все, что можно измерить, при условии, конечно, что в этом есть смысл.





В исследовании метрик ПО различают два (2) основных направления :

- поиск метрик, характеризующих наиболее специфические свойства программ, т.е. метрик оценки самого ПО;
- использование метрик для оценки технических характеристик и факторов разработки программ, т.е. метрик оценки условий разработки программ.



По виду информации, получаемой при оценке качества ПО, метрики можно разбить на три (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

ШЕСТЬ (6) 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 три (3) основные группы:

- 1) метрики *размера* программ
- метрики сложности потока управления программ
- в) метрики сложности потока данных программ



РАЗМЕРНО— ОРИЕНТИРОВАННЫЕ МЕТРИКИ



Метрики размера программ

Традиционной характеристикой размера программ является

количество строк исходного текста (кода)

(Source Lines Of Code, SLOC).

Под строкой понимается любой оператор программы.



Необходимо помнить: метрика SLOC не отражает трудоемкости по созданию программного обеспечения.



МЕТРИКА ХОЛСТЕДА

Основу метрики составляют 4 измеряемых характеристики программы:

- n1 число уникальных операторов программы, включая символыразделители, 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

длину программы

$$N = N1 + N2_{(e\partial u + u u)}$$
 (1)

объем программы

$$V = N*log2(n) (2)$$



Сложность программы:

$$Hdiff = (n1 / 2)* (N2/n2)$$

Трудоемкость разработки:

$$E = \frac{n_1 \cdot N_2 \cdot N \cdot \log_2 n}{2 \cdot n_2}$$



ABC-метрика (Fitzpatrick)

Основана на подсчете

- присваиваний значений переменным (Assignment),
- явных передач управления за пределы области видимости, т.е. вызовов функций (Branch),
- и логических проверок (Condition).

Мера записывается тройкой значений <A,B,C>

Но для оценки сложности программы вычисляется одно число, как квадратный корень из суммы квадратов A, B, C.



МЕТРИКИ СЛОЖНОСТИ *ПОТОКА* **УПРАВЛЕНИЯ**



Структурная сложность программ

Структурная сложность определяется:

- числом взаимодействующих компонент;
- числом связей между компонентами;
- сложностью взаимодействия компонент.



Структурная сложность программ

При функционировании программы разнообразие ее поведения и разнообразие связей между ее входными и результирующими данными определяется набором маршрутов, по которым она исполняется.

Сложность программного модуля связана не столько с размером (числом команд) программы, сколько с числом маршрутов ее исполнения и их сложностью.



Для оценивания сложности потока управления программы (control flow graph) используется одна из самых распространенных метрик сложности – ЦИКЛОМАТИЧЕСКАЯ СЛОЖНОСТЬ, (Томас МакКейб, Thomas McCabe)



Цикломатическая сложность

Данная метрика вычисляется на основе ориентированного графа G=(V,E),

где

V - вершины, соответствующие операторам,

Е - дуги, соответствующие переходам.

Формула вычисления цикломатической сложности выглядит:

$$Z(G) = 1 - v + 2*p$$
 (3)

где

1 – число ребер (дуг) графа программы,

v – число узлов (вершин) графа,

р – число компонентов связности графа.



Число компонентов связности ориентированного графа можно рассматривать как количество дуг, которые необходимо добавить для преобразования графа в сильносвязный.

Ориентированный граф является сильносвязным, если для каждой пары вершин Vi и Vj существует по крайней мере один путь из Vi в Vj и по крайней мере один путь из Vj в Vi.



Цикломатическое число Маккейба

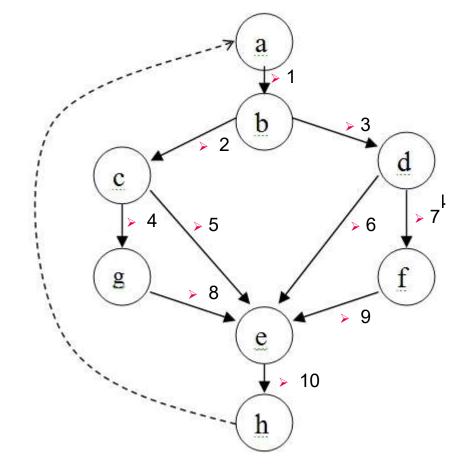
По сути **Z(G)** определяет число линейнонезависимых контуров в сильносвязном графе.

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 «покрытие ветвей».



Пример

$$n = 1$$



 Для программы цикломатическое число при I=10, v=8, n=1 определится как

$$Z(G) = 10-8+2 = 4$$



МЕТРИКА ДЖИЛБА

Простая, но достаточно эффективная метрика определения сложности программы — подсчет насыщенности программы условными операторами.



МЕТРИКА ДЖИЛБА

Вводятся две характеристики:

- CL абсолютная сложность программы, характеризующаяся количеством операторов условия;
- cl <u>относительная сложность программы</u>, характеризующаяся насыщенностью программы операторами условия, *m. e. cl определяется как отношение CL к общему* числу операторов.

* под операторами условия понимают операторы, которые содержат условные выражения.



Расширение метрики Джилба

На основе метрики Джилба построена еще одна метрика – характеристика максимального уровня вложенности оператора (CLI).

Это позволило успешно применить метрику Джилба к анализу циклических конструкций.



МЕТРИКА ГРАНИЧНЫХ ЗНАЧЕНИЙ

Пусть **G=(V,E)** — *ориентированный связный граф* программы с единственной начальной и единственной конечной вершинами.

Отрицательная степень вершины – это число входящих в вершину дуг.

Положительная степень вершины — это число исходящих из вершины дуг.



Набор вершин графа *можно разбить на две группы*:

■ вершины у которых положительная степень <=1

(принимающие вершины)

 вершины у которых положительная степень >=2
 (вершины отбора)



Для получения оценки по методу **Г3** следует разбить граф G на максимальное число подграфов, удовлетворяющих условиям:

- вход в подграф осуществляется только через вершину отбора;
- каждый подграф включает вершину (называемую нижней границей подграфа), в которую можно попасть из любой другой вершины подграфа.

<u>Например</u>,

вершина отбора соединенная сама с собой дугой петлей, образует подграф.



МЕТРИКА ГРАНИЧНЫХ ЗНАЧЕНИЙ

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 (S_a).



МЕТРИКА ГРАНИЧНЫХ ЗНАЧЕНИЙ

Граничная сложность программы определяется как:

$$S_0 = 1 - (v-1) / S_a$$

где

- S_0 относительная граничная сложность программы;
- S_a абсолютная граничная сложность программы,
- v общее число вершин графа программы.



МЕТРИКИ СЛОЖНОСТИ ПОТОКА ДАННЫХ



метрика сложности потока данных – **спен**

Спен – это число утверждений, содержащих данный идентификатор, между его первым и последним появлением в тексте программы.

Идентификатор, появившийся п 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.



Суть метода состоит в оценке информационной прочности отдельно взятого программного модуля с помощью анализа характера использования переменных из списка ввода-вывода.



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

- P вводимые переменные для расчетов и для обеспечения вывода.
- M модифицируемые или создаваемые внутри программы переменные.
- С переменные, участвующие в управлении работой программного модуля (управляющие переменные).
- T не используемые в программе («паразитные») переменные.

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



Значение метрики Чепина рассчитывается по формуле:

Q = a1P + a2M + a3C + a4T

где *a1, a2, a3, a4* – весовые коэффициенты.



С учетом весовых коэффициентов выражение примет вид:

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