КОНСПЕКТ ЛЕКЦИЙ К РАЗДЕЛУ 2. Стандарты в области информационных систем

Содержание лекции:

1. Стандарты и полнота документации. Стадии разработки ПО, регламентированные ГОСТами
2. Математические модели оценки характеристик качества и надежности программного и информационного обеспечения

1. Стандарты и полнота документации. Стадии разработки ПО, регламентированные ГОСТами

В нашей стране жизненный цикл разработки ПО установлен стандартом ГОСТ 19.102-77 «Стадии разработки программ и программной документации» и содержит следующие стадии и этапы:

1. Техническое задание (ТЗ).
2. Эскизный проект (ЭП).
3. Технический проект (ТП).
4. Рабочий проект (РП).
5. Внедрение.

Техническое задание. На стадии Техническое задание выполняются следующие работы, входящие в состав соответствующих этапов.

1. Обоснование необходимости разработки программ: постановка задачи; сбор исходных материалов; выбор и обоснование критериев эффективности и качества; обоснование необходимости проведения НИР.

2. Выполнение научно-исследовательских работ: определение структуры входных и выходных данных; предварительный выбор методов реше-ния задач; обоснование целесообразности применения ранее разработанных программ; определение требований к техническим средствам; обоснование принципиальной возможности решения поставленных задач.

3. Разработка и утверждение технического задания: определение требований к программе; разработка технико-экономического обоснования раз работки программы; определение стадий, этапов и сроков разработки программы и документации на нее; выбор языков программирования; определение необходимости проведения НИР на последующих стадиях; согласование и утверждение ТЗ.

Результатом выполнения данной стадии является **техническое задание**, оформленное в соответствии с ГОСТ 19.105- 78 (изм. 09.1981) «Общие требования к программным документам» и ГОСТ 19.106-78 «Общие требования к программным документам, выполненным печатным способом на листах формата 11 и 12».

Эскизный проект. Конкретное содержание работ стадии эскизного проекта и их объем определяет степень сложности разрабатываемого ПО. Результатом выполнения данной стадии является полное описание архитектуры ПО. Как правило, это описание делается на нескольких уровнях иерархии. На верхнем уровне детализации выделяются основные подсистемы, которым присваиваются имена , устанавливаются связи между подсистемами , их функции, получаемые путем декомпозиции предполагаемых функций ПО.

Затем процедура декомпозиции выполняется для каждой подсистемы, выделяются модули, составляющие данную подсистему. В конечном итоге, получается иерархически организованная система, состоящая из уровней, каждый из которых представляет собой совокупность взаимосвязанных модулей. Единицы, выделяемые на различных иерархических уровнях функциональной архитектуры системы, определяются по усмотрению разработчика. Стандарты ЕСПД различают программные единицы только с точки зрения их документирования.

Результаты эскизного проекта отображаются в документе Пояснительная записка к эскизному проекту, оформленному в соответствии с ГОСТ 19.105-78 и ГОСТ 19.404-79. После утверждения пояснительной записки она становится программным документом, правила дублирования, учета, хранения которого определяется ГОСТ 19.601-78 «Общие правила дублирования, обращения, учета и хранения» и ГОСТ 19.602-78 «Правила дублирования, учета и хранения программных документов, выполненных печатным способом». Последующие стадии и этапы разработки ПО могут выявить необходимость внесения изменений в ЭП. Эти изменения должны быть отражены в пояснительной записке в соответствии с ГОСТ 19.603-78 «Общие правила внесения изменений в программные документы» и ГОСТ 19.602-78 «Правила внесения изменений в программные документы, выполненные печатным способом».

Технический проект. Содержанием работ на этой стадии является проектирование структуры ПО. Результатом – реализующий заданный и утвержденный в техническом задании комплекс программ как иерархическая структура программных модулей, заданных своими функциональными спецификациями. Форма представления результата – *пояснительная записка* к техническому проекту согласно ГОСТ 19.105 -78, ГОСТ 19.404-79. Разработка структуры ПО заключается в выделении всех программных ком-понентов по функциональным признакам, определение функциональных спецификаций модулей и уточнение внешних функциональных специфика-ций, структуры входных и выходных данных, определении операционной среды, языковых средств и конфигурации аппаратных средств. Спецификации модулей являются внешними характеристиками и содержат все сведения, необходимые вызывающим модулям. На последующих стадиях разработки спецификации оформляются в виде комментариев в начале текста исходной программы модуля. На данной стадии спецификации оформляются
в виде комментария на принятом в организации, занимающейся разработкой ПО, языке спецификаций

Оформление пояснительной записки и ведомости технического проекта ПО осуществляется в соответствии с ГОСТ 19.105-78, ГОСТ 19.404-79 и ГОСТ 2.106-68 ЕСКД «Текстовые документы».

Рабочий проект. Содержанием работ на этой стадии является описание ПО на выбранном проблемно-ориентированном языке (кодирование), отладка, разработка, согласование и утверждение порядка и методики испытаний, разработка программных документов, проведение тестирования, корректировка программ и программной документации по результатам тестирования, проведение приемо-сдаточных испытаний. Результат – ПО в форме программной документации, в форме документации на ПО или в форме программного изделия.

2. Математические модели оценки характеристик качества и надежности программного и информационного обеспечения

Качество программного обеспечения – способность программного продукта подтвердить свою спецификацию при условии, что спецификация ориентирована на характеристики, которые желает получить пользователь.

Одной из важнейших проблем обеспечения качества программных средств является формализация характеристик качества и методология их оценки. Основой регламентирования показателей качества программных средств ранее являлся международный стандарт ISO 9126:1991 (ГОСТ Р ИСО / МЭК 9126-93) «Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению».

Методологии и стандартизации оценки характеристик качества готовых программных средств и их компонентов (программного продукта) на различных этапах жизненного цикла посвящен международный стандарт ISO 14598, состоящий из шести частей. Рекомендуется следующая общая схема процессов оценки характеристик качества программ:

- установка исходных требований для оценки – определение целей испытаний, идентификация типа метрик программного средства, выделение адекватных показателей и требуемых значений атрибутов качества;
- селекция метрик качества, установление рейтингов и уровней приоритета метрик субхарактеристик и атрибутов, выделение критериев для проведения экспертиз и измерений;
- планирование и проектирование процессов оценки характеристик и атрибутов качества в жизненном цикле программного средства;
- выполнение измерений для оценки, сравнение результатов с критериями и требованиями, обобщение и оценка результатов.

Функциональная пригодность – наиболее неопределенная и объективно трудно оцениваемая субхарактеристика программного средства. Области применения, номенклатура и функции комплексов программ охватывают столь разнообразные сферы деятельности человека , что невозможно выде-

лить и унифицировать небольшое число атрибутов для оценки и сравнения этой субхарактеристики в различных комплексах программ.

Оценка корректности программных средств состоит в формальном определении степени соответствия комплекса реализованных программ исходным требованиям контракта, технического задания и спецификаций на программное средство и его компоненты. Путем верификации должно быть определено соответствие исходным требованиям всей совокупности к компонентов комплекса программ, вплоть до модулей и текстов программ и описаний данных.

Оценка способности к взаимодействию состоит в определении качества совместной работы компонентов программных средств и баз данных с другими прикладными системами и компонентами на различных вычислительных платформах, а также взаимодействия с пользователями в стиле, удобном для перехода от одной вычислительной системы к другой с подобными функциями.

Оценка защищенности программных средств включает определение полноты использования доступных методов и средств защиты программного средства от потенциальных угроз и достигнутой при этом безопасности функционирования информационной системы. Наиболее широко и детально методологические и системные задачи оценки комплексной защиты информационных систем изложены в трех частях стандарта ISO 15408:1999-1--3 «Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий».

Оценка надежности – измерение количественных метрик атрибутов субхарактеристик в использовании: завершенности, устойчивости к дефектам, восстанавливаемости и доступности/готовности.

Потребность в ресурсах памяти и производительности компьютера в процессе решения задач значительно изменяется в зависимости от состава и объема исходных данных. Для корректного определения предельной пропускной способности информационной системы с данным программным средством нужно измерить экстремальные и средние значения длительностей исполнения функциональных групп программ и маршруты, на которых они достигаются. Если предварительно в процессе проектирования производительность компьютера не оценивалась, то, скорее всего, понадобится боль-шая доработка или даже замена компьютера на более быстродействующий.

Оценка практичности программных средств проводится экспертами и включает определение понятности, простоты использования, изучаемости и привлекательности программного средства. В основном это качественная (и субъективная) оценка в баллах, однако некоторые атрибуты можно оценить количественно по трудоемкости и длительности выполнения операций при

использовании программного средства, а также по объему документации, необходимой для их изучения.

Сопровождаемость можно оценивать полнотой и достоверностью документации о состояниях программного средства и его компонентов, всех предполагаемых и выполненных изменениях, позволяющей установить текущее состояние версий программ в любой момент времени и историю их развития. Она должна определять стратегию, стандарты, процедуры, распределение ресурсов и планы создания, изменения и применения документов на программы и данные.

Оценка мобильности – качественное определение экспертами адаптируемости, простоты установки, совместимости и замещаемости программ, выражаемое в баллах. Количественно эту характеристику программного средства и совокупность ее атрибутов можно (и целесообразно) оценить в экономических показателях: стоимости, трудоемкости и длительности реализации процедур переноса на иные платформы определенной совокупности программ и данных.

Выбор характеристик и оценка качества программных средств - лишь одна из задач в области обеспечения качества продукции, выпускаемой компаниями - разработчиками ПО. Комплексное решение задач обеспечения качества программных средств предполагает разработку и внедрение той или иной системы управления качеством. В мировой практике наибольшее распространение получила система, основанная на международных стандартах серии ISO 9000, включающей десяток с лишним документов, в том числе стандарт, регламентирующий обеспечение качества ПО (ISO 9000/3). Эти стандарты должны служить руководством для ведущих специалистов компаний, разрабатывающих ПО на заказ.

Определения характеристик и субхарактеристик качества (ISO 9126-1)

Функциональные возможности – способность программного средства обеспечивать решение задач, удовлетворяющих сформулированные потребности заказчиков и пользователей при применении комплекса программ в заданных условиях.

Функциональная пригодность – набор и описания субхарактеристики и ее атрибутов, определяющие назначение, номенклатуру, основные, необходимые и достаточные функции программного средства, соответствующие техническому заданию и спецификациям требований заказчика или потенциального пользователя.

Правильность (корректность) – способность программного средства обеспечивать правильные или приемлемые для пользователя результаты и внешние эффекты.

Способность к взаимодействию – свойство программных средств и их компонентов взаимодействовать с одной или большим числом компонентов внутренней и внешней среды.

Защищенность – способность компонентов программного средства защищать программы и информацию от любых негативных воздействий.

Надежность – обеспечение комплексом программ достаточно низкой вероятности отказа в процессе функционирования программного средства в реальном времени.

Эффективность – свойства программного средства, обеспечивающие требуемую производительность решения функциональных задач, с учетом количества используемых вычислительных ресурсов в установленных условиях.

Практичность (применимость) – свойства программного средства, обусловливающие сложность его понимания, изучения и использования, а также привлекательность для квалифицированных пользователей при применении в указанных условиях.

Сопровождаемость – приспособленность программного средства к модификации и изменению конфигурации и функций.

Мобильность – подготовленность программного средства к переносу из одной аппаратно-операционной среды в другую.

Надёжность программного обеспечения

Надежность программного обеспечения – способность программного продукта безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой **вероятностью**. *Степень надежности* характеризуется **вероятностью** работы программного продукта без отказа в течение определенного периода времени.

Случайное изменение исходных данных и накопленной при обработке информации, множество условных переходов в программе создают такое огромное количество различных маршрутов исполнения программы, что их нельзя протестировать полностью из-за ограниченного времени отладки и испытаний. Источниками ненадёжности являются непроверенные сочетания исходных данных, при которых отлаженные программы дают неверные результаты и отказы.

Отказ – это нарушение работоспособности программного изделия, являющееся следствием таких явлений, как нарушения кодов записи программ в памяти, стирания или искажения данных в оперативной или долговременной памяти, нарушения нормального хода вычислительного процесса.

Сбой – это самоустраняющийся отказ , не требующий внешнего вмешательства. Основное различие между сбоем и отказом – по временному показателю длительности восстановления. Если длительность восстановления больше какого-то порогового значения, то аномалию в работе программы относят к отказам, иначе – к сбоям.

Понятие правильной (корректной) программы рассматривается статически вне временного функционирования. Неправильность программы определяется вероятностью совмещения следующих событий: попадания исходных данных в область непроверенных при отладке и испытаниях; проявления ошибки в программе при обработке таких данных.

Правильность программы не зависит от динамики функционирования в реальном времени.

Надёжная программа должна обеспечивать низкую **вероятность** отказа в процессе функционирования. Количество ошибок в программе, как оказалось, не имеет никакого отношения к ее надежности:

1. Число ошибок в программе – величина «ненаблюдаемая», наблюдаются не сами ошибки, а результат их проявления.

2. Неверное срабатывание программы может быть следствием не одной, а сразу нескольких ошибок.

3. Ошибки могут компенсировать друг друга, так что после исправления какой-то одной ошибки программа может начать «работать хуже».

4. Надежность характеризует частоту проявления ошибок, но не их количество; в то же время хорошо известно, что ошибки проявляются с разной часто-той: некоторые ошибки остаются не выявленными после многих месяцев и даже лет эксплуатации, но, с другой стороны одна единственная ошибка мо-жет привести к неверному срабатыванию программы при любых исходных данных, т.е. к нулевой надежности.

Теперь подробнее остановимся на характеристиках безопасности, которые определяют стандарты.

Характеристика как внутреннего, так и внешнего качества «защищённость» используется в стандартах в значении: способность ПП защищать информацию и данные так, чтобы неавторизованные субъекты или процессы не смогли читать или модифицировать их, а авторизованным пользователям и процессам не было отказано в доступе к ним. В стандартах подчеркивается, что данное требование также относится и к данным, которые находятся в процессе пересылки.

Характеристика качества «безопасность» вводится как характеристика качества в использовании, данная характеристика определяет способность ПП достигать приемлемого уровня риска для здоровья людей, их бизнеса, ПО, имущества или окружающей среды при данном способе (контексте) применения.

Если учесть, что информационная безопасность включает в себя вопросы обеспечения целостности, конфиденциальности и доступности, то можно прийти к выводу, что при комплексной оценке безопасности ПП и его использования в соответствии с вышеназванными стандартами нельзя ограничиться только этими характеристиками. Необходимо также учесть (полностью или частично) такие характеристики как «надёжность», «сопровождаемость», а также ряд смежных субхарактеристик (в составе других характеристик), которые косвенно затрагивают атрибуты безопасности ПП и системы, в которой он используется.

Допустим, мы определились с теми характеристиками и субхарактеристиками, которые мы собираемся оценивать для разрабатываемого ПП. Следующий этап, это собственно «язык чисел». Стандарт определяет набор метрик, при помощи которых можно численно оценить каждую субхарактеристику.

Для иллюстрации подхода ниже будут приведены метрики субхарактеристик внешнего и внутреннего качества «защищённость» и качества в использовании «безопасность». В стандарте рекомендуется разрабатывать собственные метрики, в которых будет более точно учтены проблемы безопасности, затрагивающие ПС.

Таблица – Метрики субхарактеристик внешнего и внутреннего качества «защищённость» и качества в использовании «безопасность»:

Метрика	Формула	Примечания из стандарта

1. Внешние метрики безопасности: 1.1 протоколирование доступа;	X = A / B; A = число «фактов доступа пользователя к системе и данным», зафиксированных в протоколе системы; B = число «фактов доступа пользователя к системе и данным», которые были произведены во время оценки;	1. рекомендуется использовать «тесты на проникновения» для эмуляции атак на систему; 2. под записью протокола «факт доступа пользователя к системе и данным» может подразумеваться запись «факт обнаружения вируса» для обеспечения антивирусной безопасности системы; 3. метрика носит экспериментальный характер;
1.2 контролируемость доступа;	X = A / B; A = число обнаруженных видов несанкционированного доступа; B = число видов несанкционированного доступа в спецификации;	1. необходимо проверить способность системы определять факты несанкционированного доступа при неправильном применении функций системы; 2. рекомендуется использовать «тесты на проникновения» для эмуляции атак на систему; 3. метрика носит экспериментальный характер;
1.3 предотвращение повреждения данных;	a) X = 1 – A / N; A = число фактов существенного повреждения данных; N = число видов тестов, при помощи которых пытались спровоцировать факт повреждения данных; b) Y = 1 – B / N; B = число фактов незначительного повреждения данных; c) X = A / T или B / T; T = время выполнения операции;	1. необходимо проверить корректность работы системы при неправильном применении её функций; 2. необходимо построить классификацию эффекта от событий повреждения данных; 3. для вычисления внешних метрик следует использовать информацию, доступную извне системы. Порядок подсчёта событий здесь отличается от порядка подсчёта событий при оценке аналогичных внутренних данных; 4. рекомендуется использовать «тесты на проникновения» для эмуляции атак на систему; 5. метрика носит экспериментальный характер; 6. Резервирование данных – один из наиболее эффективных способов предотвращения фактов повреждения данных, однако, резервирование данных относится к метрике «надёжность».
2. внутренние метрики безопасности: 2.1 протоколирование доступа;	X = A / B; A = число типов доступа, которые были зарегистрированы корректно, как определено в спецификации;	

	B = число типов доступа, которые должны регистрироваться по спецификации;	
2.2 контроль доступа;	X = A / B; A = число требований контроля доступа, реализованных корректно, в соответствии со спецификацией; B = число требований контроля доступа в спецификации;	
2.4 предотвращение повреждения данных;	X = A / B; A = число реализованных механизмов защиты от повреждения данных; B = число механизмов, требуемых по спецификации;	необходимо учитывать уровни безопасности при использовании этой метрики;
2.5 криптографическая защита данных;	X = A / B; A = число реализованных механизмов; B = число требуемых механизмов по спецификации;	криптографическая защита данных может касаться, например, данных в открытых базах данных или общедоступных данных.
3. метрики безопасности качества в использовании: 3.1 безопасность пользователей и их здоровья;	X = 1 – A / B; A = число пользователей, сообщивших о наличии проблем; B = число пользователей;	Проблемы со здоровьем могут включать: травмы от многократно повторяющихся мышечных напряжений, утомление, головная боль и т.д.;
3.2 безопасность людей, задействованных в использовании системы;	X = 1 – A / B; A = число людей, подверженных риску; B = число людей, задействованных в использовании продукта;	
3.3 экономический ущерб;	X = 1 – A / B; A = число событий экономического ущерба; B = общее число использования системы;	также можно учитывать ситуации, где был риск экономического ущерба;
3.4 повреждение прочего ПО;	X = 1 – A / B; A = число событий повреждения прочего ПО; B = общее число использования системы;	также можно учитывать ситуации, где был риск повреждения прочего ПО; метрика также может быть вычислена как X = суммарная стоимость повреждённого ПО / время использования.

Методология оценки характеристик

Методологии оценивания характеристик качества готовых ПП на различных этапах жизненного цикла посвящен международный стандарт ISO / IEC 14598-1-6:1998-2001 «Software engineering — Product evaluation» (Оценивание программного продукта), состоящий из шести частей:

Часть 1. 1999. Общий обзор.

Часть 2. 2000. Планирование и управление.

Часть 3. 2000. Процесс для разработчиков.

Часть 4. 1999. Процесс для приобретателей.

Часть 5. 1998. Процесс для оценщиков (испытателей).

Часть 6. 2001. Документирование оценки модулей.

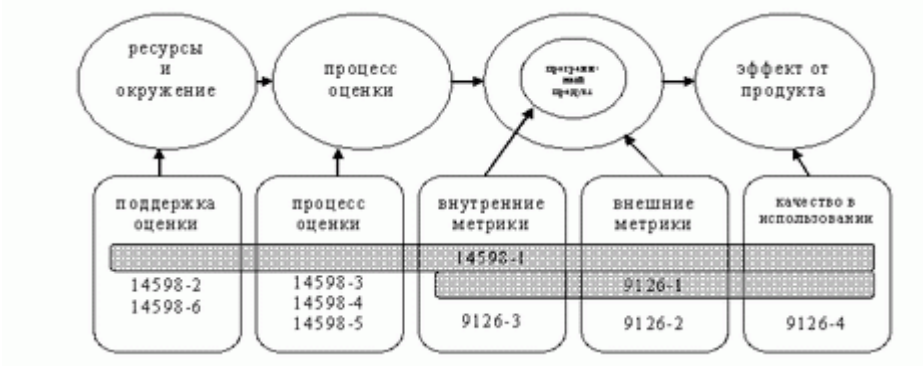

Рисунок 1 — Взаимосвязь стандартов ISO/IEC 9126 и 14598

Методология оценки характеристик безопасности ПП в соответствии со стандартом ISO/ IEC 14598 в общем виде будет представлять следующее:

- разработка исходных требований для проведения оценки (определение целей испытаний; выбор характеристик, субхарактеристик, выбор метрик, определение их требуемых значений);
- определение методики оценивания характеристик качества ПС, установление уровней приоритета метрик, выделение критериев для проведения измерений;
- планирование и проектирование процесса оценки характеристик качества в жизненном цикле ПС;
- выполнение измерений для оценивания; сравнение результатов с критериями и требованиями;
- обобщение и оценка результатов.

Для каждой характеристики качества рекомендуется сформировать шкалу измерений с выделением требуемых, допустимых и неудовлетворительных значений.

Заключение

Ниже перечислены основные преимущества и недостатки рассматриваемого подхода.

Преимущества:

- подход обращает внимание участников разработки ПО на необходимость учёта требований реализации безопасного ПО;
- возможность применения для оценки качества как для разработанного ПО, так и для ПО, находящегося в процессе разработки;
- подход представляет базу для разработки собственной методики оценки качества характеристик безопасности разрабатываемого ПО, и использования её для взаимодействия с заказчиками и оценщиками;
- возможность планировать и контролировать значения метрик безопасности в случае реализации ПО в ограниченные сроки или при ограниченном бюджете проекта;
- ряд метрик в стандартах необходимо вычислять экспериментальным путём (использование «тестов на проникновение»), что обычно не практикуется в случае обычного функционального тестирования ПО, где происходит тестирование функций в соответствии со спецификацией;
- интеграция со всем процессом разработки ПО в соответствии с их жизненным циклом (ISO/ IEC 12207), а также родственными стандартами ISO 9000-9001, которые уже сейчас активно применяются в странах СНГ;

Недостатки:

- не учтена специфика разрабатываемого ПО (угрозы действующие на функции ПО, величины рисков этих угроз, величины возможных ущербов);
- стандартный набор метрик не может в полной мере характеризовать качество безопасности разрабатываемого ПО и требуется расширение набора;
- подход применяется для оценки качества безопасности ПО, но не даёт гарантий безопасности, однако может давать хороший материал для анализа, в случае проведения оценки защищённости в соответствии с ISO/ IEC 15408 («Общие критерии»). С другой стороны, в случае, если наряду с данным подходом применяется также и ISO/ IEC 15408 (имеется профиль защиты или задание на обеспечение безопасности ПО или системы, в состав которой входит данное ПО), то подход на базе «Общих критериев» может

предоставить возможность для более точного определения необходимых метрик безопасности и более гибкого контроля их значений.

Следует отметить, что предыдущие версии такого стандарта как ISO/ IEC 9126:1991 уже приняты в качестве национальных в ряде стран СНГ, так например, РБ — СТБ ИСО/МЭК 9126-2003 «Информационные технологии. Оценка программной продукции. Характеристики качества и руководства по их применению»; РФ — ГОСТ Р ИСО/МЭК 9126-93) — «Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению», что говорит о высокой заинтересованности национальных нормотворческих органов в поддержке подходов, рекомендуемых стандартами ISO/ IEC.

Инженеры качества ПО и специалисты в области информационной безопасности занимаются одной проблемой – разработкой и реализацией качественных информационных систем, поэтому им следует объединить свои усилия для достижения общей цели.