

Лекция 3. Качество программного обеспечения, метрический анализ качества, метрики ПО.

Как проверить, что требования определены достаточно полно и описывают все, что ожидается от будущей программной системы? Это можно сделать, проследив, все ли необходимые аспекты *качества ПО* отражены в них.

Именно понятие *качественного ПО* соответствует представлению о том, что *программа* достаточно успешно справляется со всеми возложенными на нее задачами и не приносит проблем ни конечным пользователям, ни их начальству, ни службе поддержки, ни специалистам *по* продажам. Да и самим разработчикам создание качественной программы приносит гораздо больше удовольствия.

Если попросить группу людей высказать свое мнение *по* поводу того, что такое *качественное ПО*, можно получить следующие варианты ответов:

- Его легко использовать.
- Оно демонстрирует хорошую *производительность*.
- В нем нет ошибок.
- Оно не портит пользовательские данные при сбоях.
- Его можно использовать на разных платформах.
- Оно может работать 24 часа в сутки и 7 дней в неделю.
- В него легко добавлять новые возможности.
- Оно удовлетворяет потребности пользователей.
- Оно хорошо документировано.

Все это действительно имеет непосредственное *отношение к качеству ПО*. Но эти ответы выделяют характеристики, важные для конкретного пользователя, разработчика или группы таких лиц. Для того чтобы удовлетворить потребности всех сторон (конечных пользователей, заказчиков, разработчиков, администраторов систем, в которых оно будет работать, регулирующих организаций и пр.), для достижения прочного положения разрабатываемого *ПО* на рынке и повышения потенциала его развития необходим учет всей совокупности характеристик *ПО*, важных для всех заинтересованных лиц.

Приведенные выше ответы показывают, что *качество ПО* может быть описано большим набором разнородных характеристик. Такой подход к описанию сложных понятий называется **холистическим** (от греческого слова, *целое*). Он не дает единой концептуальной основы для рассмотрения затрагиваемых вопросов, какую дает *целостная система* представлений (например, Ньтоновская механика в физике или классическая теория вычислимости на основе машин Тьюринга), но позволяет, *по крайней мере*, не упустить ничего существенного.

Качество программного обеспечения определяется в стандарте *ISO 9126* как вся совокупность его характеристик, относящихся к возможности удовлетворять высказанные или подразумеваемые потребности всех заинтересованных лиц.

Тот же стандарт *ISO 9126* [1,2,3,4] дает следующее *представление* качества.

Для всех этих аспектов качества введены метрики, позволяющие оценить их. Кроме того, для создания добротного *ПО* существенно качество технологических процессов его разработки. Взаимоотношения между этими аспектами качества *по* схеме, принятой *ISO 9126*, показано на рис. 1.1.



- **ISO 9000:2000 Quality management systems — Fundamentals and vocabulary.**

- **ISO 9001:2000 Quality management systems — Requirements. Models for quality assurance in design, development, production, installation, and servicing.**

Определяет общие правила обеспечения качества результатов во всех процессах жизненного цикла. (Аналог — ГОСТ Р-2001).

- Управление качеством.
- Управление ресурсами.
- Развитие системы управления.
- Исследования рынка.
- Проектирование продуктов.

- Приобретения.
- Производство.
- Оказание услуг.
- Защита продуктов.
- Оценка потребностей заказчиков.
- Поддержка коммуникаций с заказчиками.
- Поддержка внутренних коммуникаций.
- Управление документацией.
- Ведение записей о деятельности.
- Планирование.
- Обучение персонала.
- Внутренние аудиты.
- Оценки управления.
- Мониторинг и измерения.
- Управление несоответствиями.
- Постоянное совершенствование.
- Управление и развитие системы в целом.

Для каждого процесса требуется иметь планы развития процесса, состоящие как минимум из следующих разделов:

- Проектирование процесса.
- Документирование процесса.
- Реализация процесса.
- Поддержка процесса.
- Мониторинг процесса.
- Управление процессом.
- Усовершенствование процесса.

Помимо поддержки и развития системы процессов, нацеленных на удовлетворение нужд заказчиков и пользователей, *ISO 9001* требует:

Определить, документировать и развивать собственную систему качества на основе измеримых показателей.

Использовать эту систему качества как средство управления процессами, нацеливая их на большее удовлетворение нужд заказчиков, планируя и постоянно отслеживая качество результатов всех видов деятельности, в том числе и самого управления.

Обеспечить использование качественных ресурсов, качественного (компетентного, профессионального) персонала, качественной инфраструктуры и качественного окружения.

Постоянно контролировать соблюдение требований к качеству на практике, во всех процессах проектирования, производства, предоставления услуг и при приобретениях.

Предусмотреть процесс устранения дефектов, определить и контролировать качество результатов этого процесса.

Ранее использовавшиеся стандарты *ISO 9002:1994 Quality systems — Model for quality assurance in production, installation and servicing* и *ISO 9003:1994 Quality systems — Model for quality assurance in final inspection and test* в 2000 году были заменены соответствующими им частями *ISO 9001*.

- **ISO 9004:2000 Quality management systems — Guidelines for performance improvements.**

Системы управления качеством. Руководство по улучшению деятельности. (Аналог — ГОСТ Р-2001).

- **ISO/IEC 90003:2004 Software engineering — Guidelines for the application of ISO 9001:2000 to computer software.**

Этот стандарт конкретизирует положения *ISO 9001* для разработки программных систем, с упором на обеспечение качества при процессе проектирования. Он также определяет некоторый набор техник и процедур, которые рекомендуется применять для контроля и обеспечения качества разрабатываемых программ.

Стандарт *ISO 9126* [1,2,3,4] предлагает использовать для описания внутреннего и внешнего качества ПО многоуровневую модель. На верхнем уровне выделено 6 основных характеристик качества ПО. Каждая характеристика описывается при помощи нескольких входящих в нее **атрибутов**. Для каждого атрибута определяется набор метрик, позволяющих его оценить. Множество характеристик и атрибутов качества согласно *ISO 9126* показано на рис. 1.2.



Рис 1.2 Множество характеристик и атрибутов качества

Качество *ПО* - это относительное понятие, которое имеет смысл только при учете реальных условий его применения, поэтому требования, предъявляемые к качеству, ставятся в соответствии с условиями и конкретной областью их применения. Оно характеризуется тремя аспектами: *качество программного продукта*, качество процессов ЖЦ и качество сопровождения или внедрения (рис. 1.3).

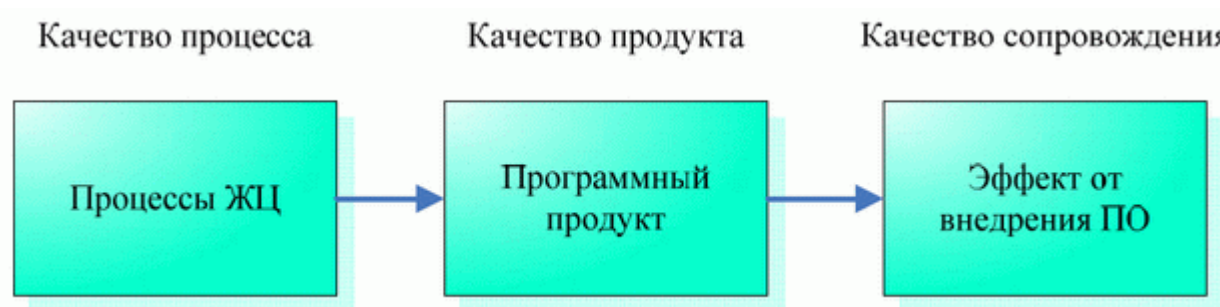


Рис. 1.3. Основные аспекты качества *ПО*

Аспект, связанный с процессами ЖЦ, определяет степень формализации, достоверности самих процессов ЖЦ разработки *ПО*, а также верификацию и валидацию промежуточных результатов на этих процессах. *Поиск* и устранение ошибок в готовом *ПО* проводится методами тестирования, которые снижают количество ошибок и повышают качество этого продукта.

Качество продукта достигается процедурами контроля промежуточных продуктов на процессах ЖЦ, проверкой их на достижение необходимого качества, а также методами сопровождения продукта. Эффект от внедрения ПС в значительной степени зависит от знаний обслуживающего персонала функций продукта и правил их выполнения. Модель качества *ПО* имеет следующие четыре уровня представления.

Первый уровень соответствует определению характеристик (показателей) качества *ПО*, каждая из которых отражает отдельную точку зрения пользователя на качество. Согласно стандарту *ISO 9126* в модель качества входит шесть характеристик или шесть показателей качества:

1. функциональность (functionality);
2. надежность (realibility);
3. удобство (usability);
4. эффективность (efficiency);
5. сопровождаемость (maitainnability);
6. переносимость (portability).

Согласно современному стандарту ГОСТ Р ИСО/МЭК 25010-2015 Информационные технологии (ИТ). Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE) Модели качества систем и программных продуктов, явившимся результатом пересмотра ИСО/МЭК 9126-1, у программных продуктов и преимущественно программных вычислительных систем много заинтересованных сторон, в число которых входят разработчики, приобретатели, пользователи или клиенты компаний, использующих преимущественно программные вычислительные системы. Подробная спецификация и оценка качества программного обеспечения и преимущественно программных вычислительных систем являются ключевыми факторами в обеспечении полезности для заинтересованных сторон.

Оценка может быть выполнена на основе определения необходимых и требуемых характеристик качества, связанных с задачами заинтересованных сторон и целями системы, включая характеристики качества, относящиеся к системе программного обеспечения и данным, а кроме того, и воздействие системы на ее заинтересованные стороны. Важно, чтобы, по возможности, характеристики качества были определены, измерены и оценены с использованием проверенных или широко распространенных показателей и методов измерения. Для идентификации соответствующих характеристик качества, которые могут далее использоваться для определения требований, критериев их удовлетворения и соответствующих показателей, могут быть использованы модели качества из настоящего документа.

Настоящий международный стандарт разработан на основе ИСО/МЭК 9126 "Программная инженерия - Качество продукта", который был разработан для удовлетворения вышеуказанных нужд и в котором были определены шесть характеристик качества и описана модель процесса оценки программного продукта.

Настоящий стандарт определяет:

а) модель качества при использовании, в состав которой входят пять характеристик, некоторые из которых, в свою очередь, подразделены на подхарактеристики. Эти характеристики касаются результата взаимодействия при использовании продукта в определенных условиях. Данная модель применима при использовании полных человеко-машинных систем, включая как вычислительные системы, так и программные продукты;

б) модель качества продукта, в состав которой входят восемь характеристик, которые, в свою очередь, подразделены на подхарактеристики. Характеристики относятся к статическим и динамическим свойствам программного обеспечения и вычислительных систем. Модель применима как к компьютерным системам, так и к программным продуктам.

Характеристики, определяемые обеими моделями, применимы к любым программным продуктам и компьютерным системам. Характеристики и подхарактеристики обеспечивают единую терминологию для определения спецификации, измерения и оценки качества систем и программного обеспечения. Модели предоставляют также множество характеристик качества, с которыми для полноты картины можно сравнить заявленные требования к качеству.

Второму уровню соответствуют атрибуты (ИСО/МЭК 9126-1) или подхарактеристики (ГОСТ Р ИСО/МЭК 25010-2015) для каждой характеристики качества, которые детализируют разные аспекты конкретной характеристики. Набор атрибутов (подхарактеристик) характеристик качества используется при оценке качества.

Третий уровень предназначен для измерения качества с помощью метрик, каждая из них согласно стандарту определяется как комбинация метода измерения атрибута (подхарактеристики) и шкалы измерения значений атрибутов. Для оценки атрибутов (подхарактеристик) качества на этапах ЖЦ (при просмотре документации, программ и результатов тестирования программ) используются метрики с заданным оценочным весом для нивелирования результатов метрического анализа совокупных атрибутов конкретного показателя и качества в целом. *Атрибут* (подхарактеристика) качества определяется с помощью одной или нескольких методик оценки на этапах ЖЦ и на завершающем этапе разработки ПО.

Четвертый уровень - это оценочный элемент метрики (*вес*), который используется для оценки количественного или качественного значения отдельного атрибута показателя *ПО*. В зависимости от назначения, особенностей и условий сопровождения *ПО* выбираются наиболее важные характеристики качества и их атрибуты.

Выбранные атрибуты (подхарактеристики) и их приоритеты отражаются в требованиях на разработку систем либо используется соответствующие приоритеты эталона класса *ПО*, к которому это *ПО* относится.

Метрический анализ качества.

Исследования метрического анализа качества показывают, что не существует единственной метрики, которая бы дала универсальный рейтинг качества программного обеспечения. Измерения качества дают спектр проектно-зависимых метрик, которые являются руководящей основой для принятия решений в процессе разработки, заказа и сопровождения программного обеспечения.

Следует отметить, что метрики качества являются изначально неочевидной категорией. Исторически сначала были выделены ряд универсальных и неполных метрик на основе следующих шагов:

1. Определение множества характеристик, которые, являясь важными для программного обеспечения, допускают несложное измерение и не перекрываются.
2. Выделение кандидатов в метрики, которые измеряют степень удовлетворения указанным характеристикам.
3. Исследование характеристик и связанных метрик, для определения корреляции, значимости, степени автоматизируемости.
4. Исследование корреляции между метриками, степени перекрытия, зависимости и недостатков.
5. Рафинирование множества метрик в целом во множество метрик, которые в совокупности адекватно отражают качество программного обеспечения.
6. Корректировка каждой метрики в итоговом множестве в контексте зафиксированных множеств характеристик и метрик.

Нижний слой характеристик в иерархии должен быть строго дифференцирован для того, чтобы исключить (или минимизировать) перекрытия. Данный слой должен состоять из примитивных характеристик, допускающих измерение.

Измерение характеристик нижнего слоя может происходить путем ручного сбора информации, специальными автоматизированными средствами, возможен экспертный способ. Каждая из собранных метрик будет иметь собственные характеристики. Область применения метрик может локализоваться внутри проекта, внутри платформы разработки или быть универсальной. Степень влияния метрик на итоговое качество также является различным. Указанные свойства метрик должны быть документированы и доступны при их практическом использовании. Для мониторинга метрик качества и подготовки информации для принятия решений собранные метрики должны представляться в наглядном виде, обеспечивающим полноту информации, что особенно важно при отсутствии консолидированных метрик качества.

Стандарт ISO/IEC 9126-2 определяет следующие типы мер:

- мера размера ПО в разных единицах измерения (число функций, строк в программе, размер дисковой памяти и др.);
- мера времени (функционирования системы, выполнения компонента и др.);
- мера усилий (производительность труда, трудоемкость и др.);
- мера учета (количество ошибок, число отказов, ответов системы и др.).

Специальной мерой может служить уровень использования повторных компонентов и измеряется как отношение размера продукта, изготовленного из готовых компонентов, к размеру системы в целом. Данная мера используется также при определении стоимости и качества ПО. Примеры метрик:

- общее число объектов и число повторно используемых;
- общее число операций, повторно используемых и новых операций;
- число классов, наследующих специфические операции;
- число классов, от которых зависит данный класс;
- число пользователей класса или операций и др.

При оценке общего количества некоторых величин часто используются среднестатистические метрики (среднее число операций в классе, наследников класса или операций класса и др.).

Как правило, меры в значительной степени являются субъективными и зависят от знаний экспертов, производящих количественные оценки атрибутов компонентов программного продукта.

Стандартная оценка значений показателей качества

Оценка качества ПО согласно четырехуровневой модели качества начинается с нижнего уровня иерархии, т.е. с самого элементарного свойства оцениваемого атрибута показателя качества согласно установленных мер. На этапе проектирования устанавливают значения оценочных элементов для каждого атрибута показателя анализируемого ПО, включенного в требования.

Примеры внутренних показателей даны в ИСО/МЭК ТО 9126-3 (подлежат замене ИСО/МЭК 25022). Характеристики и подхарактеристики могут быть измерены извне в той степени, в которой эта возможность обеспечивается системой, содержащей программное обеспечение. Примеры внешних показателей даны в ИСО/МЭК ТО 9126-2 (подлежат замене ИСО/МЭК 25023).

По определению стандарта ISO/IEC 9126-2 метрика качества ПО представляет собой "модель измерения атрибута, связываемого с показателем его качества". При измерении показателей качества данный стандарт позволяет определять следующие типы мер:

меры размера в разных единицах измерения (количество функций, размер программы, объем ресурсов и др.);

меры времени - периоды реального, процессорного или календарного времени (время функционирования системы, время выполнения компонента, время использования и др.);

меры усилий - продуктивное время, затраченное на реализацию проекта (производительность труда отдельных участников проекта, коллективная трудоемкость и др.);

меры интервалов между событиями, например, время между последовательными отказами;

счетные меры - счетчики для определения количества обнаруженных ошибок, структурной сложности программы, числа несовместимых элементов, числа изменений (например, число обнаруженных отказов и др.).

Метрики качества используются при оценке степени тестируемости с помощью данных (безотказная работа, выполнимость функций, удобство применения интерфейсов пользователей, БД и т.п.) после проведения испытаний ПО на множестве тестов.

Наработка на отказ как атрибут надежности определяет среднее время между появлением угроз, нарушающих безопасность, и обеспечивает трудноизмеримую оценку ущерба, которая наносится соответствующими угрозами. Очень часто оценка программы проводится по числу строк.

При сопоставлении двух программ, реализующих одну прикладную задачу, предпочтение отдается короткой программе, так как её создает более квалифицированный персонал и в ней меньше скрытых ошибок и легче модифицировать. По стоимости она дороже, хотя времени на отладку и модификацию уходит больше. Т.е. длину программы можно использовать в качестве вспомогательного свойства для сравнения программ с учетом одинаковой квалификации разработчиков, единого стиля разработки и общей среды.

Если в требованиях к ПО было указано получить несколько показателей, то просчитанный после сбора данных показатель умножается на соответствующий весовой коэффициент, а затем суммируются все показатели для получения комплексной оценки уровня качества ПО.

На основе измерения количественных характеристик и проведения экспертизы качественных показателей с применением весовых коэффициентов, нивелирующих разные показатели, вычисляется итоговая оценка качества продукта путем суммирования результатов по отдельным показателям и сравнения их с эталонными показателями ПО (стоимость, время, ресурсы и др.).

Т.е. при проведении оценки отдельного показателя с помощью оценочных элементов просчитывается весомый коэффициент -метрика, -показатель, -атрибут. Например, в качестве -показателя возьмем переносимость. Этот показатель будет вычисляться по пяти атрибутам, причем каждый из них будет умножаться на соответствующий коэффициент.

Все метрики - атрибута суммируются и образуют показатель качества. Когда все атрибуты оценены по каждому из показателей качества, производится суммарная оценка отдельного показателя, а потом и интегральная оценка качества с учетом весовых коэффициентов всех показателей ПО.

В конечном итоге результат оценки качества является критерием эффективности и целесообразности применения методов проектирования, инструментальных средств и методик оценивания результатов создания программного продукта на стадиях ЖЦ.

Метрики качества программного обеспечения

В настоящее время в программной инженерии еще не сформировалась окончательно система метрик. Действуют разные подходы к определению их набора и методов измерения.

Система измерения включает метрики и модели измерений, которые используются для количественной оценки качества ПО.

При определении требований к ПО задаются соответствующие им внешние характеристики и их атрибуты (подхарактеристики), определяющие разные стороны управления продуктом в заданной среде. Для набора характеристик качества ПО, приведенных в требованиях, определяются соответствующие метрики, модели их оценки и диапазон значений мер для измерения отдельных атрибутов качества.

Согласно стандарту метрики определяются по модели измерения атрибутов ПО на всех этапах ЖЦ (промежуточная, внутренняя метрика) и особенно на этапе тестирования или функционирования (внешние метрики) продукта.

Остановимся на классификации метрик ПО, правилах для проведения метрического анализа и процесса их измерения.

Типы метрик.

Существует три типа метрик:

- метрики программного продукта, которые используются при измерении его характеристик - свойств;
- метрики процесса, которые используются при измерении свойства процесса ЖЦ создания продукта.
- метрики использования.

Метрики программного продукта включают:

- внешние метрики, обозначающие свойства продукта, видимые пользователю;
- внутренние метрики, обозначающие свойства, видимые только команде разработчиков.

Внешние метрики продукта - это метрики:

- надежности продукта, которые служат для определения числа дефектов;
- функциональности, с помощью которых устанавливаются наличие и правильность реализации функций в продукте;
- сопровождения, с помощью которых измеряются ресурсы продукта (скорость, память, среда); применимости продукта, которые способствуют определению степени доступности для изучения и использования;
- стоимости, которыми определяется стоимость созданного продукта.

Внутренние метрики продукта включают:

- метрики размера, необходимые для измерения продукта с помощью его внутренних характеристик;
- метрики сложности, необходимые для определения сложности продукта;
- метрики стиля, которые служат для определения подходов и технологий создания отдельных компонентов продукта и его документов.

Внутренние метрики позволяют определить производительность продукта и являются релевантными по отношению к внешним метрикам.

Внешние и внутренние метрики задаются на этапе формирования требований к ПО и являются предметом планирования и управления достижением качества конечного программного продукта.

Метрики продукта часто описываются комплексом моделей для установки различных свойств, значений модели качества или прогнозирования. Измерения проводятся, как правило, после калибровки метрик на ранних этапах проекта. Общая мера - степень трассируемости, которая определяется числом трасс, прослеживаемых с помощью моделей сценариев типа UML и оценкой количества:

- требований;
- сценариев и действующих лиц;
- объектов, включенных в сценарий, и локализация требований к каждому сценарию;
- параметров и операций объекта и др.

Примером широко используемых внешних метрик программ являются метрики Холстеда - это характеристики программ, выявляемые на основе статической структуры программы на конкретном языке программирования: число вхождений наиболее часто встречающихся операндов и операторов; длина описания программы как сумма числа вхождений всех операндов и операторов и др.

На основе этих атрибутов можно вычислить время программирования, уровень программы (структурированность и качество) и языка программирования (абстракции средств языка и ориентация на проблему) и др.

В качестве **метрик процесса** могут быть время разработки, число ошибок, найденных на этапе тестирования и др. Практически используются следующие метрики процесса:

- общее время разработки и отдельно время для каждой стадии;
- время модификации моделей;
- время выполнения работ на процессе;
- число найденных ошибок при инспектировании;
- стоимость проверки качества;
- стоимость процесса разработки.

Метрики использования служат для измерения степени удовлетворения потребностей пользователя при решении его задач. Они помогают оценить не свойства самой программы, а результаты ее эксплуатации - эксплуатационное качество.

Примером может служить - точность и полнота реализации задач пользователя, а также затраченные ресурсы (трудозатраты, производительность и др.) на эффективное решение задач пользователя. *Оценка требований* пользователя проводится с помощью внешних метрик.