

## ПОСТРОЕНИЕ МОДЕЛИ КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

---

**Abstract:** In the paper the generalized recommendations on construction of model of quality of software in accordance with recommendations of international standard ISO/IEC 9126-(1:4) are discussed. Short recommendations on forming requirements for development information system and control of quality of the system on different design times are also given.

**Key words:** ISO, ISO standards, software, life cycle, quality model, quality metrics, quality monitoring.

**Аноація:** У статті розглянуто узагальнені рекомендації щодо побудови моделі якості програмного забезпечення у відповідності до рекомендацій міжнародного стандарту ISO/IEC 9126-(1:4). Також надано короткі рекомендації щодо формування вимог до розроблюваної інформаційної системи та контролю якості системи на різних етапах розробки.

**Ключові слова:** ISO, стандарти ISO, програмне забезпечення, життєвий цикл, модель якості, показники якості, контроль якості.

**Аннотація:** В статье рассмотрены обобщенные рекомендации по построению модели качества программного обеспечения в соответствии с рекомендациями международного стандарта ISO/IEC 9126-(1:4). Также даны краткие рекомендации по формированию требований к разрабатываемой информационной системе и контролю качества системы на разных этапах разработки.

**Ключевые слова:** ISO, стандарты ISO, программное обеспечение, жизненный цикл, модель качества, показатели качества, контроль качества.

### 1. Введение

По мере увеличения сложности и расширения области применения современных информационных систем (ИС) при одновременном росте ответственности за выполняемые функции резко повысились требования к качеству и безопасности применения программных комплексов. В некоторых сферах человеческой деятельности применение некачественного программного обеспечения (ПО) может нанести значительный и непоправимый ущерб. В связи с этим одним из важных моментов в жизненном цикле ИС стало обеспечение требуемого качества программ и данных.

Очевидно также, что качественный продукт невозможно создать, если на предприятии или в его подразделении не внедрена система качества. Даже в неполном объеме внедренная система качества позволяет более эффективно управлять процессом разработки и сопровождения ПО, что значительно повышает потребительские качества программных продуктов. В общих чертах система качества это, во-первых, пакет разработанных и утвержденных нормативно-методических документов (инструкций, методик, директив и т.д.) на каждый выделенный технологический процесс проектирования, изготовления и распространения изделия; во-вторых, контроль поэтапного выполнения технологических процессов в соответствии с утвержденным планом.

### 2. Комплексное решение задачи обеспечения качества программных систем

Для обеспечения качества информационной системы нужно, как минимум, выполнить четыре условия. Во-первых, разработать пакет нормативно-методических документов в соответствии с требованиями и рекомендациями международных, национальных и отраслевых стандартов. Во-вторых, построить базовую модель качества информационной системы на базе международных стандартов серии ISO/IEC 9126, которые регламентируют характеристики и метрики качества программного обеспечения. В-третьих, в требованиях на программное обеспечение четко

задекларировать сведения о понятиях и требуемых значениях характеристик качества ПО. В-четвертых, реализовать систему контроля качества на всех или самых важных этапах разработки программного обеспечения.

Комплексное решение задачи обеспечения качества программных систем предполагает разработку и внедрение той или иной системы управления качеством. В мировой практике наибольшее распространение получили системы, основанные на международных стандартах серии ISO 9000, включающей, в том числе, стандарт ISO 9003:2004, который регламентирует обеспечение качества программных продуктов. Стандарты этой серии должны служить руководством для ведущих специалистов компаний, разрабатывающих программное обеспечение на заказ. Внедренная система управления качеством при разработке и производстве программного обеспечения позволяет создавать конкурентоспособную продукцию регионального и мирового уровня.

Построение модели качества в виде многоуровневой структуры показателей качества позволяет системно описать требования к программному обеспечению, определяя, какие его свойства по каждой из шести характеристик верхнего уровня хотят видеть заинтересованные стороны. В большинстве случаев при выборе показателей качества приоритетными свойствами соответствующего программного обеспечения являются назначение и функциональная пригодность. Достаточно полное и корректное описание требований к этим свойствам служит базой для определения требований к остальным характеристикам ПО. Процесс определения требований к качеству программного обеспечения состоит из двух этапов:

- Обоснование и выбор перечня показателей качества ПО.
- Выбор и утверждение перечня метрик, оценочных элементов, шкал измерения показателей качества.

Процедура контроля качества позволяет убедиться, что определенные характеристики качества программного обеспечения достигнуты. Для оценки многих атрибутов качества не существует более эффективных способов, чем тестирование. Организация тестирования ПО регламентируется международными стандартами ISO/IEC и IEEE.

### **3. Построение модели качества**

Что же в реальности представляет собой модель качества информационной системы и какие характеристики желательно задекларировать в требованиях на систему? Как было сказано выше, модель качества информационной системы желательно строить на базе международных стандартов, которые регламентируют показатели качества программного обеспечения. Набор показателей качества в стандартах жестко не закреплен, поэтому на этапе построения модели качества конкретной проектируемой системы необходимо сформировать, исходя из ее функционального назначения, полный перечень требуемых характеристик и свойств. При этом необходимо пользоваться последними редакциями международных и региональных стандартов, которые постоянно развиваются и модифицируются.

В качестве примера рассмотрим построение обобщенной модели качества информационной системы в виде 4-уровневой иерархической структуры показателей качества. Уровнями модели будут выступать:

- Факторы – первый уровень.
- Характеристики – второй уровень.
- Метрики – третий уровень.
- Оценочные элементы – четвертый уровень.

В простейшем варианте модели качества выделим таких три основных фактора, которые принадлежат самому верхнему уровню в иерархии структуры показателей качества:

- Программа и данные.
- Документация пользователя.
- Описание продукта.

Рассмотрим формирование требований собственно к программам и данным. По стандарту ISO/IEC 9126 необходимо определить свойства программного обеспечения по каждой из шести характеристик, а именно:

- Функциональность.
- Надежность.
- Используемость (удобство использования).
- Эффективность.
- Сопровождаемость.
- Мобильность.

Для каждой из шести характеристик необходимо обосновать и выбрать полный перечень метрик для построения модели качества системы. Например, характеристика “Функциональность” может быть описана такими метриками:

- Инсталляция.
- Наличие функций.
- Корректность.
- Совместимость внутренняя.
- Совместимость внешняя.

Для каждой из метрик необходимо обосновать и выбрать полный перечень оценочных элементов. Например, метрика “Инсталляция” может быть описана следующими оценочными элементами:

- Наличие возможности выполнить инсталляцию продукта пользователем.
- Достаточность информации в инструкции по инсталляции для успешной установки продукта.
- Наличие возможности автозапуска процесса инсталляции (autorun).
- Наличие возможности альтернативного запуска процесса инсталляции в случае, когда не срабатывает автозапуск.
- Наличие возможности выборочной инсталляции, если продукт состоит из автономных компонентов.

- Наличие возможности деинсталляции продукта.
- Наличие возможности выборочной деинсталляции компонентов продукта.
- Наличие возможности повторной инсталляции продукта, если инсталляция была прервана или не завершена по разным причинам.

• Наличие возможности проверить дееспособность продукта после инсталляции с помощью соответствующих тестов, если это необходимо.

Метрика “Наличие функций” может быть описана следующими оценочными элементами:

- Наличие в полном объеме всех задекларированных функций.
- Наличие в полном объеме всех задекларированных функциональных связей.
- Соблюдение технологии обработки данных (доступность на данный момент только тех функций, которые отвечают технологической последовательности обработки данных).
- Наличие функции тестирования, если она была задекларирована.
- Наличие тестов и описания к ним, если функция тестирования была задекларирована.
- Возможность функционирования в многотерминальном режиме.
- Возможность функционирования в среде локальной сети.
- Возможность функционирования в среде Internet.
- Возможность функционирования в среде Intranet.
- Все функции, указанные в документации пользователя, должны выполняться по форме, приведенной в документации, с соответствующими средствами, свойствами и данными.
- Все функции, указанные в документации пользователя, должны выполняться в пределах приведенных там граничных значений.

Рассмотрим формирование требований собственно к документации на программное изделие. Формирование требований к документации пользователя определяется следующими пятью характеристиками:

- Полнота.
- Корректность.
- Совместимость.
- Понятность.
- Легкость восприятия.

Например, характеристика “Полнота” может быть описана такими метриками:

- Полнота документации.
- Инструкция по инсталляции.
- Инструкция по обслуживанию.

Например, метрика “Полнота документации” может быть описана следующими оценочными элементами:

- Наличие полного перечня документации.
- Наличие короткой аннотации.
- Наличие описания класса решаемых задач.
- Наличие описания структуры функций программного обеспечения.
- Наличие подробного описания всех функций, заявленных в продукте.

- Наличие подробного описания всех функций, которые вызывает пользователь в программе.

- Наличие описания алгоритмов.
- Наличие описания методов решения задач.
- Наличие описания репозитория (библиотеки базовых объектов).
- Наличие описания интерфейсов пользователя.
- Наличие описания входных и выходных данных.
- Наличие описания служебных данных.
- Наличие описания диагностических сообщений.
- Наличие описания всех граничных значений, которые были указаны в описании продукта.
- Наличие описания программной среды функционирования ПО.
- Наличие описания аппаратной среды функционирования ПО.
- Наличие информации о технологии переноса для мобильных программ.
- Наличие описания методологии работы с программой.

Описывая в таком ключе детально и последовательно сверху вниз показатели качества многоуровневой структуры показателей, строится модель качества проектируемой информационной системы. С приведенного примера видно, что оценочный элемент является тем критерием в модели качества, который непосредственно определяет наличие той или иной функции или свойства в программном изделии. Построение модели качества идет сверху вниз от характеристик до оценочных элементов, а оценка достигнутых показателей качества идет в обратном направлении: от оценки единичных оценочных элементов до агрегированной оценки вышестоящих показателей качества в модели качества информационной системы.

В идеале компания, которая разрабатывает программное обеспечение на заказ, может спроектировать обобщенную базовую модель качества абстрактной информационной системы с максимально возможным перечнем показателей качества и в каждом конкретном случае на основе базовой модели строить модель качества требуемой информационной системы, расширяя или сужая диапазон базовых показателей качества.

#### **4. Формирование требований к разрабатываемой системе**

Далеко не последнюю роль в обеспечении качества программных комплексов играет процедура формирования требований к разрабатываемой системе. Наиболее дорогие ошибки совершаются на первых фазах жизненного цикла – это ошибки в определении требований, выборе архитектуры, высокоуровневом проектировании. Необходимо четко определять требования к программному обеспечению, но не “требования вообще”, а требования, которые можно проверить. Для удобства последующего контроля и тестирования требования желательно сгруппировать по видам, например, как функциональные, нефункциональные требования к качеству, требования к дизайну. Также в требованиях должна быть заложена возможность в последующих испытаниях сравнить текущее функционирование системы с ее ожидаемым поведением. Для корректного формирования требований к проектируемой системе на этапе принятия решения о разработке определяется вид

будущего программного изделия. С точки зрения потребительских свойств могут быть два вида программных изделий:

- Программное изделие, которое разработано независимым разработчиком с целью удовлетворения потребностей рынка. Такое программное обеспечение называется программным продуктом. Оно тиражируется в промышленных условиях и с целью насыщения рынка поставляется в розничную или дистрибьюторскую торговую сеть.

- Программное изделие, которое разработано для собственных нужд или по заказу конкретного потребителя с целью удовлетворения потребностей заказчика изделия.

Программное изделие, разработанное по заказу конкретного потребителя, при выполнении определенных условий может быть преобразовано в программный продукт. Программный продукт – это пакет программ, который состоит из одного или нескольких программных изделий. Программный продукт полностью ориентирован на потребителя и должен отвечать определенным требованиям, которые предъявляются к товарам, услугам, продукции на потребительском рынке. Перечень основных требований к потребительским свойствам программного продукта следующий:

- **Отчужденность программного продукта от разработчика.** Программный продукт может распространяться и поддерживаться третьим лицом посредством дистрибьюторской сети, размеры и необходимость создания которой определяются долей рынка, которую занимает программный продукт, и ареалом распространения в региональном или мировом масштабе.

- **Создание и поддержка сети дистрибьюторов.** Разработчик обеспечивает создание и информационно-сервисную поддержку разветвленной сети дистрибьюторов (информационная поддержка, выпуск новых версий, курсы, консультации, обучение, семинары и прочее), а дистрибьюторы на местах формируют сеть розничной торговли и сервисное обслуживание клиентов.

- **Коробочный формат поставки программного продукта.** Такой формат предполагает наличие документации на программный продукт в типографском или электронном виде и на электронных носителях с программным обеспечением.

- **Маркировка.** Электронные носители и документация должны быть маркированы торговой маркой или логотипом фирмы-производителя.

- **Электронные носители.** Электронные носители должны включать краткое описание продукта, конфигурацию поставки (перечень и назначение программных изделий, входящих в пакет поставки), условия инсталляции, условия эксплуатации и другую необходимую информацию. В электронном виде такая информация, как правило, представляется в текстовых ознакомительных файлах типа readme. Для гарантии читабельности ознакомительного файла информация в нем должна быть представлена, как минимум, на двух языках: английском и региональном или национальном.

- **Гарантийные обязательства – вершина системы качества.** Наличие в пакете поставки сервисных и гарантийных обязательств:

- Координаты разработчика или дистрибьютора.
- Гарантийные обязательства.

- Условия сопровождения и поддержки программного продукта (исправление ошибок, обновление и прочее).

- Наличие информационного сайта разработчика или дистрибьютора.
- Наличие возможности консультаций по “горячей телефонной линии” или посредством e-mail.

- **Инсталляция пакета.** Прозрачность процедуры инсталляции программного пакета:

- Наличие возможности автоматического (autorun) или альтернативного запуска процедуры инсталляции.

- Обеспечение простоты и прозрачности процедуры инсталляции.

- Наличие возможности выборочной инсталляции, если продукт состоит из автономных компонент, которые могут быть установлены по желанию.

- Наличие в процедуре инсталляции подсказки-описания назначения каждого компонента, выполняемых им функций, функциональных связей с другими компонентами.

- Наличие возможности полной деинсталляции продукта.

- Наличие возможности выборочной деинсталляции автономных компонент.

- Наличие в процедуре деинсталляции подсказки-предупреждения о возможных нарушениях функциональных связей между установленными и удаляемыми компонентами.

- Обеспечение возможности повторной инсталляции продукта, если первичная инсталляция была прервана или по разным причинам завершена некорректно.

- **Возможность запуска программного продукта в эксплуатацию потребителем.**

Документация содержит достаточно подробные пошаговые инструкции по инсталляции и настройке продукта для запуска программы в эксплуатацию пользователем с разным уровнем компьютерной грамотности.

Приведем общие рекомендации по формированию требований к проектируемой системе. Требования к программному обеспечению формируются на базе, как минимум, трех аспектов:

- Объект автоматизации.
- Функциональное назначение.
- Качество программного обеспечения.

Объект автоматизации определяет условия, в которых должно эксплуатироваться программное обеспечение, и, в некоторых случаях, определенные аспекты архитектуры аппаратно-программного комплекса. Для формирования требований необходимо определить:

- организационную структуру объекта автоматизации;
- информационную структуру (поток информации, направления движения, объемы);
- технологическую последовательность обработки информации;
- функциональную взаимосвязь данных.

Функциональное назначение программного обеспечения определяет архитектуру программно-аппаратного комплекса. Для проектирования и разработки программного обеспечения необходимо определить:

- перечень решаемых задач;
- набор функций, которые должны быть реализованы для решения каждой из задач;

- архитектуру аппаратного комплекса;
- архитектуру программного комплекса;
- архитектуру и дизайн интерфейса программного обеспечения;
- структуру и взаимосвязь данных;
- структуру и состав технической и пользовательской документации.

Нефункциональное назначение программного обеспечения определяет:

- требования к занимаемым ресурсам;
- ограничения на время решения задачи;
- скорость доступа к данным;
- объем обрабатываемых данных;
- безопасность и т.п.

На основе этих данных и с учетом требований к качеству программного обеспечения формируются требования к проектируемой информационной системе с различной степенью детализации. В системе ISO существуют наборы стандартов, которые могут быть полезны на разных этапах формирования требований к проектируемой информационной системе.

## **5. Контроль качества разрабатываемой системы**

Качество программного обеспечения определяет набор мероприятий, которые должны быть осуществлены для обеспечения качества информационной системы на всех этапах ее жизненного цикла. Контроль качества – это одна из мер в комплексе мер по обеспечению качества программного изделия, одним из важнейших этапов которой является тестирование программного обеспечения.

Тестирование может проводиться на разных фазах жизненного цикла и может быть как ручное, так и автоматизированное. Бесспорно, гораздо эффективнее поэтапно проводить контроль в процессе проектирования и разработки системы. Тестирование может проводиться как на соответствие заранее определенным требованиям в требованиях на систему, так и на соответствие требованиям, задекларированным в стандартах и нормативных документах разных уровней. Разные виды тестирования (модульное, функциональное, нагрузочное и т.д.) могут проводиться на разных этапах разработки ПО и на разных объектах (система в целом, отдельные подсистемы, компоненты, модули). Критерии тестирования и методы их оценки также зависят от вида и объекта тестирования и стадии разработки.

Для успешного проведения тестирования чрезвычайно важно тщательно спланировать эти работы. Необходим системный подход к процедуре тестирования, его организации и проведению. Рекомендуется разрабатывать пакет документов, регламентирующих проведение тестирования в соответствии с требованиями международных стандартов, таких как IEEE 829-1983, IEEE 829-2008 и других.

## **6. Выводы**

За последние несколько лет создано множество международных стандартов, регламентирующих процессы и продукты жизненного цикла программных средств и баз данных. Применение этих



стандартов может служить основой для систем обеспечения качества программных средств, однако требуются корректировка, адаптация или исключение некоторых положений стандартов применительно к принципиальным особенностям технологий и характеристик этого вида продукции.

#### **СПИСОК ЛИТЕРАТУРЫ**

1. ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model.
2. ISO/IEC 9126-2:2003 Software engineering – Product quality – Part 2: External metrics.
3. ISO/IEC 9126-3:2003 Software engineering – Product quality – Part 3: Internal metrics.
4. ISO/IEC 9126-4:2004 Software engineering – Product quality – Part 4: Quality in use metrics.
5. ISO/IEC 90003:2004 Software engineering – Guidelines for the application of ISO 9001:2000 to computer software.
6. IEEE 829-1998 Standard for Software Test Documentation.
7. IEEE 829-2008 Standard for Software and System Test Documentation.

*Стаття надійшла до редакції 14.11.2008*