Министерство науки и образования Украины

Харьковский национальный университет им. В.Н.Каразина

Факультет математики и информатики

Реферат на тему:

«Модели качества программного продукта SQuaRE»

Подготовили:

Макаренко Евгения,

Кузяева Татьяна,

Фомина Инна

Группа МФ-51

Харьков 2017

# Оглавление

[Оглавление 2](#_Toc483322246)

[Введение 3](#_Toc483322247)

[Идентификация, анализ и детализация объекта изучения 5](#_Toc483322248)

[Раскрытие спецификации (детализации структуры) анализируемых процессов в соответствии с ISO/IEC/IEEE 12207:2008 7](#_Toc483322249)

[Группы процессов жизненного цикла включают в себя: 7](#_Toc483322250)

[1. Процесс «Менеджмент качества в системном контексте» (6.2.5) 7](#_Toc483322251)

[2. Процесс «Обеспечение (гарантирование) качества ПО» (7.2.3) 9](#_Toc483322252)

[3. Процесс «Верификация и Валидация ПО» (7.2.4, 7.2.5) 9](#_Toc483322253)

[4. Процесс «Квалификационное тестирование ПО в системном и программном контекстах» (6.4.6, 7.1.7) 11](#_Toc483322254)

[5. Процесс «Измерения» (6.3.7). 12](#_Toc483322255)

[Модели качества ПО 14](#_Toc483322256)

[Характеристика показателей качества 16](#_Toc483322257)

[Серия стандартов ISO/IEC 25000. Структура 22](#_Toc483322258)

[ISO/IEC 25010. (SQuaRE) Модели: качества в использовании, качества программного продукта, качества данных 25](#_Toc483322259)

[Методы измерения. Метрики. 27](#_Toc483322260)

[«Гарантоспособность» критического ПО 29](#_Toc483322261)

[Аппарат радиальных метрических диаграмм (РМД) 30](#_Toc483322262)

[Заключение 32](#_Toc483322263)

[Список литературы: 33](#_Toc483322264)

# Введение

Программное обеспечение (ПО) — все или часть программ, процедур, правил и соответствующей документации системы обработки информации.

Жизненный цикл создания и использования компьютерных программ отражает различные их состояния, начиная с момента возникновения необходимости в данном программном изделии и заканчивая моментом его полного выхода из употребления у всех пользователей. Традиционно выделяют следующие основные этапы жизненного цикла программного обеспечения:

* анализ требований,
* проектирование,
* кодирование (программирование),
* тестирование и отладка,
* эксплуатация и сопровождение.

Основной нормативный документ, регламентирующий ЖЦ ПО – международный стандарт ISO/IEC 12207. Он определяет структуру ЖЦ, содержащую процессы, действия и задачи, выполняемые во время создания ПО.

Согласно этому стандарту, структура ЖЦ ПО базируется на трёх группах процессов:

1. основные процессы ЖЦ ПО (приобретение, поставка, разработка, эксплуатация, сопровождение);
2. вспомогательные процессы (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, оценка, аудит, решение проблем);
3. организационные процессы (управление проектами, создание инфраструктуры проекта, определение, оценка и улучшение самого ЖЦ, обучение).

Разработка ПО – это анализ, проектирование и реализация. Она включает все работы по созданию ПО и его компонент в соответствии с заданными требованиями, в том числе оформление проектной и эксплуатационной документации, подготовку материалов, необходимых для оценки качества, ввода в эксплуатацию и сопровождения.

Управление проектом связано с вопросами планирования и организации работ, создания коллективов разработчиков; контроля за сроками и качеством выполняемых работ.

Обеспечение качества проекта связано с проблемами верификации, проверки и тестирования ПО.

Оценка качества (ГОСТ 28195-89) осуществляется на всех этапах жизненного цикла программных средств (ПС) при:

* планировании показателей качества ПС;
* контроле качества на отдельных этапах разработки (техническое задание, технический проект, рабочий проект);
* контроле качества в процессе производства ПС;
* проверке эффективности модификации ПС в процессе сопровождения.

Проверка позволяет оценить соответствие параметров разработки с заявленным требованиями.

Управление конфигурацией – один из вспомогательных процессов, поддерживающих основные процессы жизненного цикла ПО, прежде всего процессы разработки и сопровождения ПО ИС.

Управление конфигурацией позволяет организовать, систематически учитывать и контролировать внесение изменений в ПО на всех стадиях ЖЦ. Общие принципы и рекомендации конфигурационного учёта, планирования и управления конфигурациями ПО отражены в проекте стандарта ISO/IEC 12207.

Каждому этапу ЖЦ ПО соответствуют программные продукты, разработанные согласно международным стандартам.

В данной работе подробно рассмотрен программный продукт SQuaRE, который разработан для определения модели качества программного продукта и процессов ЖЦ ПО.

# Идентификация, анализ и детализация объекта изучения

Объект изучения: программный продукт SQuaRE (Software product Quality Requirements and Evaluation).

Software product Quality Requirements and Evaluation **- серия стандартов**, разрабатываемая организациями ISO и IEС, определяющая модели качества программного продукта и процессов ЖЦ ПО.

Процедура контроля качества предназначена для того, чтобы убедиться, что определённые характеристики качества ПО достигнуты. Для оценки многих атрибутов качества не существует более эффективных способов, чем тестирование. Тестирование – наиболее широко применяемый метод контроля качества. Тестировать можно соблюдение любых требований, соответствие которым выявляется во время работы ПО. Одним из стандартов, которые регламентируют организацию тестирования ПО, является ISO/IEC 25051:2006 Software engineering - Software product Quality Requirements and Evaluation (SQuaRE).

SQuaRE определяет требования качества к программным продуктам и к документации по тестированию. Регламентирует содержание документации по тестированию, которая должна включать план тестирования, описание используемых методов тестирования, описание наборов тестов и результатов тестирования. Этот стандарт в 2006 году пришел на смену ISO/IEC 12119:1994.

Модель качества SQuaRE образовывает совокупность стандартов ISO/IEC 9126 и стандартов ISO/IEC 1598. В отличии от каждого из указанных стандартов отдельно, модель SQuaRE определяет модель характеристик качество ПО, которая состоит из таких атрибутов качества как:

* внутренние атрибуты качества (требования к качеству кода и к внутренней архитектуре);
* внешние атрибуты качества (требования к функциональным возможностям);
* атрибуты «качества в использовании» (те атрибуты качества, которые устанавливаются не только к ПО, но и ко всей информационной системе).

Именно атрибуты «качества в использовании» характеризуют эффект для пользователя от использования ПО в разных контекстах использования. Соответственно, уместным является анализ модели SQuaRE, как основания обеспечения требований к качеству и оценки программных устройств. Это позволит уменьшить неопределенность при совместной работе организации касательно разработке, внедрения и сопровождения ПО, например, между заказчиками, разработчиками и пользователями. Уместность применения модели SQuaRE обусловливается тем, что используемые подходы могут быть также применены в работе с заграничными партнерами.

# Раскрытие спецификации (детализации структуры) анализируемых процессов в соответствии с ISO/IEC/IEEE 12207:2008

## Группы процессов жизненного цикла включают в себя:

* процессы соглашения;
* процессы организационного обеспечения проекта;
* процессы проекта;
* технические процессы;
* процессы реализации программных средств;
* процессы поддержки программных средств;
* процессы повторного применения программных средств.

Рассмотрим процессы в группах относительно данного программного продукта.

1. Процесс «Менеджмент качества в системном контексте» (6.2.5)

Процесс менеджмента качества программных средств относится к группе процесса организационного обеспечения проекта.

Менеджмент качества - это скоординированная и взаимосвязанная деятельность по управлению, выстроенная таким образом, чтобы обеспечить надежную и бесперебойную работу организации.

Управление организацией, применительно к качеству, означает, что вся деятельность подчиняется установленным целям по качеству, и для достижения этих целей в организации разработана система планов, есть необходимые ресурсы, выполняются действия по достижению поставленных целей.

Цель процесса управления качеством - мониторинг (систематический контроль) качества для гарантии того, что продукт будет удовлетворять потребителю и предполагает выполнение следующего:

* определение количественных свойств качества, основанных на выявленных и предусмотренных потребностях пользователей;
* управление реализацией поставленных целей для достижения качества.

Процесс управления качеством основывается на гарантии того, что:

* цели достижения требуемого качества установлены для всех рабочих продуктов в контрольных точках продукта;
* определена стратегия достижения качества, метрики, критерии, приемы, требования к процессу измерения и др.;
* определены и выполняются действия, связанные с предоставлением продуктам свойств качества;
* проводится контроль качества (верификация и валидация) и целей;
* выполняются процессы измерения и оценивания конечного продукта на достижение требуемого качества.

Менеджмент качества включает в себя четыре основных составляющих:

* контроль качества;
* обеспечение качества;
* планирование качества;
* улучшение качества.

Рассмотрим подробнее:

* Контроль качества - это деятельность по оценке соответствия объекта контроля установленным требованиям. Деятельность по оценке может включать в себя измерения, испытания, наблюдения, мониторинг, проверку, калибровку и пр. мероприятия, результатом которых является сравнение значений наблюдаемых характеристик с заданными.
* Обеспечение качества представляет собой систематическую (регулярную) деятельность, за счет которой можно выполнить установленные требования. Она включает в себя работы по производству, управлению, материальному обеспечению, техническому обслуживанию и пр.
* Планирование качества – это действия, предусматривающие определение необходимых характеристик объекта и установление их целевых значений. Менеджмент качества называет такие действия постановкой целей в области качества. Также, в планирование качества входит определение процессов и ресурсов, необходимых для достижения целей.
* Улучшение качества заключается в реализации действий, за счет которых можно повысить возможности организации выполнить требования, предъявляемые к объекту. Под понятием «объект» менеджмент качества рассматривает продукцию, процессы, систему управления и организацию в целом.
* Как таковой, менеджмент качества, представляет собой достаточно большой и объемный раздел прикладной науки, который содержит и философию менеджмента качества, и теорию, и практические методы.

Одной из наиболее известных и популярных систем, которую менеджмент качества включает в себя на сегодняшний день, является ISO 9000 – система менеджмента качества, построенная на основе международных [стандартов ISO серии 9000](http://www.kpms.ru/standart.htm). Это одна из наиболее популярных и формализованных систем. Она фокусируется на строгой регламентации деятельности, четком взаимодействии сотрудников и непрерывном улучшении как отдельных подсистем, так и организации в целом.

1. Процесс «Обеспечение (гарантирование) качества ПО» (7.2.3)

Процесс обеспечения гарантии качества программных средств относится к процессу поддержки ПС.

Обеспечение качества заключается в выполнении и проверки того, что объект разработки выполняет указанные требования к качеству.

Цели обеспечения качества могут быть внутренние и внешние. Внутренние цели - создание уверенности у руководителя проекта, что качество обеспечивается. Внешние цели - это создание уверенности у пользователя, что требуемое качество достигнуто и получено качественное программное обеспечение.

1. Процесс «Верификация и Валидация ПО» (7.2.4, 7.2.5)

**Верификация** – это процесс оценки системы или её компонентов с целью определения того, удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа. То есть, выполняются ли задачи, цели и сроки по разработке продукта.

Основные цели верификации:

— системные требования, предназначенные для программной реализации, были должным образом переработаны в требования верхнего уровня к ПО, которые удовлетворяют этим системным требованиям;

— требования верхнего уровня были переработаны в архитектуру ПО и требования нижнего уровня, которые удовлетворяют требованиям верхнего уровня; если разработано несколько уровней требований к ПО между требованиями верхнего уровня и требованиями нижнего уровня, то каждый последующий уровень требований разработан так, чтобы удовлетворять требованиям более высокого уровня;

— архитектура ПО и требования нижнего уровня должным образом преобразованы в исходный код, удовлетворяющий им;

— исполняемый объектный код удовлетворяет требованиям к ПО;

— инструментальные средства, используемые для выполнения указанных работ, являются технически корректными и полными для заданного уровня ПО.

В состав задач процесса входит последовательная проверка того, что в программной системе:

• общие требования к информационной системе, предназначенные для программной реализации, корректно переработаны в спецификацию требований высокого уровня к комплексу программ, удовлетворяющих исходным системным требованиям;

• требования высокого уровня правильно переработаны в архитектуру ПО и в спецификации требований к функциональным компонентам низкого уровня, которые удовлетворяют требованиям высокого уровня;

• спецификации требований к функциональным компонентам ПО, расположенным между компонентами высокого и низкого уровня, удовлетворяют требованиям более высокого уровня;

• архитектура ПО и требования к компонентам низкого уровня корректно переработаны в удовлетворяющие им исходные тексты программных и информационных модулей;

• исходные тексты программ и соответствующий им исполняемый код не содержат ошибок.

**Валидация** – это определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе.

Цель валидации простая: доказать, что объект валидации действительно приводит к ожидаемым результатам. Другими словами, валидация должна показать, что производитель полностью управляет процессом производства. Валидация является интегральной частью «обеспечения качества», показывая, что производитель понимает причины изменчивости процесса и, главным образом, понимает какие параметры необходимо контролировать для обеспечения стабильности процесса. Эффективная валидация основана на менеджменте риска и современном уровне науки.

Задачи валидации:

* разработать стратегии и критериев валидации для всех рабочих продуктов;
* действия по проведению валидации;
* демонстрация соответствия разработанных программных продуктов требованиям заказчика и правилам их использования;
* согласование с заказчиком полученных результатов валидации.

1. Процесс «Квалификационное тестирование ПО в системном и программном контекстах» (6.4.6, 7.1.7)

Процесс квалификационного тестирования системы относится к техническим процессам ЖЦ ПО.

Целью квалификационного тестирования ПО является демонстрация того, что ПО удовлетворяет своим спецификациям и готово к использованию в заданных условиях эксплуатации. Такое тестирование выполняется для каждого компонента программного продукта по всем разделам требований при широком варьировании тестов. При этом также проверяется полнота технической и пользовательской документации и ее адекватность самим компонентам ПО.

План квалификационного тестирования ПО содержит информацию для проведения квалификационного тестирования (испытаний) систем и подсистем ПО, описание тестовой среды, которая будет использована при тестировании, идентифицирует выполняемые тесты и указывает план-график выполнения тестирования.

Для каждой предполагаемого тестового ПО должны быть указаны:

— идентификация, перечень и используемые версии ПО, для которых будет выполнено тестирование на данной установке, их назначение;

— идентификация, перечень и используемые виды аппаратных средств, интерфейсного оборудования, устройств связи, дополнительных внешних устройств, генераторов тестовых сообщений, устройств синхронизации тестов и т. п.;

— права собственности и лицензирование;

— организации, принимающие участие в квалификационном тестировании, их роли и ответственность.

Кроме того, в данном документе должны быть представлены план-график тестирования и матрица трассирования тестов к требованиям к ПО.

Компоненты, которые следует тестировать перед испытаниями комплекса на соответствие требованиям должны включать:

* Идентификатор версии испытываемого программного продукта;
* Результаты исправления дефектов, обнаруженных в предыдущей версии, если список дефектов приводится в спецификации, необходимо указать ссылку на спецификации требований функции и характеристик;
* Описание внешней среды, используемой для применения программного продукта;
* Эксплуатационные документы для пользователей, такие как руководство пользователя, инструкции по инсталляции, особенности версии конкретного продукта.

1. Процесс «Измерения» (6.3.7).

Процесс измерения системы относится к процессам поддержки проекта.

Согласно стандарту ISO 9004:2000:

Измерять - определять или устанавливать пространственную величину или количество (чего-нибудь);

Цель: определять или устанавливать (пространственную величину или количество) путем приложения некоторого предмета с известным размером или объемом или путем сравнения с некоторой фиксированной единицей измерения.

Измерять можно двумя способами: в одних случаях количество определяют путем пересчета или сравнивая с некоей величиной, принятой за единицу измерения, в других - расчетным путем (например, вычисляя среднее значение, значение размаха, медианы, дисперсии и т.п.). Последние действия не являются, конечно же, измерением в чистом виде, но цели действий в обоих случаях одни и те же: определить численное значение интересуемой нас характеристики.

Согласно анализу требований раздела 8.2.4 ISO 9001:2000 требуется проводить мониторинг и измерение тех характеристик, которые используются «для верификации того, выполнены ли требования к продукции». Нет сомнения, что потребитель, а, вслед за ним и изготовитель, устанавливает характеристики и требования к ним для конечной продукции, включая и свои дополнительные требования. Некоторые из этих характеристик могут быть приобретены и на промежуточных этапах создания продукции, поэтому в стандарте сказано: «Это должно осуществляться на соответствующих стадиях процесса создания продукции в соответствии с запланированными мероприятиями».

Согласно ISO 9000:2015:

СМК – система менеджмента качества:

Система менеджмента качества СМК включает в себя деятельность, посредством которой организация устанавливает свои цели и определяет процессы и ресурсы, требуемые для достижения желаемых результатов.

* СМК управляет взаимодействующими процессами и ресурсами, требуемыми для создания ценности и производства продукции для соответствующих заинтересованных сторон.
* СМК позволяет высшему руководству оптимизировать использование ресурсов с учетом краткосрочных и долгосрочных последствий принимаемых решений.
* СМК обеспечивает средства определения действий для принятия мер в отношении ожидаемых и незапланированных ситуаций при поставке продуктов и услуг.

Процесс - совокупность взаимосвязанных или взаимодействующих видов деятельности, которая использует входы для производства запланированных результатов.

Контекст организации:

Понимание контекста организации – это процесс. Этот процесс определяет факторы, которые влияют на намерения организации, ее цели и устойчивость. Он учитывает внутренние факторы, такие как ценности, культуру, знания и производственные характеристики организации. Он также принимает во внимание внешние факторы, которые связаны с юридической, технологической, культурной, социальной и экономической областями, сферой конкуренции и рынка. Примерами того, как может быть выражено предназначение организации являются ее видение, миссия, политики и цели.

# Модели качества ПО

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. 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 в модель качества входит шесть характеристик или шесть показателей качества:

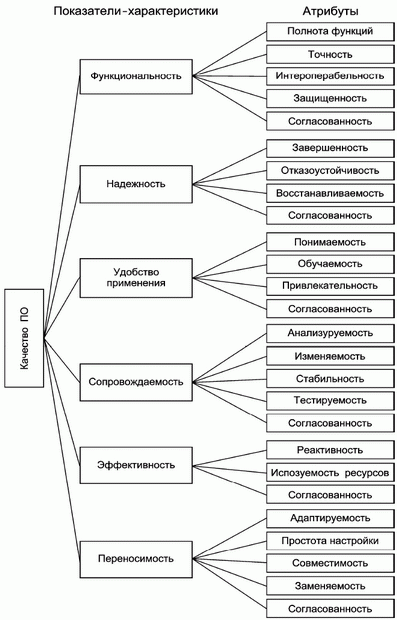
1. функциональность (functionality);
2. надежность (realibility);
3. удобство (usability);
4. эффективность (efficiency);
5. сопровождаемость (maitainnability);
6. переносимость (portability).

Второму уровню соответствуют атрибуты для каждой характеристики качества, которые детализируют разные аспекты конкретной характеристики. Набор атрибутов 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 и шкалы измерения значений атрибутов. Для оценки атрибутов 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 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 и их приоритеты отражаются в требованиях на разработку систем либо используется соответствующие приоритеты эталона класса ПО, к которому это ПО относится.

[](http://www.intuit.ru/EDI/03_01_17_9/1483395677-6027/tutorial/364/objects/10/files/10_02.gif)

## Характеристика показателей качества

1. Функциональность - совокупность свойств, определяющих способность ПО выполнять перечень функций в заданной среде и в соответствии с требованиями к обработке и общесистемным средствам. Под функцией понимается некоторая упорядоченная последовательность действий для удовлетворения потребительских свойств. Функции бывают целевые (основные) и вспомогательные.

К атрибутам функциональности относятся:

* + функциональная полнота - свойство компонента, которое показывает степень достаточности основных функций для решения задач в соответствии с назначением ПО;
  + правильность (точность) - атрибут, который показывает степень достижения правильных результатов;
  + интероперабельность - атрибут, который показывает возможность взаимодействовать на ПО специальными системами и средами (ОС, сеть);
  + защищенность - атрибут, который показывает на способность ПО предотвращать несанкционированный доступ (случайный или умышленный) к программам и данным.

1. Надежность - совокупность атрибутов, которые определяют способность ПО преобразовывать исходные данные в результаты при условиях, зависящих от периода времени жизни (износ и его старение не учитываются). Снижение надежности ПО происходит из-за ошибок в требованиях, проектировании и выполнении. Отказы и ошибки в программах появляются на заданном промежутке времени.

К подхарактеристикам надежности ПО относятся:

* + безотказность - атрибут, который определяет способность ПО функционировать без отказов (как программы, так и оборудования);
  + устойчивость к ошибкам - атрибут, который показывает на способность ПО выполнять функции при аномальных условиях (сбой аппаратуры, ошибки в данных и интерфейсах, нарушение в действиях оператора и др.);
  + восстанавливаемость - атрибут, который показывает на способность программы к перезапуску для повторного выполнения и восстановления данных после отказов.

К некоторым типам систем (реального времени, радарных, систем безопасности, коммуникация и др.) предъявляются требования для обеспечения высокой надежности (недопустимость ошибок, точность, достоверность, удобство применения и др.). Таким образом, надежность ПО в значительной степени зависит от числа оставшихся и не устраненных ошибок в процессе разработки на этапах ЖЦ. В ходе эксплуатации ошибки обнаруживаются и устраняются.

Если при исправлении ошибок не вносятся новые или, по крайней мере, новых ошибок вносится меньше, чем устраняется, то в ходе эксплуатации надежность ПО непрерывно возрастает. Чем интенсивнее проводится эксплуатация, тем интенсивнее выявляются ошибки и быстрее растет надежность ПО.

К факторам, влияющим на надежность ПО, относятся:

* + совокупность угроз, приводящих к неблагоприятным последствиям и к ущербу системы или среды ее функционирования;
  + угроза как проявление нарушения безопасности системы;
  + целостность как способность системы сохранять устойчивость работы и не иметь риска.

Обнаруженные ошибки могут быть результатом угрозы извне или отказов, они повышают риск и уменьшают некоторые свойства надежности системы.

Надежность - одна из ключевых проблем современных программных систем, и ее роль будет постоянно возрастать, поскольку постоянно повышаются требования к качеству компьютерных систем. Новое направление - инженерия программной надежности (Software reliability engineering) - ориентировано на количественное изучение операционного поведения компонентов системы по отношению к пользователю, ожидающему надежную работу системы, и включает:

* + измерение надежности, т.е. проведение ее количественной оценки с помощью предсказаний, сбора данных о поведении системы в процессе эксплуатации и современных моделей надежности;
  + стратегии и метрики конструирования и выбора готовых компонентов, процесс разработки компонентной системы, а также среда функционирования, влияющая на надежность работы системы;
  + применение современных методов инспектирования, верификации, валидации и тестирования при разработке систем, а также при эксплуатации.

Верификация применяется для определения соответствия готового ПО установленным спецификациям, а валидация - для установления соответствия системы требованиям пользователя, которые были предъявлены заказчиком.

В инженерии надежности термин dependability (пригодноспособность) обозначает способность системы иметь свойства, желательные для пользователя, который уверен в качественном выполнении функций ПС, заданных в требованиях. Данный термин определяется дополнительным количеством атрибутов, которыми должна обладать система, а именно:

* + готовность к использованию (availability);
  + готовностью к непрерывному функционированию (reliability);
  + безопасность для окружающей среды, т.е. способность системы не вызывать катастрофических последствий в случае отказа (safety);
  + секретность и сохранность информации (сonfidential);
  + способность к сохранению системы и устойчивости к самопроизвольному ее изменению (integrity);
  + способность к эксплуатации ПО, простота выполнения операций обслуживания, а также устранения ошибок, восстановление системы после их устранения и т.п. (maintainability);
  + готовность и сохранность информации (security) и др.

Достижение надежности системы обеспечивается предотвращением отказа (fault prevention), его устранением (removal fault), а также оценкой возможности появления новых отказов и мер борьбы с ними с применением методов теории вероятности.

Оценка надежности ПО - это трудоемкий процесс, требующий создания устойчивой работы системы по отношению к отказам ПО, т.е. вероятности того, что система восстановится самопроизвольно в некоторой точке после возникновения в ней отказа (fault).

1. Удобство применения характеризуется множеством атрибутов, которые показывают на необходимые и пригодные условия использования (диалоговое или не диалоговое) ПО заданным кругом пользователей для получения соответствующих результатов. В стандарте удобство применения определено как специфическое множество атрибутов программного продукта, характеризующих его эргономичность.

К подхарактеристиками удобства применения относятся:

* + понимаемость - атрибут, который определяет усилия, затрачиваемые на распознавание логических концепций и условий применения ПО;
  + изучаемость (легкость изучения) - атрибут, который определяет усилия пользователей на определение применимости ПО путем использования операционного контроля, диагностики, а также процедур, правил и документации;
  + оперативность - атрибут, который показывает на реакцию системы при выполнении операций и операционного контроля;
  + согласованность - атрибут, который показывает соответствие разработки требованиям стандартов, соглашений, правил, законов и предписаний.

1. Эффективность - множество атрибутов, которые определяют взаимосвязь уровней выполнения ПО, использования ресурсов (средства, аппаратура, материалы - бумага для печатающего устройства и др.) и услуг, выполняемых штатным обслуживающим персоналом и др.

К подхарактеристикам эффективности ПО относятся:

* + реактивность - атрибут, который показывает время отклика, обработки и выполнения функций;
  + эффективность ресурсов - атрибут, показывающий количество и продолжительность используемых ресурсов при выполнении функций ПО;
  + согласованность - атрибут, который показывает соответствие данного атрибута с заданными стандартами, правилами и предписаниями.

1. Сопровождаемость - множество свойств, которые показывают на усилия, которые надо затратить на проведение модификаций, включающих корректировку, усовершенствование и адаптацию ПО при изменении среды, требований или функциональных спецификаций.

Cопровождаемость включает подхарактеристики:

* + анализируемость - атрибут, определяющий необходимые усилия для диагностики отказов или идентификации частей, которые будут модифицироваться;изменяемость - атрибут, который определяет удаление ошибок в ПО или внесение изменений для их устранения, а также введение новых возможностей в ПО или в среду функционирования;
  + стабильность - атрибут, указывающий на постоянство структуры и риск ее модификации;
  + тестируемость - атрибут, показывающий на усилия при проведении валидации и верификации с целью обнаружения несоответствий требованиям, а также на необходимость проведения модификации ПО и сертификации;
  + согласованность - атрибут, который показывает соответствие данного атрибута соглашениям, правилам и предписаниям стандарта.

1. Переносимость - множество показателей, указывающих на способность ПО адаптироваться к работе в новых условиях среды выполнения. Среда может быть организационной, аппаратной и программной. Поэтому перенос ПО в новую среду выполнения может быть связан с совокупностью действий, направленных на обеспечение его функционирования в среде, отличной от той среды, в которой оно создавалось с учетом новых программных, организационных и технических возможностей.

Переносимость включает подхарактеристики:

* + адаптивность - атрибут, определяющий усилия, затрачиваемые на адаптацию к различным средам;
  + настраиваемость (простота инсталляции) - атрибут, который определяет необходимые усилия для запуска данного ПО в специальной среде;
  + сосуществование - атрибут, который определяет возможность использования специального ПО в среде действующей системы;
  + заменяемость - атрибут, который обеспечивают возможность интероперабельности при совместной работе с другими программами с необходимой инсталляцией или адаптацией ПО;
  + согласованность - атрибут, который показывает на соответствие стандартам или соглашениями по обеспечению переноса ПО.

# Серия стандартов ISO/IEC 25000. Структура

Структура серии стандартов SQuaRE

На данный момент времени ISO и IEC ведут активную работу по созданию серии стандартов с названием «Системная и программная инженерия – Требования к качеству и оценка программного продукта». Эта серия заменит серии стандартов ISO/IEC 9126-1-4:2001-2004 и ISO/IEC 1598-1-6:1998-2001.

Основными преимуществами серии стандартов SQuaRE являются:

· координация руководств по измерению и оценке качества программных продуктов;

· наличие руководства по спецификации требований к качеству программного продукта;

· гармонизация со стандартом ISO/IEC 15939:2007 в форме эталонной модели измерений качества.

Серия стандартов SQuaRE разделена на следующие группы (разделы):

1. **ISO/IEC 2500n – группа управления качеством**. Стандарты из данной группы определяют общие модели, термины и определения, которые используются в остальных стандартах серии SQuaRE. Данная группа стандартов содержит также руководство по использованию стандартов серии SQuaRE.
2. **ISO/IEC 2501n – группа модели качества**. В стандартах данной группы представлены подробные модели качества для компьютерных систем и программных продуктов, качества в использовании и качества данных. Данная группа стандартов содержит также практическое руководство по использованию представленных моделей качества.
3. **ISO/IEC 2502n – группа измерения качества**. Стандарты данной группы включают эталонную модель измерений качества программного продукта, математические определения мер качества и практическое руководство по их применению. Даются примеры внутренних и внешних мер качества программных продуктов и систем, а также мер качества в использовании. Определены и представлены элементы мер качества, являющиеся основой этих мер.
4. **ISO/IEC 2503n – группа требований к качеству**. Стандарты данной группы помогают определить требования к качеству, основываясь на моделях и мерах качества. Эти требования к качеству могут использоваться в процессе выявления требований к качеству разрабатываемого программного продукта или как входные данные для процесса оценки.

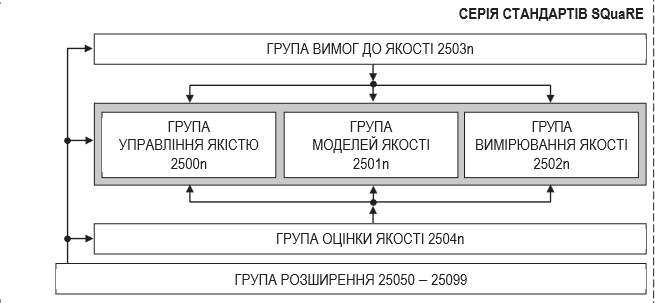


Рис.1 Взаимосвязь и организация серии стандартов SQuaRE

Группа стандартов ISO/IEC 25000

|  |  |
| --- | --- |
| ISO/IEC 25000 – Группа управления качеством | |
| ISO/IEC 25000  Программная инженерия – требования к качеству и оценка программного продукта SQuaRE | ISO/IEC 25000  Программная инженерия – требования к качеству и оценка программного продукта – планирование и управления |

Табл.1 Группа стандартов ISO/IEC 25000

Стандарт ISO/IEC 25000:2005 включает в себя:

* Основные термины и определения в области оценки качества ПО и систем;
* Описание структуры серии стандартов SQuaRE;
* Описание взаимосвязей стандартов серии SQuaRE, ISO/IEC 9126 и ISO/IEC 14598.

В стандарте ISO/IEC 25001:2007 представлено:

* Концепции управления оценками;
* Рекомендации касательно определения требований к качеству и оценки качества;
* Образец плана оценки качества.

Группу стандартов ISO/IEC 25001 представлено в виде табл.2:

|  |  |
| --- | --- |
| ISO/IEC 25001 – Группа управлением качества | |
| ISO/IEC 25010:2011  Системная и программная инженерия – требования к качеству и оценки программного продукта – модели качества систем и программных средств | ISO/IEC 25012:2008  Программная инженерия – Требования к качеству и оценка программного продукта – модель качества данных |

Согласно со стандартом ISO/IEC 25010:2011, качество системы – это степень удовлетворения требований системой, как заданных, так и тех, которые имеет ввиду пользователь. Эти требования представляются моделями качества, приведенными в стандарте ISO/IEC 25010:2011. Модели имеют иерархичную структуру, где верхний уровень - это характеристики, которые, по требованию, могут разделяться на подхарактеристики.

Измеримые свойства, связаны с качеством системы или продукта, в стандарте ISO/IEC 25010:2011 называют свойствами качества (или мерами качества). Согласно со стандартом, чтобы получить меру характеристики или подхарактеристики качества без их непосредственного измерения, необходимо:

1. Определить совокупность свойств, которые одновременно покрывают характеристику и подхарактеристику;
2. Получить меру качества для каждого свойства;
3. Объединить полученные меры с помощью вычислений для получения результирующей меры качества, связанной с характеристикой или подхарактеристикой качества.

ISO/IEC 25010. (SQuaRE) Модели: качества в использовании, качества программного продукта, качества данных

В стандарте ISO/IEC 25010:2011 определено две модели качества:

Модель 1: модель качества в использовании, что состоит из 5 характеристик, которые имеют отношение к результатам взаимодействия продукта при его использовании в заданном контексте использования.

Модель 2: модель качества продукт, что состоит из восьми характеристик, которые имеют отношение к статичным свойства ПО и динамичным свойствам компьютерной системы.

Модель 1 является моделью системы, которая применяется к системам «человек-компьютер». Модель 2 может быть применена как к компьютерным системам, так и к программным продуктам.

Стандартом ISO/IEC 25010:2011 предполагается использование «модели качества в использовании».

Качество в использовании – это степень применимости продукта или системы, заданными пользователями для удовлетворения их потребностей в достижении поставленных целей с результативностью, эффективностью, свободой от риска и удовлетворенностью в заданных контекстах использования. Понятие контекста использования свойственно как для качества в использовании, так и для качества продукта. Во втором случае оно определяется стандартом как заданные условия.

Качество в использовании характеризует влияние, которое продукт совершает на правовладельцев. Он определяется качеством ПО, аппаратных средств и эксплуатационной среды, а также характеристиками пользователей, заданий и окружения.

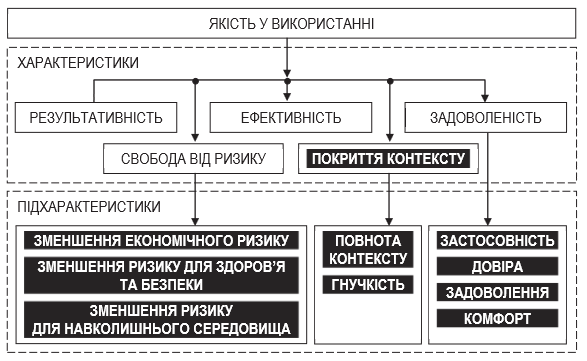


Рис.2 Модель качества в использовании согласно стандарту ISO/IEC 25010:2011

# Методы измерения. Метрики.

Измерение – выраженное в количественных величинах сокращение неопределенности в оценке состояния объекта управления. Измерение является одним из аспектов управления, и его важность возрастает с ростом сложности объекта управления.

Методы измерения:

* Инструментальный – с помощью контрольного или измерительного оборудования;
* Социологический – анкетирование;
* Экспертный – оценки специалистов;
* Расчетный (статистический) – математическая обработка массивов данных или выборок.

Метрики программного обеспечения – искусственно введенная численная мера, позволяющая выделить и оценить определенные свойства программного обеспечения либо его спецификаций.

Классификация метрик:

* метрики оценки программного обеспечения;
* метрики оценки условий разработки программного обеспечения.

В каждой из этих категорий могут использоваться метрики с разными шкалами оценки свойств программного обеспечения: номинальной, порядковой, интервальной и относительной.

* Номинальная шкала показывают наличие либо отсутствие у программного обеспечения какого-либо свойства.
* Порядковой шкале соответствуют метрики, служащие для ранжирования определенных характеристик с помощью сравнения с некоторыми опорными значениями. Измерения по этой шкале по факту показывают взаиморасположение конкретных программ.
* Интервальная шкала показывает уже не только взаиморасположение программ, но и то, насколько далеко они отстоят друг от друга.
* Относительная шкала позволяет не только разместить программы определенным образом и оценить их размещение относительно друг друга, но также может показать, насколько далеко оценки отстоят от точки, начиная с которой характеристика вообще может быть измерена.

По своему назначению выделяется три типа метрик:

* прогнозирующие (попытаются предсказать те проблемы, с которыми может столкнуться проект);
* диагностические (мониторинг текущего состояния проекта);
* ретроспективные (помогают в будущем избежать ошибок из прошлого опыта).

Прогнозирующие метрики описывают: предполагаемый объем работ, запланированный бюджет, фактор сложности проекта, сложность плана проекта, количество людских ресурсов, суммарный запас времени и т.д. При этом желательно иметь значения этих показателей для других проектов этого же профиля, чтобы иметь возможность для сравнения. Значительные различия в показателях схожих проектов часто указывают на проблемы.

Диагностические метрики описывают: соотношение планового бюджета выполненных работ к фактическому, и соотношение плановых трудозатрат к фактическим, и производительность труда работников, занятых на проекте, и показатели качества, и еще много-много чего еще.

Ретроспективные метрики описывают те же параметры, что и диагностические. Единственная разница между ними в цели использования: если диагностические метрики нужны для оценки текущей ситуации на проекте и реагирования на актуальные проблемы, то ретроспективные помогают оценить уже прошедшие события. Такой ретроспективный анализ позволяет избежать типовых ошибок в будущем.

Таким образом, использование метрик позволяет эффективно управлять проектом: своевременно обнаруживать недостатки, устранять их и при этом видеть, насколько хорошо работают найденные методы решения проблем.

# «Гарантоспособность» критического ПО

Гарантоспособная компьютерная система (ГКС) – это система, обладающая полным или частичным набором первичных свойств (атрибутов), составляющих гарантоспособность. Иными словами ГКС – отказоустойчивая, высоконадежная, безопасная система с гарантированно достоверными вычислениями. Также, гарантоспособность это способность компьютерных систем предоставлять требуемые услуги, которым можно оправданно управлять.

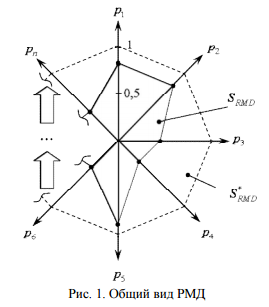
Ключевым фактором обеспечения характеристик гарантоспособности (надежности, готовности, обслуживаемости) и функциональной безопасности критического ПО являются информационно-управляющие системы (ИУС). Уровень критичности ИУС определяются от тяжести последствий неправильного функционирования. В силу этого возрастает зависимость безопасности от качества критического ПО. Не выявленные на этапах тестирования дефекты ПО являются факторами риска аномального функционирования систем, что может привести к плохим последствиям. Это обуславливает статус критического ПО, как важного элемента нормативного регулирования, определяющего гарантоспособность и функциональную безопасность ИУС в целом. Функциональная безопасность критического ПО трактуется как нахождение системы в условиях проектного риса аномального функционирования в течении установленного срока эксплуатации.

Рентабельность оценки характеристик гарантоспособность и функциональной безопасности критического ПО означает достижение приемлемых прогнозируемых уровней рисков, связанных с остаточными дефектами ПО, при допустимых затратах ресурсов.

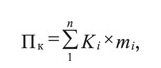
Одним из строго рекомендуемых требования является проведение независимой верификации критического ПО, обеспечивающая:

* независимую верификацию на основе усовершенствованной методологии статического анализа исходного ПО, обеспечивающую повышение достоверности оценок характеристик ПО и прогнозирование рисков аномального функционирования из-за остаточных дефектов ПО;
* интегрированную инструментальную поддержку сценария независимой верификации критического ПО на аналитическом, информационном и организационном уровнях;
* рентабельную оценку и прогнозирование рисков, связанных с остаточными дефектами критического ПО.

# Аппарат радиальных метрических диаграмм (РМД)

РМД задает n-мерное метрическое пространство, в котором осуществляется оценивание объекта, где n – число метрик pi,i= отображающихся в виде лучей диаграммы. 

В случае, когда построены несколько РМД, оценивающих объект в целом, они образуют иерархическую структуру. В результате свертки отдельной РМД нижнего уровня на основании значений метрик и коэффициентов их значимости



где Ki – значение оценки i-го единичного показателя, mi– значение коэффициента весомости i-го показателя; n – количество показателей в данной группе.

формируется обобщенный показатель, значение которого затем откладывается на соответствующем луче комплексной РМД верхнего уровня. Комплексные показатели находятся на более высоких уровнях, чем единичные.

Аналогичным образом для комплексной РМД может быть получен интегральный показатель. Рассмотрим вариант использования РМД для каждого этапа ЖЦ ПО применительно к V – образной модели. На каждом этапе разработки по уровням РМД рассчитываются следующие показатели:

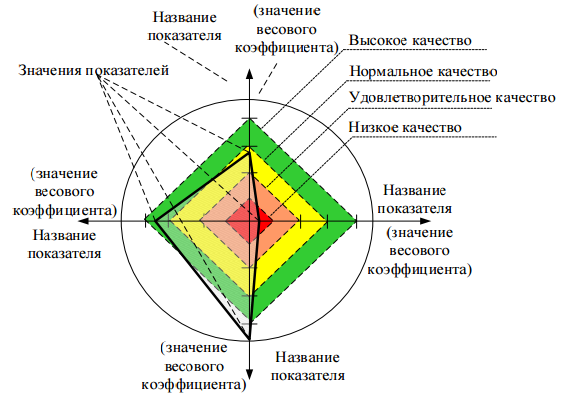
− на первом уровне пары показателей прогнозируемый профиль дефектов, показатель полноты процесса тестирования и верификации типов дефектов и показатель полноты качества процесса тестирования и верификации дефектов, показатель полноты верификации типов дефектов, на основании которых при аддитивной свёртке формируются показатели следующего уровня РМД;

− на втором уровне пара показателей ПП, ППВ, на основании которых при аддитивной свёртке формируется показатель следующего уровня РМД;

− на третьем уровне показатель ЧП;

− на четвёртом уровне показатель ОП, формируемый на основе свёртки показателей ЧП, количество которых зависит от количества этапов, для которых проводится оценка качества тестирования и верификации.

Обозначения РМД



# Заключение

В результате проведенного анализа программного продукта SQuaRE, нами были рассмотрены темы:

* Идентификация, анализ и детализация объекта изученияSQuaRE.
* Раскрытие спецификации (детализации структуры) анализируемых процессов в соответствии с ISO/IEC/IEEE 12207:2008
* Модели качества ПО в различных контекстах.
* Серия стандартов ISO/IEC 25000. Структура.
* ISO/IEC 25010. (SQuaRE) Модели качества
* Методы измерения. Метрики.
* Многокомпонентность характеристики «Гарантоспособность» критического ПО.
* Параметризация моделей качества ПО, определение измеряемых параметров модели и соответствующих метрик.
* Усовершенствование подхода на основе Kiviat – diagrams. Аппарат радиально – метрических диаграмм. Иерархия. Свертка.

Таким образом, программный продукт SQuaRE позволяет значительно облегчить выявление несоответствий заявленным требованиям ПО. Серия стандартов структурирует оценку качества программного продукта, тем самым сокращая время, затраченное на разработку ПО.

# Список литературы:

1. В.В.Липаев, Тестирование крупных комплексов программ на соответствие требованиям – М.-Б.: Директ-Медиа, 2015. – 377 с.
2. Ю.Г. Карпов, Качество ПО КС – СПб.: БХВ-Петербург, 2010. – 560 с.
3. ISO/IEC 25010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE)
4. [Электронный источник]: http://www.pqm-online.com/assets/files/pubs/translations/std/iso-9000-2015-(rus).pdf
5. [Электронный источник]: <https://www.khai.edu/csp/nauchportal/Arhiv/REKS/2007/REKS607/pdf/Konorev.pdf>
6. [Электронный источник]: https://www.projectmanagement.com