

# Модели качества программного продукта SQuaRE

## Введение

Инженерия разработки программного обеспечения (ПО) – это применение систематического подхода, стандартов и количественного измерения характеристик ПО к разработке, использованию и сопровождению программного обеспечения. В связи с развитием технологий важность программной инженерии постоянно растет, а соответственно все более востребованным являются методы определения качества программного обеспечения. Сложность процесса разработки и сопровождения программного обеспечения во многом обуславливается особыми требованиями, предъявляемыми к его качеству. Этот фактор обосновывает важность разработки формализованных методов управления качеством ПО. В настоящий момент используются несколько определений понятия качества ПО, которые в целом совместимы друг с другом. Обобщая определения стандартов, можно заключить, что качество программного обеспечения – это способность программного продукта (ПП) к удовлетворению установленных или предполагаемых потребностей при использовании в заданных условиях.

Качество программного обеспечения играет важную роль для всей системы в целом. Так, качество ПО рассматривается как очень важный аспект для разработчиков, пользователей и руководителей проектов. Качество программного обеспечения – величина отражающая в каком объеме в программный продукт включен набор желаемых функций для повышения эффективности ПП в течение жизненного цикла. Для любой системы, использующей программное обеспечение, должны быть разработаны три вида спецификаций, такие как функциональные требования, требования к качеству, требования к ресурсам. Качество включает в себя все характеристики и существенные особенности продукта или его функционирования, которые относятся к удовлетворению заданных спецификациями требований.

Назначение стандарта SQuaRE состоит в том, чтобы определить и/или разработать начальное множество элементов показателя качества, которые будут использоваться повсеместно в течение жизненного цикла продукции в интересах выработки требований и оценки качества систем и программного обеспечения (SQuaRE). В настоящем стандарте также приведен ряд правил для проектирования элементов показателей качества (ЭПК) или верификации проекта существующего ЭПК.

## Глоссарий

**оценка, оценивание, оценка** (evaluation): Систематическое определение степени, с которой определенный объект удовлетворяет установленным критериям.

**действия по оценке** (evaluation activity): Оценка систем или программной продукции на соответствие целевым значениям определенных и применяемых характеристик качества, выполняемая с использованием прикладных методов и методик.

**группа оценки** (evaluation group): Организация, ответственная за задание требований к качеству си

стем и программного обеспечения, а также за реализацию и управление действиями по оценке качества систем и программного обеспечения путем обеспечения технологий, инструментариев, опыта и управленческих навыков.

**технология оценки (технология, используемая для оценки)** [evaluation technology (technology used for evaluation)]: Методики, процессы, инструментарии, показатели и соответствующая техническая информация, используемые для оценки.

**методики, техника** (techniques): Методы и навыки, требуемые для выполнения определенной деятельности.

**качество данных** (data quality): Степень, с которой характеристики данных удовлетворяют заявленным и подразумеваемым требованиям при использовании в заданных условиях (ИСО/МЭК 25012).

**показатель внешнего качества программного обеспечения** (external measure of software quality): Показатель степени, с которой программная продукция позволяет поведению (функционированию) системы удовлетворять заявленным и реализованным требованиям к этой системе, включая программное обеспечение при использовании в заданных условиях.

**показатель внутреннего качества программного обеспечения** (internal measure of software quality): Показатель степени, с которой множество статических свойств программной продукции удовлетворяет заявленным и подразумеваемым требованиям для этой продукции при использовании в заданных условиях.

**показатель (measure):** Переменная, в которой значение определено результатом измерения.

**модель (model):** Спецификация понятий, отношений и правил, используемых для определения методологии (ИСО/МЭК 24744 Программная инженерия. Модель для методологий разработки).

**свойство для количественного определения (property to quantify):** Свойство целевой сущности, которое имеет отношение к элементу показателя качества и которое может быть определено количественно с помощью метода измерения.

**элемент показателя качества (ЭПК) (quality measure element (QME):** Показатель, определенный в терминах свойства и метода измерения для количественного определения этого свойства, включая выборочно преобразования с помощью математической функции.

**повторяемость (результатов измерения) (repeatability (of results of measurement)):** Близость соответствия между результатами последовательных измерений того же самого показателя, выполненных при тех же самых условиях измерения (ТО ИСО/МЭК, 14143-3).

**воспроизводимость (результатов измерения) (reproducibility (of results of measurement)):** Близость соответствия между результатами последовательных измерений того же самого показателя, выполненных при измененных условиях измерения (ТО ИСО/МЭК 14143-3).

**целевой объект (целевая сущность) (target entity):** Фундаментальное по отношению к пользователю понятие, информация о котором хранится и которая подлежит измерению.

**полноценность (usefulness):** Степень удовлетворенности пользователя достижением прагматических целей, включая результаты использования и последствия использования.

**доверие (trust):** Степень уверенности пользователя или другого заинтересованного лица в том, что продукт или система будут выполнять свои функции так, как это предполагалось.

**свобода от риска** (freedom from risk): Способность продукта или системы смягчать потенциальный риск для экономического положения, жизни, здоровья или окружающей среды.

**функциональная пригодность** (functional suitability): Степень, в которой продукт или система обеспечивают выполнение функции в соответствии с заявленными и подразумеваемыми потребностями при использовании в указанных условиях.

Для выполнения разнообразных функций как в бизнесе, так и для персонального назначения в современных условиях все большее распространение получают программные продукты и преимущественно программные вычислительные системы. Реализация целей и задач для удовлетворения личных потребностей, для успеха в бизнесе и/или для безопасности человека опирается на высококачественное программное обеспечение и системы. Высококачественные программные продукты и преимущественно программные вычислительные системы имеют важное для заинтересованных сторон значение в производстве материальных ценностей и предотвращении возможных негативных последствий.

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.

## **Сокращения**

**ЭПК** – элемент показателя качества (quality measure element);

**ПК** – показатель качества (quality measure).

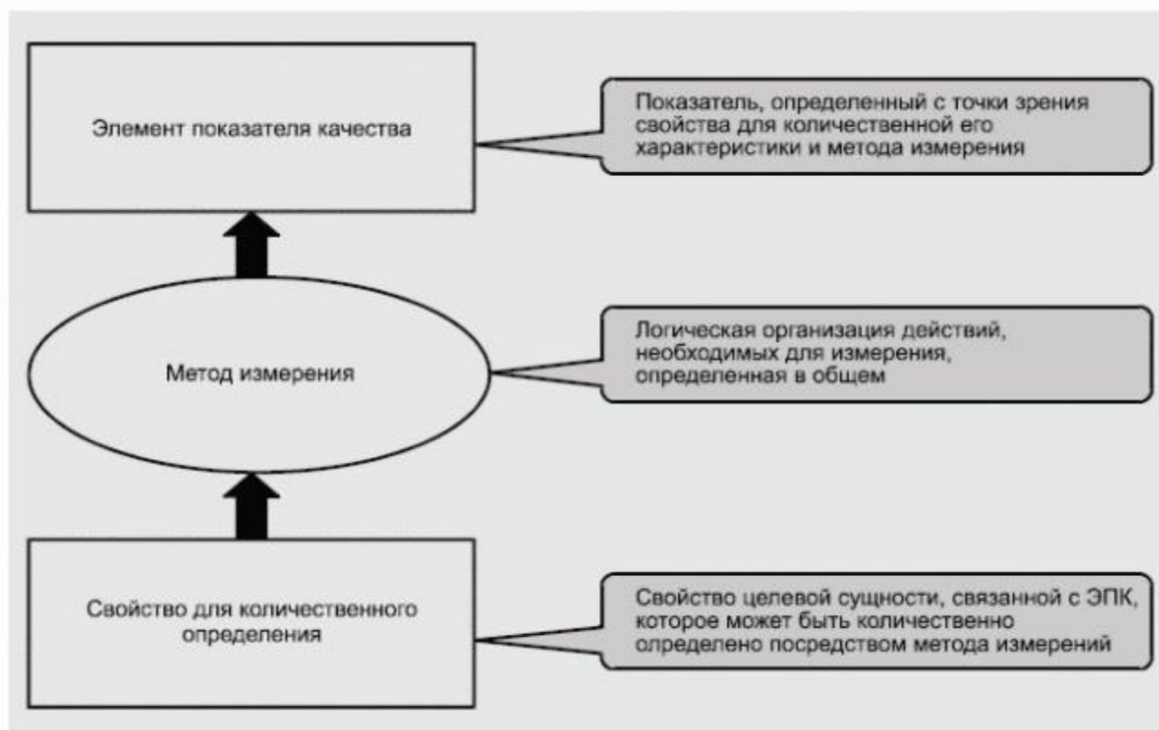
## **Основная часть**

Определение для выбранного состава процессов на основе аналитических обзоров рекомендованной литературы предметной области выбранной темы, представленной соответствующей методологией и технологиями в терминах: **абстракции, модели, методы, меры и метрики, информационные технологии, руководства и стандарты.**

### **Модель метода измерения**

ПК и соответственно ЭПК определяют для понимания и указания характеристик и подхарактеристик качества.

Функция измерения применена к ЭПК для генерации ПК. Метод измерения должен быть применен к свойству для установления и идентификации способа количественного определения ЭПК.



**Рисунок 1 - Взаимосвязь между свойством количественного определения, метода измерения и ЭПК**

Пользователь метода измерения должен идентифицировать и собирать данные, связанные с количественным определением свойства (см. рисунок 1). В зависимости от контекста использования и целей ЭПК ряд свойств и подсвойств может быть идентифицирован. Они и являются входными данными для метода измерений. Эти свойства определяются и извлекаются из артефактов, компонентов, содержимого или поведения целевого объекта (например, документации, кода).

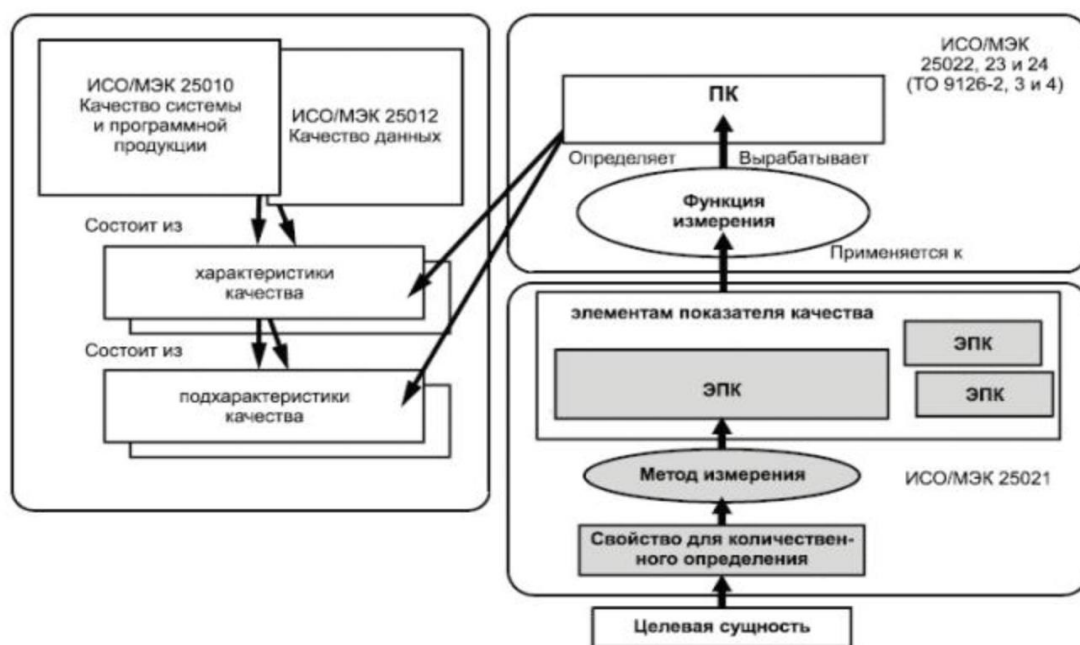


Рисунок 2 - Взаимосвязь свойства количественного определения, метода измерения, ЭПК и ПК

На рисунке показано, что:

- качество продукции выражено как множество характеристик качества, которые состоят из подхарактеристик;
- показатели качества продукции используют для указания интереса с помощью характеристик и подхарактеристик качества;
- существует взаимосвязь свойства количественного определения, метода измерения и ЭПК.

Примечание. Рисунок базируется на определении эталонной модели измерения качества систем и программной продукции (ИСО/МЭК 25020).

Данная модель подразделяет свойства качества системы или программного продукта на восемь **характеристик**:

- Функциональное соответствие** — степень обеспечения программным продуктом или системой функций, отвечающих заданным или подразумеваемым потребностям при использовании в

заданных условиях. **Подхарактеристики:** функциональная полнота, функциональная правильность, функциональная пригодность.

2. **Эффективность функционирования** – зависимость функционирования от количества ресурсов, используемых в заданных условиях. Ресурсы могут включать другие программные продукты, программную и аппаратную конфигурацию системы, материалы. **Подхарактеристики:** поведение во времени, использование ресурсов, емкость.
3. **Совместимость** – степень возможностей программного продукта, системы или компонента обмениваться информацией с другими продуктами, системами или компонентами и/или выполнять свои заданные функции при совместном использовании одной и той же аппаратной или программной среды. **Подхарактеристики:** сосуществование, способность к взаимодействию.
4. **Практичность** – степень применимости программного продукта или системы заданными пользователями для достижения заданных целей с результативностью, эффективностью и удовлетворенностью в заданном контексте использования. **Подхарактеристики:** распознаваемость пригодности, обучаемость, простота использования, защита от ошибок пользователей, эстетичность пользовательского интерфейса, доступность.
5. **Надежность** – степень выполнения системой, программным продуктом или компонентом заданных функций в заданных условиях в течении заданного периода времени. **Подхарактеристики:** завершенность, готовность, устойчивость к ошибке, восстанавливаемость.
6. **Защищенность** – степень защиты программным продуктом или системой информации и данных так, чтобы люди, другие продукты или системы имели степень доступа к данным, соответствующую типам и уровням их авторизации. **Подхарактеристики:** конфиденциальность, целостность, неопровержимость, идентифицируемость, аутентичность.
7. **Сопровождаемость** – степень результативности и продуктивности модификации программного продукта или системы запланированным персоналом сопровождения. Модификации включают: исправления, улучшения, адаптация к новым требованиям. **Подхарактеристики:** модульность, повторная



используемость, анализируемость, модифицируемость, тестируемость.

8. **Мобильность** – степень результативности и эффективности переноса системы, программного продукта или компонента из одной аппаратной, программной или иной эксплуатационной или используемой среды в другую. **Подхарактеристики:** адаптируемость, простота установки, взаимозаменяемость.

К числу важнейших характеристик, определяющих качество критического ПО наряду с **Функциональной** и **Информационной безопасностью (Safety, Security)** относится характеристика **Гарантоспособность (Dependability)**. В общем случае **Гарантоспособность** основывается на методах достижения бездефектности, включая предупреждение дефектов при проектировании, обнаружение, диагностирование и «извлечение дефектов» (коррекции ПО) при тестировании и парирование оставшихся скрытых дефектов при использовании критического ПО на основе реализации принципа разнообразия (многоверсионности).

Оценка (прогноз) вероятности существования скрытых (латентных) дефектов критического ПО является исходной позицией для определения характеристик **Гарантоспособность** и **Безопасность** критического ПО.

Именно необходимость достижения упомянутых выше показателей при разработке критического ПО определяет специфические задачи и особенности технологий критической ПИ. Специфическим требованием к таким технологиям является достижение предельно-возможных значений бездефектности в результате верификации, валидации и квалификационных испытаний критического ПО. В силу того, что риски ущерба из-за скрытых дефектов критического ПО могут быть неприемлемо высокими.

### **Табличный формат элементов показателей качества**

Элементы информации, перечисленные в форме таблицы элементов ЭПК (см. таблицу 1), следует использовать для предоставления необходимой или полезной информации при определении ГОСТ Р ИСО/МЭК 25021-2014 Информационные технологии (ИТ). Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Элементы показателя качества и/или проектировании ЭПК.

**Примечание.** Объекты ниже приведенной таблицы структурированы в четыре группы, назначение которых: а) для идентификации ЭПК, б) - д) для определения, что представляет собой ЭПК, е) - к) для определения, как измерить ЭПК и л) - н) для управления приложением ЭПК.

**Таблица 1** - Табличный формат ЭПК

а) Имя ЭПК	ЭПК должен иметь уникальное имя и при необходимости должен быть идентифицирован порядковым номером. В большинстве случаев, имя начинается с "число... (шкала отношений)".
б) Целевой объект (целевая сущность)	<p>У ЭПК должен быть целевой объект, характеристики которого должны быть получены путем измерения его свойств.</p> <p>Целевой объект должен представлять собой результат работы или поведение системы, программного обеспечения или заинтересованных лиц, таких как пользователи, операторы, разработчики, тестировщики или специалисты по сопровождению.</p>
с) Цели и свойство, которое должно быть выражено количественно	<p>Идентификация свойства, которое должно быть выражено количественно, как правило, связано с именем ЭПК. Выбранным свойством, которое должно быть выражено количественно, должно быть то свойство, которое имеет самое непосредственное отношение к необходимому измерению. Данное свойство может быть включено в несколько измерительных моделей. Например, число отказов программного обеспечения - это ЭПК, а отказ - это свойство программного обеспечения, количественное выражение которого нужно получить. Цель ЭПК должна быть специфицирована вместе с определением свойства, которое нужно выразить количественно. Необходимо ответить на вопросы:</p> <p>- что предполагается узнать с помощью определения свойства, выраженного количественно с использованием ЭПК?;</p>

	<p>- какую необходимую информацию ожидается получить от этого ЭПК?</p> <p>Из идентификации и определения свойства, которое выражается количественно, следует определение, что должно быть оценено (например, строки кода, дефекты, продолжительность).</p> <p>Полезно определить, какие виды компонентов или событий в указанном целевом объекте должны быть идентифицированы, определены и выражены количественно.</p> <p><b>Примеры:</b></p> <p><i>1 Могут быть идентифицированы, определены и выражены количественно строки, функции, пути или метки в исходном коде программы, предназначенные для выполнения определенной функции;</i></p> <p><i>2 могут быть идентифицированы, определены и выражены количественно все события тестовых ошибок и случаев при тестировании программного обеспечения;</i></p> <p><i>3 могут быть идентифицированы, определены и выражены количественно все события, когда пользователь системы не в состоянии выполнять предназначенные для него задачи.</i></p>
d) Релевантный показатель (показатели) качества	<p>Должна быть определена ссылка на определенный показатель (показатели) качества, в которой используется этот ЭПК.</p> <p>Примеры показателей качества можно найти в документах серии ИСО/МЭК 9126, серии 25000 SQuaRE и других документах. Составления исчерпывающего списка показателей качества не требуется</p>
е) Метод измерения	<p>Метод измерения объясняет, как собрать данные и как преобразовать их в значение, количественно</p>

	<p>определяющее свойство посредством вычислительного правила. В состав метода измерения входит следующее: контекст применения ЭПК, процесс жизненного цикла программного обеспечения, ограничения измерений и вычислительные правила.</p> <p>Измеряющий может дать методу измерения дополнительное имя для того, чтобы было проще отличать его от имени ЭПК и имени свойства для измерения. Например, функциональные измерительные методы могли бы быть иметь имена: IFPUG FPA, COSMIC, Mark II и т.д.</p>
f) Список подсвойств, имеющих отношение к свойству для измерения (не обязательно)	<p>Свойство, определенное для измерения, можно связать, если это необходимо, с различными подсвойствами. Такая связь между свойствами должна быть выражена либо в виде схемы, либо как формула. Это и составляет суть модели метода измерений.</p> <p>Например, в методе COSMIC, процесс функционирования является единым свойством, которое может быть выражено в модели посредством некоторых подсвойств, таких, как вход, чтение, запись и выход. Этот подход может помочь идентифицировать свойство для количественного определения "перемещения данных", которое соответствует показателю, основанному на функциональном размере</p>
g) Определение каждого подсвойства (не обязательно)	<p>Если есть список подсвойств, то должно быть определено и каждое подсвойство</p>
h) Входные данные для ЭПК	<p>Входные данные должны быть описаны настолько подробно, чтобы можно было идентифицировать, какие количественные данные используются для измерения ЭПК. Любые источники входных данных, такие как задокументированные результаты работы, функциональные возможности системы и программного обеспечения, или человеческие функциональные возможности пользователей, операторов, разработчиков, татуировщиков или специалистов по обслуживанию должны также быть</p>

	<p>идентифицированы. Кроме того, входными данными могут быть подсвойства или относящиеся к ним количественные данные.</p> <p>Например, измеряющий может определить в модели данных информацию, необходимую для отслеживания действий типа чтения (перемещение данных) в единице функциональности COSMIC</p>
i) Единица измерения для ЭПК	Единица измерения и при необходимости используемая формула. Примеры единиц включают в себя значение X, процент и разряд
j) Правила вычисления	<p>Правило преобразования должно быть определено либо с точки зрения практики (как правило, текстовая форма), либо с теоретической точки зрения (как правило, математическое выражение). При определении правил преобразования зачастую сталкиваются с проблемой внутренней непротиворечивости. Важно, чтобы свойства и подсвойства, которые должны быть измерены, не противоречили друг другу. По этой причине при добавлении двух объектов важно подтвердить, что они связаны общим свойством.</p> <p>Например, измерение числа отказов даст количество отказов. Однако, если есть существенные различия в главных и второстепенных отказах, более точные показатели будут получены путем рассмотрения отдельно главных и второстепенных отказов.</p> <p>Преобразование предполагает, что для каждого свойства и подсвойства применимо предельное значение результата</p>
k) Тип шкалы	Должен быть определен тип шкалы. Шкала может быть номинальной, порядковой, интервальной или относительной (см. приложение D)
l) Контекст применения ЭПК	Это информация о планируемом использовании результатов измерений. Полезно понять возможность использования ЭПК для представления характеристик или подхарактеристик качества, определяя типичные примеры характеристик качества, подхарактеристик качества или показателей

	<p>качества (ПК), для определения которых, главным образом, предназначены результаты измерения ЭПК.</p> <p>Примечание - ЭПК допускается использовать для измерения любых характеристик или подхарактеристик качества несколькими ПК. В данной графе описываются предполагаемые и необходимые условия для целевых объектов (сущностей), их сред и обстоятельств, к которым должен быть применен метод измерения ЭПК</p>
<p>т) Процессы жизненного цикла программного обеспечения</p>	<p>В этой части таблицы должны быть идентифицированы соответствующие типичные процессы жизненного цикла, которые подходят для фактического измерения данного ЭПК относительно целевой сущности, (например, процессы, посредством которых определенный б) Целевой объект (целевая сущность) создан или реализован в степени, достаточной для производства измерения фактического значения ЭПК).</p> <p>Примечания</p> <p>1 В некоторых случаях в отдельных процессах жизненного цикла на основе исторических данных перед фактическим измерением ЭПК может быть доступна оценка. Однако все процессы жизненного цикла, перечисленные в этом разделе относятся к тем процессам, в которых мы можем получить фактические результаты измерения ЭПК. Связанные процессы жизненного цикла после получения фактических данных, дополнительного фактического измерения или использования измеренных результатов также определяются здесь. Например, число отказов кода может быть фактически измерено с использованием анализа кода, инструментов анализа кода или поблочного тестирования во время процесса разработки (кодирование и поблочное тестирование). Кроме того, число отказов кода может быть измерено дополнительно</p>

	<p>после исправления ошибок кода в процессе интегрированного или квалифицированного тестирования.</p> <p>Кроме того, число отказов в коде можно оценить на основе исторических данных, исходя из предполагаемого размера кода, опираясь на количество страниц спецификаций требований.</p> <p>2 Основные процессы жизненного цикла программного обеспечения, такие, как определение требований правообладателей, анализ требований к программному обеспечению, проектирование архитектуры программных средств, детальное проектирование программных средств, конструирование программных средств, комплексирование программных средств, квалификационное тестирование программных средств, инсталляция программных средств, поддержка приемки программных средств, функционирование программных средств, сопровождение программных средств, прекращение применения программных средств и т.д. определены в ИСО/МЭК 12207. Базовые процессы систем, такие как определение требований правообладателей, анализ требований, проектирование архитектуры, реализация, комплексирование, верификация, передача, валидация, функционирование, сопровождение, прекращение применения и т.д., определены в ИСО/МЭК 15288.</p> <p>3 Если используют методологию, в которую входят процессы жизненного цикла, не описанные ни в ИСО/МЭК 12207, ни в ИСО/МЭК 15288, то измеряющий может также привести методологию и отдельные используемые процессы.</p>
п) Ограничения измерений (не обязательно)	<p>Любые ограничения, связанные с методом измерения, должны быть описаны по мере необходимости.</p> <p>ЭПК может иметь ограничения измерений, такие</p>

	<p>как ошибки измерения или отклонения из-за зависимости от таких факторов, как область применения исследования, способ исследования, изменение спецификации или вариантов использования.</p> <p>Примечания</p> <p>1 Число отказов кода, например, может быть различным для недавно разработанного кода и для повторно используемого кода в одинаковых условиях. Каждый из различных способов исследования кода, таких как анализ, прогон, проверка, экспертная проверка, парное программирование, инструменты анализа кода, поблочное тестирование, причинный анализ отказов в интеграционном тестировании и т.д. дает свое отличное от других число отказов кода.</p> <p>2 Например, при подсчете числа дефектов спецификации документ спецификации должен быть доступен и не должен быть изменчивым.</p>
--	---

Пример использования формата таблицы 1 приведен в таблице 2.

**Таблица 2** - Пример использования таблицы 1 для дефекта (кода)

а) Имя ЭПК	Число дефектов (кода)
б) Целевой объект (целевая сущность)	Исходный код программы
в) Цели и свойство, которое нужно измерить количественно	<p>Цель - измерить число дефектов кода относительно спецификаций проекта и/или стандартов программирования.</p> <p>Измерить нужно число ошибочных строк кода. Отказ в данном случае - это свойство, которое нужно определить количественно.</p> <p>Определения отказа: (1) проявление ошибки в программном обеспечении (ИСО/МЭК 24765 Системная и программная инженерия - Словарь)</p>



	<p>и (2) неправильный шаг, процесс или определение данных в компьютерной программе (ИСО/МЭК 24765 Системная и программная инженерия - Словарь).</p> <p>Необходимо отметить, что ошибка, если она имеет место, может вызвать отказ</p>
d) Релевантные показатели качества	<p>Для определения надежности программного обеспечения, используя плотность ошибок:</p> <ul style="list-style-type: none"> <li>- оценивается скорость обнаружения ошибок на этапе кодирования;</li> <li>- оценивается скорость устранения ошибок на этапе кодирования.</li> </ul> <p>Показатели: уровни завершенности (подхарактеристика) и надежности (характеристика) программного обеспечения</p>
e) Метод измерения	<p>Метод измерения ошибок кода программного обеспечения.</p> <p>Необходимо рассмотреть или проанализировать отличия пересмотренного исходного кода программы и идентифицировать исправления кода, которые состоят из измененных строк, добавленных строк и удаленных строк кода.</p> <p>Примечание - Для исключения дефектов в комплексном тестировании, исходный код программы, как правило, пересматривается в результате таких действий на этапах верификации и валидации как анализ кода, поблочное тестирование, анализ причин.</p>
f) Список подсвойств, имеющих отношение к свойству для измерения (дополнительно)	<p>Связанные подсвойства: исполняемые операторы, строки кода с ошибками, исправленные строки кода</p>
д) Определение каждого подсвойства (дополнительно)	<p>Исполняемые операторы: операторы, которые могут быть отнесены к этой категории, такие как операторы с меткой, выражения, операторы</p>

	<p>выбора, операторы цикла и перехода.</p> <p>Невыполняемые операторы: операторы, к которым относятся спецификации объявлений и объявления.</p> <p>Строки кода с ошибками: строки кода, которые содержат ошибки. Ошибочен ли исходный текст, должна определить спецификация.</p> <p>Корректные строки кода: строки кода без ошибок.</p> <p>Примечание - В отдельных случаях возможно, что строки кода корректны, а спецификация должна быть изменена. В таком случае строки кода не должны считаться ошибочными.</p>
h) Входные данные для ЭПК	Исходный код, спецификации проекта и стандарты программирования
i) Единица измерения ЭПК	Строки кода
j) Правила вычисления	<p>Добавление общего числа ошибочных строк кода.</p> <p>С практической точки зрения числовое правило использует следующие измерительные действия:</p> <p>а) Рассматриваем или анализируем отличия пересмотренного исходного кода программы и идентифицируем исправленные строки кода, в число которых входят измененные, добавленные и удаленные строки кода</p>
k) Тип шкалы	Отношение
l) Контекст применения ЭПК	ЭПК главным образом выбран для того, чтобы измерять уровни завершенности (подхарактеристика) и надежности программного обеспечения (характеристика)
m) Процессы жизненного цикла программного обеспечения	Разработка программного обеспечения (кодирование и тестирование программных модулей), процесс реализации
n) Ограничения	Исходные коды должны быть доступны, чтобы

измерения (дополнительно)	<p>обеспечивать возможность сравнения фактических строк кода со спецификациями проекта.</p> <p>Кроме того, устоявшиеся спецификации проекта должны быть доступны для того, чтобы обеспечивать вышеуказанное сравнение и идентификацию ошибок.</p> <p>Для проверки соответствия кодов стандартам необходимо наличие соответствующих инструментариев или контрольного списка</p>
------------------------------	--

### ***Методы измерения. Метрики.***

**Метрика программного обеспечения** – мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций.

**Метрики качества программных средств в использовании** – метрики, измеряющие соответствие продукта потребностям, заданных пользователем, достижения заданных целей с результативностью, безопасностью, продуктивностью и удовлетворением, заданным в контекстах использования. Метрики используются только в процессе эксплуатации ПС в реальной среде окружения. И основаны на измерении поведения типичных пользователей и системы, содержащей данное программное средство.

**Метрики результативности** оценивают, достигают ли задачи, выполняемые пользователем, заданных целей с точностью и полнотой в заданном контексте использования.

**Метрики продуктивности** оценивают ресурсы, которые затрачивают пользователи в соответствии с достигнутой результативностью в заданном контексте использования.

**Метрики безопасности** оценивают уровень риска причинения вреда людям, бизнесу, программному обеспечению, имуществу или окружающей среде в заданном контексте использования.

**Метрики удовлетворенности** оценивают отношение пользователя к использованию продукта в заданном контексте использования.

Метрика должна также быть эффективной по отношению к стоимости. Это значит, что более дорогие метрики должны обеспечивать лучшие результаты оценки.

Разработчик метрики должен доказать ее обоснованность. Метрика должна удовлетворять хотя бы одному из следующих критериев обоснованности метрики:

1. **корреляция**; изменение в значениях характеристик качества (оперативно определенных по результатам измерения основных метрик), обусловленное изменением в значениях метрики, должно определяться линейной зависимостью;
2. **трассировка**; если метрика  $M$  непосредственно связана с величиной характеристики качества  $Q$  (оперативно определенной по результатам измерения основных метрик), то изменение величины  $Q$  ( $T1$ ), имеющейся в момент времени  $T1$ , к величине  $Q$  ( $T2$ ), полученной в момент времени  $T2$ , должно сопровождаться изменением значения метрики от  $M(T1)$  до  $M(T2)$  в том же направлении (например, если увеличивается  $Q$ , то  $M$  тоже увеличивается);
3. **непротиворечивость**; если значения характеристик качества (оперативно полученные по результатам измерения основных метрик)  $Q1, Q2, \dots, Qn$ , связанные с продуктами или процессами 1, 2...,  $n$ , определяются соотношением  $Q1 > Q2 > \dots > Qn$ , то соответствующие значения метрики должны удовлетворять соотношению  $M1 > M2 > \dots > Mn$ .
4. **предсказуемость**; если метрика используется в момент времени  $T1$  для прогноза значения (оперативно полученного по результатам измерения основных метрик) характеристики качества  $Q$  в момент времени  $T2$ , то ошибка прогнозирования, определяемая выражением

$$\frac{\text{прогнозное } Q(T2) - \text{фактическое } Q(T2)}{\text{фактическое } Q(T2)},$$

должна попадать в допустимый диапазон ошибок прогнозирования;

5. **селективность**; метрика должна быть способной различать высокое и низкое качество программного средства.

***Серия стандартов ISO/IEC 25000. Структура. Перспективы развития.***

Основные разделы серии стандартов SQuaRE:

1. ISO/IEC 2500n – раздел "Управление качеством". Международные стандарты, входящие в этот раздел, определяют общие модели, термины и определения, используемые далее во всех других международных стандартах серии SQuaRE. В разделе также представлены требования и методические материалы, касающиеся поддерживающих функций, отвечающих за управление требованиями и оценкой продукции;
2. ISO/IEC 2501n – раздел "Модель качества". Международные стандарты, входящие в этот раздел, представляют детализированные модели качества систем и программной продукции, качества при использовании и качества данных. Кроме того, приведено практическое руководство по использованию модели качества;
3. ISO/IEC 2502n – раздел "Измерение качества". Международные стандарты, входящие в этот раздел, включают в себя эталонную модель измерения качества программной продукции, математические определения показателей качества и практическое руководство по их использованию. В этом разделе приведены показатели внутреннего и внешнего качества программного обеспечения и показатели качества при использовании. Кроме того, в разделе определены и представлены элементы показателей качества (ЭПК), формирующие основу для вышеперечисленных показателей;
4. ISO/IEC 2503n – раздел "Требования к качеству". Международные стандарты, входящие в этот раздел, определяют требования к качеству на основе моделей и показателей качества. Такие требования к качеству могут быть использованы в процессе формирования требований к качеству программного обеспечения до начала разработки или как входные данные для процесса оценки;
5. ISO/IEC 2504n – раздел "Оценка качества". Международные стандарты, входящие в этот раздел, представляют требования, рекомендации и методические материалы для оценки программной

продукции, выполняемой как оценщиками, так и заказчиками или разработчиками. Кроме того, раздел обеспечивает поддержку документирования измерения как модуля оценки;

6. ISO/IEC 25050-25099 – раздел расширения SQuaRE. Международные стандарты этого раздела в настоящее время включают в себя требования к международным стандартам и техническим отчетам по качеству систем и программной продукции в специальных областях приложения или для дополнения одного или более стандартов серии SQuaRE.

***Идентификация, анализ и детализация объекта изучения, представленного соответствующими процессами системного и программного контекстов Опорной (ссылочной) модели процессов ЖЦ ПО (стандарт ISO/IEC/IEEE 12207:2008 [1])***

Модели качества данного международного стандарта в сочетании с ИСО/МЭК 12207 и ИСО/МЭК 15288 могут использоваться, в частности, для процессов, связанных с определением требований, для верификации и валидации с особым акцентом на спецификации и оценки требований к качеству. В ИСО/МЭК 25030 определено, каким образом модели качества можно использовать для требований к качеству программного обеспечения, а ИСО/МЭК 25040 описывает применение модели качества в процессе оценки качества программного обеспечения.

В сочетании с ИСО/МЭК 15504, который относится к оценке процессов программного обеспечения, настоящий международный стандарт обеспечивает:

- основы определения качества программного продукта в процессах "поставщик-потребитель";
- поддержку анализа, верификации и валидации и основы количественной оценки качества в процессах поддержки;
- поддержку настройки целей качества в процессе управления организацией.

Настоящий стандарт может быть использован в сочетании с ИСО 9001, который посвящен процессам обеспечения качества, для обеспечения:

- поддержки определения цели качества;
- поддержки анализа, верификации и валидации проекта.

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

Данный стандарт содержит требования и рекомендации для организации, ответственной за реализацию и управление действиями по заданию требований и оценке качества систем и программного обеспечения с использованием технологий, инструментариев, опыта и управленческих навыков.

Обязанности группы оценки включают мотивацию специалистов и их обучение для осуществления действий по заданию требований и оценке, подготовке соответствующих документов, определению или разработке необходимых методов и ответов на вопросы по соответствующим технологиям.

Управление технологией связано с планированием и управлением процесса задания и оценки требований к качеству систем и программного обеспечения, измерениями и инструментариями. Сюда относится управление разработкой, приобретением, стандартизацией, контролем, передачей и обратной связью в рамках организации при задании требований и использовании технологического опыта оценки.

Пользователи настоящего стандарта являются ответственными:

- за технологии управления, используемые для задания требований и выполнения оценки;
- определение требований к качеству систем и программного обеспечения;
- поддержку оценки качества систем и программного обеспечения;
- управление организациями, разрабатывающими системы и программное обеспечение;
- функции гарантий качества.

Вместе с тем, все это также применимо к руководителям, вовлеченным в иные системы или программное обеспечение, связанные с осуществляемыми действиями.

ИСО/МЭК 25012 определяет модель качества данных, которая дополняет данную модель.

Область применения моделей качества включает в себя спецификацию поддержки и оценку программного обеспечения и преимущественно программных вычислительных систем с разных точек зрения, которые связаны с их приобретением, требованиями, разработкой, использованием, оценкой, поддержкой, обслуживанием, обеспечением качества и управлением им, а также менеджментом и аудитом. Модели

могут, к примеру, использоваться разработчиками, приобретателями, персоналом обеспечения качества и управления им, а также независимыми оценщиками, в особенности ответственными за спецификацию и оценку качества программного продукта. Деятельность во время разработки продукции, при которой могут быть использованы модели качества, включает в себя:

- определение требований к программному обеспечению и системе;
- подтверждения полноты определения требований;
- определение целей проектирования программного обеспечения и системы;
- определение целей тестирования программного обеспечения и системы;
- идентификацию критериев контроля качества в рамках обеспечения качества;
- определение критериев приемки программного продукта и/или преимущественно программной вычислительной системы;
- установление необходимых для этого показателей характеристик качества

### **Модели качества**

Качество системы – это степень удовлетворения системой заявленных и подразумеваемых потребностей различных заинтересованных сторон, которая позволяет, таким образом, оценить достоинства. Эти заявленные и подразумеваемые потребности представлены в международных стандартах серии SQuaRE посредством моделей качества, которые представляют качество продукта в виде разбивки на классы характеристик, которые в отдельных случаях далее разделяются на подхарактеристики. (Некоторые подхарактеристики разделяются далее на под-подхарактеристики.) Подобная иерархическая декомпозиция обеспечивает удобную разбивку качества продукта на классы. Однако множество подхарактеристик, связанных с характеристикой, избранной для представления типичных проблем, необязательно будет исчерпывающим.

Измеримые, связанные с качеством свойства системы называют свойствами качества, связанными с соответствующими показателями качества. Чтобы прийти к показателям характеристики или подхарактеристики качества в случаях, когда характеристика или подхарактеристика не может быть непосредственно измерена, необходимо



идентифицировать подмножество свойств, которое в совокупности покрывает характеристику или подхарактеристику, получить показатели качества для каждого свойства и, объединив их в вычислительном отношении, достигнуть полученного показателя качества, соответствующего характеристике или подхарактеристике качества.

К настоящему времени в серии SQaRE имеются три модели качества: модель качества при использовании и модель качества продукта, определенные в настоящем стандарте, и модель качества данных, определенная в ИСО/МЭК 25012. Совместное использование моделей качества дает основание считать, что учтены все характеристики качества. Данные модели обеспечивают множество характеристик качества, в которых заинтересован широкий круг лиц, таких как: разработчики программного обеспечения, системные интеграторы, приобретатели, владельцы, специалисты по обслуживанию, подрядчики, профессионалы обеспечения и управления качеством и пользователи.

Не все характеристики качества из полного множества, обеспечиваемого этими моделями, значимы для конкретной заинтересованной стороны. Тем не менее каждая категория заинтересованных лиц должна быть учтена при анализе и рассмотрении важности характеристик качества для каждой модели до завершения формирования набора характеристик качества, которые будут использоваться, чтобы установить, например, требования к производительности продукции и системы или критерии оценки.

Модель качества в использовании по стандарту iso/iec 25010:2011.

**Качество в использовании** – степень применимости продукта или системы заданными пользователями для удовлетворения их потребностей в достижении заданных целей с результативностью, эффективностью, свободой от риска и удовлетворенностью в заданных контекстах использования.

Качество в использовании характеризует влияние, которое продукт оказывает на правообладателей. Оно определяется качеством программного обеспечения, аппаратных средств и эксплуатационной среды, а также характеристиками пользователей, задач и социального окружения.



Рисунок 3 – Качество в использовании

Данная модель состоит из 5 характеристик:

1. **Результативность** – точность и полнота, с которой пользователи достигают заданных целей.
2. **Эффективность** – ресурсы, затрачиваемые в зависимости от точности и полноты, с которыми пользователь достигает целей.
3. **Удовлетворенность** – степень удовлетворения потребностей пользователя при применении продукта или системы в заданном контексте использования. **Подхарактеристики:** применимость, доверие, удовольствие, комфорт.
4. **Свобода от риска** – степень уменьшения продуктом или системой потенциального риска для экономического статуса, человеческой жизни, здоровья или окружающей среды. **Подхарактеристики:** уменьшение экономического риска, уменьшение риска для здоровья и безопасности, уменьшение риска для окружающей среды.
5. **Покрытие контекста** – степень применимости продукта или системы с результативностью, эффективностью, свободой от риска и удовлетворенностью как в заданных контекстах использования, так и вне них. **Подхарактеристики:** покрытия контекста являются полнота контекста, гибкость.

**Модель качества продукта.** Она сводит свойства качества системы/программного продукта к восьми характеристикам, которыми являются: функциональная пригодность, уровень производительности, совместимость, удобство пользования, надежность, защищенность, сопровождаемость и переносимость (мобильность). Каждая характеристика, в свою очередь, состоит из ряда соответствующих подхарактеристик

Примечание. Требования соответствия стандартам или другим нормативам могут быть определены как часть требований к системе, однако они выходят за рамки модели качества.



Рисунок 4 - Качество продукта

Целью модели качества продукта является компьютерная система, в которую входит целевой программный продукт, а цель модели качества при использовании - это совокупная человеко-машинная система, которая включает в себя и целевую компьютерную систему, и целевой программный продукт. В целевую компьютерную систему входят также компьютерное оборудование, нецелевые программные продукты, нецелевые данные и целевые данные, которые, в свою очередь, являются объектом анализа модели качества данных (см. С.8). Целевая компьютерная система является частью информационной системы, в состав которой могут быть также включены одна или более компьютерных систем и системы связи, такие как локальная сеть и Интернет. В состав информационной системы в более крупной человеко-машинной системе (такой как корпоративная система, встроенная система или крупномасштабная система управления) могут входить пользователи, техническая и физическая среда использования. Рамки целевой системы

определяются исходя из области применения требований или оценки и из того, кто рассматривается в качестве пользователей.

***Пример: Если в качестве пользователей самолета с компьютерной системой управления полетом рассматривать пассажиров, то система, от которой они зависят, включает летный экипаж, сам самолет, аппаратное и программное обеспечение системы управления полетом. В случае, если в качестве пользователей рассматривать летный экипаж, то система, от которой они зависят, состоит только из самого самолета и системы управления полетом.***

С качеством также связаны и другие заинтересованные стороны, такие как разработчики программного обеспечения, системные интеграторы, приобретатели, владельцы, специалисты по обслуживанию, подрядчики, профессионалы обеспечения и управления качеством.

Модели качества продукции и качества при использовании могут быть использованы для определения требований, выработки показателей и выполнения оценки качества (см. приложение С). Определенные характеристики качества могут использоваться в качестве контрольного списка для обеспечения детального исследования требований к качеству, обеспечивая таким образом основу для оценки необходимых в процессе разработки систем последующих трудозатрат и действий. Характеристики в модели качества при использовании и модели качества продукта предназначены для использования в качестве набора при спецификации или оценке качества программного продукта или компьютерной системы.

Практически невозможно определить или измерить все подхарактеристики для всех частей большой компьютерной системы или программного продукта. Аналогично в большинстве случаев практически не применимо определение или измерение качества при использовании для всех возможных сценариев задач пользователя. Относительная важность характеристик качества зависит от целей высокого уровня и целей проекта. В связи с этим перед использованием для выделения из требований тех характеристик и подхарактеристик, которые наиболее важны, модель должна быть соответствующим образом адаптирована, а ресурсы распределены между различными типами показателей в зависимости от целей заинтересованных лиц и целей продукта.

## Качество с точки зрения различных заинтересованных сторон

Модели качества обеспечивают основу для сбора требований заинтересованных сторон. Заинтересованная сторона - это следующие три типа пользователя:

1. Основной пользователь - лицо, взаимодействующее с системой для достижения основных целей.
2. Вторичные пользователи - лица, осуществляющие поддержку, например:
  - а) провайдер контента, системные инженер/администратор, руководитель безопасности;
  - б) специалист по обслуживанию, анализатор, специалист по портированию, установщик.
3. Косвенный пользователь - лицо, которое получает результаты, но не взаимодействует с системой.

У каждого из данных типов пользователя есть свои требования к качеству использования и качеству продукта в конкретных условиях использования, как это показано в таблице 1 для отдельных примеров пользователей и характеристик.

Примечание. Кроме того, провайдер контента имеет свои пользовательские требования к качеству данных.

Требования пользователя, перечисленные в таблице 1, являются примерами отправных точек для формирования требований и могут быть использованы в качестве основы для определения влияния на качество системы при использовании и обслуживании.

Требования к качеству должны быть определены с точки зрения заинтересованных лиц до разработки или приобретения программного обеспечения. Результатом анализа требований к использованию будут определенные требования к функциональности и качеству продукции, необходимые для достижения требований к использованию.

***Пример: Общие требования к надежности системы могут привести к конкретным требованиям к завершенности, готовности, отказоустойчивости и восстанавливаемости программного продукта. Надежность может также повлиять на общую эффективность, производительность, свободу от риска и удовлетворенность.***

***Усовершенствование подхода на основе Kiviat – diagrams. Аппарат радиально – метрических диаграмм. Иерархия. Свертка.***

Лепестковая диаграмма - диаграмма или график, состоящий из осей, называемых радиусами, каждая из них представляет какую-то переменную. Размер значения пропорционально величине переменной в каждой точке. Таким образом образуется линия, соединяющая значение переменной в каждой точке - внешний вид диаграммы становится похожим на звезду, от чего и происходит одно из самых распространенных названий этой диаграммы.

Она может быть использована для нахождения ответа на следующие вопросы:

- Какие наблюдения ближайшие по значению?
- Встречаются прыжки значение?

Лепестковые диаграммы – полезный способ отображения многомерных наблюдений с произвольным количеством переменных. Каждая диаграмма изображает одно наблюдение. Как правило, такие диаграммы нарисованы на одном рисунке различными линиями, каждая из которых представляет определенное исследование.

Лепестковую диаграмму можно использовать для оценки качества ПО.

***Пример: n радиусов, выходящих из центра круга, соответствуют n ключевым показателям, по которым мы хотим оценить, например, работу организации. Концентрические окружности — 1, 2, 3, 4, 5 — задают систему оценок по пятибалльной шкале. Соединяя оценки на различных радиусах, мы сразу видим, по каким показателям у нас все хорошо, а по каким предстоит много работы.***

**Аппарат радиальных метрических диаграмм.** РМД задает  $n$ -мерное метрическое пространство, в котором осуществляется оценивание объекта, где  $n$  – число метрик  $p_i$  от 1 до  $n$  отображающихся в виде лучей диаграммы

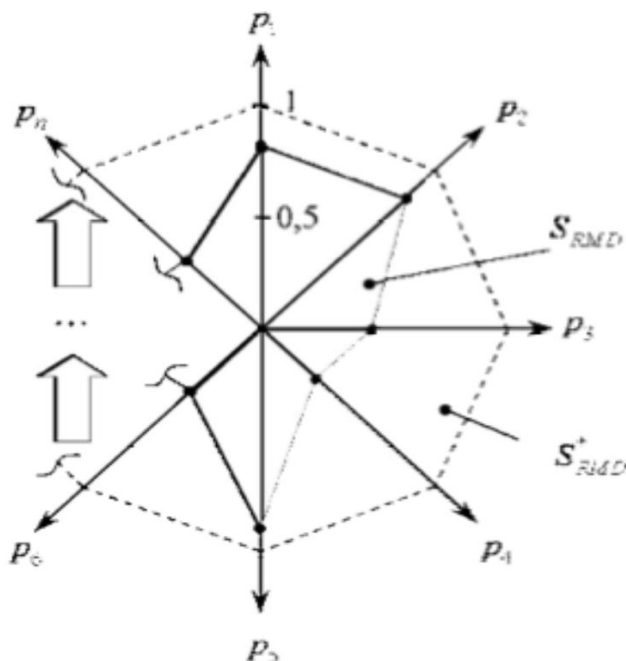


Рис. 1. Общий вид РМД

Использование аналогично лепестковой диаграмме.

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 весомости (значимости) формируется обобщенный показатель, значение которого затем откладывается на соответствующем луче комплексной РМД верхнего уровня.

## Выводы

SQuaRE – это требования к качеству и оценке систем и программного обеспечения. Качество программного обеспечения является способностью программного продукта при заданных условиях

удовлетворять установленным или предполагаемым потребностям (ISO/IEC 25000:2014).

Пользователи программного обеспечения испытывают потребности в создании моделей качества программного обеспечения для оценки качества как качественно, так и количественно. Модели качества, которые имеются в настоящее время, в большинстве случаев являются иерархическими моделями на основе критериев качества и связанных с ними показателей (метрик).

Стандарт ISO/IEC 25010:2011 (ГОСТ Р ИСО/МЭК 25010-2015) определяет модель качества продукта, которая включает восемь характеристик верхнего уровня: функциональная пригодность, уровень производительности, совместимость, удобство использования, надёжность, защищённость, сопровождаемость, мобильность, функциональная пригодность. С другой стороны, качественный программный продукт не должен выполнять никаких необъявленных действий с данными (такие действия в рамках конкретной системы могут рассматриваться просто как порча информации).

В этом стандарте модель качества продукта рассматривается отдельно от субъективного качества в использовании. Модель качества в использовании включает следующие характеристики верхнего уровня: результативность, производительность, удовлетворённость, свобода от риска, покрытие контекста.

Приведенный комплексный анализ поможет понять необходимость использования при оценке качества тех или иных характеристик и подхарактеристик качества программного обеспечения. Необходимо подчеркнуть, что для всесторонней оценки качества можно использовать параллельно разные модели.



## Список литературы

[1]	ИСО/МЭК 25022, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of quality in use (revision of ИСО/МЭК 9126-4) (to be proposed) (ИСО/МЭК 25022 Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQUARE). Измерение качества при использовании (пересмотр ИСО/МЭК 9126-4))
[2]	ИСО/МЭК 25023, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of system and software product quality (revision of ИСО/МЭК 9126-2 and 9126-3) (to be proposed) (ИСО/МЭК 25023 Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQUARE). Измерение качества систем и программной продукции (пересмотр ИСО/МЭК 9126-2 и 9126-3))
[3]	ИСО/МЭК 25024, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of data quality (to be proposed) (ИСО/МЭК 25024 Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQUARE). Измерение качества данных)
[4]	ИСО/МЭК 25042, Systems and software engineering - Systems and software product Quality Requirements and Evaluation (SQuaRE) - Evaluation modules (to be proposed) (ИСО/МЭК 25042 Системная и программная инженерия. Требования и оценка качества систем и программного продукта (SQUARE). Модули оценки)
[5]	ИСО/МЭК 25012:2008, Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Data quality model (ИСО/МЭК 25012:2008 Программная инженерия. Требования и

	оценка качества программной продукции (SQUARE). Модель качества данных)
[6]	ТО ИСО/МЭК 9126-2, Software engineering - Product quality - Part 2: External metrics [Technical Report] (ТО ИСО/МЭК 9126-2 Программная инженерия. Качество продукции. Часть 2. Внешние показатели [Технический отчет])
[7]	ТО ИСО/МЭК 9126-3, Software engineering - Product quality - Part 3: Internal metrics [Technical Report] (ТО ИСО/МЭК 9126-3 Программная инженерия. Качество продукции. Часть 3. Внутренние показатели [Технический отчет])
[8]	ТО ИСО/МЭК 9126-4, Software engineering - Product quality - Part 4: Quality in use metrics [Technical Report] (ТО ИСО/МЭК 9126-4 Программная инженерия. Качество продукции. Часть 4. Показатели качества при использовании [Технический отчет])
[10]	ИСО/МЭК 24765:2010, Systems and software engineering vocabulary (ИСО/МЭК 24765:2010 Системная и программная инженерия. Словарь)
[11]	ИСО/МЭК 12207:2008, Systems and Software Engineering - Software Life Cycle Processes (ИСО/МЭК 12207:2008 Системная и программная инженерия. Процессы жизненного цикла программного обеспечения)
[12]	ИСО/МЭК 15288:2008, Systems and Software Engineering - System Life Cycle Processes (ИСО/МЭК 15288:2008 Системная и программная инженерия. Процессы жизненного цикла систем)

[13]	<p>ИСО/МЭК 14143-1:2007, Information technology - Software measurement - Functional size Measurement (ИСО/МЭК 14143-1:2007 Информационные технологии. Измерение программного обеспечения. Измерение функционального размера)</p>
[14]	<p>IEEE 1061-1998 (R2004), IEEE Standard for Software Quality Metrics Methodology (IEEE 1061-1998 (R2004) Стандарт IEEE по Методологии показателей качества программного обеспечения)</p>
[15]	<p>IEEE 1012-2004, IEEE Standard for Software Verification and Validation (IEEE 1012-2004 Стандарт IEEE по верификации и валидации программного обеспечения)</p>