

МОДЕЛИ, МЕТОДЫ, ПОКАЗАТЕЛИ, ХАРАКТЕРИСТИКИ И МЕТРИКИ, ПРИМЕНЯЕМЫЕ В ЭКСПЕРТНЫХ СИСТЕМАХ ОЦЕНКИ КАЧЕСТВА РАЗРАБОТКИ И СОЗДАНИЯ ИННОВАЦИОННЫХ ПРОГРАММНЫХ ПРОЕКТОВ***Сергей Николаевич ЛАРИН^а, Лариса Юрьевна ЛАЗАРЕВА^б, Татьяна Сергеевна ЛАРИНА^с**^а кандидат технических наук, старший научный сотрудник,
Центральный экономико-математический институт Российской академии наук, Москва, Российская Федерация
sergey77707@rambler.ru^б кандидат экономических наук, главный специалист,
Институт международных стандартов учета и управления, Москва, Российская Федерация
lazarlu@rambler.ru^с научный сотрудник, Институт международных стандартов учета и управления, Москва, Российская Федерация
lartaser@rambler.ru

• Ответственный автор

История статьи:Получена 11.01.2017
Получена в доработанном виде
25.01.2017
Одобрена 13.02.2017
Доступна онлайн 15.06.2017

УДК 332.1:334.0

JEL: D71, D72, O11, P25, Q01

<https://doi.org/10.24891/re.15.6.1187>**Ключевые слова:**

инновационные программные проекты, оценка качества, экспертные системы, модели, методы

Аннотация**Предмет.** Существующие экспертные системы оценки качества разработки инновационных программных проектов, а также модели и методы, применяемые для их оценки. В условиях стремительного роста объемов информации возникает объективная необходимость разработки и практического использования инновационных программных проектов для планирования и управления деятельностью предприятий. Поэтому становится актуальной оценка качества их разработки на основе существующих экспертных систем.**Цели.** Проведение сравнительного анализа существующих моделей и методов, используемых для экспертной оценки качества разработки инновационных программных проектов, а также определение состава показателей, характеристик и метрик для их количественного измерения в целях формирования экспертных систем оценки.**Методология.** В процессе проведения исследования были использованы методы сравнительного статистического анализа, декомпозиции показателей, а также методы экспертных оценок, метод анализа иерархий и основы методологии нечетких множеств.**Результаты.** Выявлен ряд существующих моделей, используемых для оценки качества современных инновационных программных проектов. Определен состав показателей, их качественных характеристик и подхарактеристик, на основе которых можно формировать экспертные системы для оценки их разработки.**Выводы.** Для получения комплексной оценки качества современных инновационных программных проектов целесообразно использовать различные модели одновременно. При этом для оценки качества каждого конкретного инновационного программного проекта следует формировать индивидуальную экспертную систему показателей, характеристик и подхарактеристик его качественных свойств, а также метрик для их измерения.

© Издательский дом ФИНАНСЫ и КРЕДИТ, 2017

Введение

В связи с развитием информационных технологий постоянно возрастает потребность в разработке программных проектов самого разного назначения, в том числе и инновационных программных проектов (ИПП) для стратегического планирования и управления деятельностью предприятий. Стремительный рост объемов информации предопределяет сложность процесса разработки такого рода продуктов, которая во многом обуславливается особыми требованиями, предъявляемыми к их качеству. Это

обстоятельство повышает важность использования формализованных методов для определения качества разработки программных проектов вообще и современных инновационных программных проектов в частности.

Для любой информационной системы (ИС), использующей программное обеспечение (ПО), обычно разрабатываются три вида спецификаций, содержащих требования к ее функциональным свойствам, качественным и ресурсным характеристикам. При этом современные инновационные программные проекты вполне могут рассматриваться как сложные информационные системы. Поэтому, по мнению И.М. Горбаченко [1], в состав качественных характеристик инновационных программных проектов включаются все существенные

* Статья подготовлена при финансовой поддержке Российского фонда фундаментальных исследований, проект № 15-06-00044а «Развитие методологии экспертизы результатов поисковых научных исследований для проектов создания инновационной продукции, услуг и технологий: экономические методы, модели, инструментальный и алгоритмы обработки информации».

показатели, обеспечивающие их функционирование и позволяющие экспертам получить оценку степени их соответствия заданным спецификациями требованиям. Таким образом, качество инновационных программных проектов является обобщением их функциональных характеристик и показателей, которые используются экспертами для оценки его способности удовлетворять заданным спецификациями требованиям.

Набор характеристик и отношений между ними, которые фактически обеспечивают основу как для определения соответствия установленных в спецификациях требований показателей качества их реальным значениям, так и оценки качества программного обеспечения, обычно определяется в моделях качества¹. В ряде работ зарубежных специалистов (А. Seffah и Н.К. Padda [2], а также R. Fitzpatrick [3]), есть подтверждения того, что модели качества используются для структурирования и декомпозиции указанных наборов свойств на ряд дополнительных характеристик/подхарактеристик, которые необходимы для оценки экспертами уровня достижения инновационных программных проектов заданных целей их функционирования.

Обзор современных моделей качества

Существующие модели качества в большинстве случаев являются иерархическими моделями на основе критериев качества и связанных с ними показателей (метрик). Все модели могут быть разделены на три вида в соответствии с методами, на основе которых они были созданы:

- к первому виду можно отнести теоретические модели, основанные на гипотезе отношений между переменными качества;
- ко второму виду относятся модели «управления данными», основанные на статистическом анализе;
- к третьему виду относится комбинированная модель, в которой интуиция исследователя используется для определения нужного вида модели, а анализ данных применяется для определения констант модели качества.

Первая модель для оценки качества программного обеспечения была предложена в работах таких исследователей, как J.A. McCall, P.K. Richards, G.F. Walters [4–6]. Ее основное предназначение заключалось в получении полной оценки качества программного обеспечения через его различные

характеристики. В модели МакКолла имеются три основные характеристики для оценки качества программного обеспечения:

- использование, метриками для которой являются корректность, надежность, эффективность, целостность и практичность;
- модификация, метриками для которой являются тестируемость, гибкость, а также сопровождаемость, подразумевающая наличие важных факторов качества для разработки новой версии программного обеспечения;
- переносимость, метриками для которой являются мобильность, возможность многократного использования, функциональная совместимость, то есть те факторы качества, которые важны для переноса программного обеспечения на другие аппаратные и программные платформы.

Второй основополагающей моделью оценки качества программного обеспечения является модель Б. Боэма [7], которая позволяет более точно определять качество его основных характеристик, заданных набором показателей и метрик. Модель качества Б. Боэма представляет характеристики программного обеспечения в более крупном масштабе, чем модель МакКолла, поскольку в ее основе лежит оценка качества программного обеспечения на всех этапах его жизненного цикла. Модель Б. Боэма сходна с моделью качества МакКолла, поскольку она также относится к иерархическим моделям качества, структурирование показателей в которых осуществляется сначала на основе высокоуровневых, затем промежуточных и, наконец, индивидуальных характеристик. При этом каждая из них вносит свой вклад в формирование оценки качества программного обеспечения. В этой модели практичность показывает, насколько легко, надежно и эффективно программное обеспечение может быть использовано для решения конкретных задач; сопровождаемость характеризует наличие возможности изменить и повторно протестировать программное обеспечение; мобильность определяет возможности использования программного обеспечения на различных программных средствах и аппаратных платформах.

Однако модель Б. Боэма не лишена и недостатков, свойственных многим современным моделям, которые оценивают качество программного обеспечения автоматически, то есть не всегда точно.

¹ ISO/IEC 9126-1:2001. Software Engineering – Software Product Quality – Part 1: Quality Model.

По аналогии с моделями МакКолла и Б. Боэма для оценки качества программного обеспечения R.B. Grady и D.L. Caswell предложили использовать *FURPS* модель [8]. Ее отличие от предыдущих моделей заключается в наличии двух слоев показателей качества. При этом первый слой определяет только основные качественные характеристики показателей, а во втором слое представлены связанные с ними атрибуты. Современной модификацией этой модели стала модель *FURPS+*.

Модель *FURPS/FURPS+* получила свое название по первым буквам основных категорий показателей качества программного обеспечения:

- **Functionality** – функциональность, которая включает в себя в качестве дополнительных показателей особенности, возможности и безопасность используемого программного обеспечения в инновационных программных проектах;
- **Usability** – практичность, которая включает в себя в качестве дополнительных показателей возможность учета человеческого фактора, эргономичность и наличие полного комплекта пользовательской документации (инструкции, правила инсталляции и деинсталляции и др.);
- **Reliability** – надежность, которая включает в себя в качестве дополнительных показателей частоту отказов в работе программного обеспечения, наличие возможностей для восстановления информации, а также прогнозируемость действий пользователя в ближайшей перспективе;
- **Performance** – производительность, которая включает в себя в качестве дополнительных показателей время отклика, пропускную способность обработки определенных объемов информации, точность получаемых решений, простоту и доступность программного обеспечения для пользователей, использование всех возможностей информационных ресурсов обработки данных;
- **Supportability** – эксплуатационная пригодность, которая включает в себя в качестве дополнительных показателей возможности тестирования программного обеспечения и его отдельных компонентов (модулей), его способности к расширению до более совершенных версий, уровень адаптируемости используемого программного обеспечения, необходимость организации сопровождения в процессе эксплуатации, возможности совместимости используемого программного

обеспечения с другими информационными программами и платформами, возможности изменения конфигурации, простоту обслуживания в процессе эксплуатации, стандартные требования к установке, а также возможности локального функционирования.

Символ «+» обеспечивает расширение *FURPS* модели за счет добавления к ней:

- ограничений проекта по информационным ресурсам, требованиям к языкам и средствам разработки, требований к аппаратному обеспечению;
- ограничений интерфейса инновационных программных проектов, накладываемых на взаимодействие с внешними системами;
- функциональных требований к выполнению заявленного в спецификации набора операций;
- соответствия эргономическим требованиям, обеспечения выполнения необходимых требований по лицензированию информационного проекта.

Концептуальной основой модели качества *FURPS/FURPS+* является декомпозиция характеристик программного обеспечения в разрезе двух категорий:

- функциональной *F*;
- нефункциональной *URPS*.

Такой подход к разработке указанных моделей позволяет использовать эти категории в виде требований для оценки качества программного обеспечения, а также в виде основных показателей при оценке качества инновационных программных проектов. В настоящее время целесообразность использования *FURPS+* модели в ходе разработки программного обеспечения и при идентификации требований к разрабатываемым инновационным программным проектам определяется наличием достаточно полного представления в ней наиболее универсального перечня характеристик для оценки качества программного обеспечения и разрабатываемого на основе его использования инновационного программного проекта.

В модели К. Гецци и его соавторов используются различные подходы к определению качества программного обеспечения и процесса его эксплуатации в составе инновационного программного проекта [9]. В соответствии с этой моделью качество программного обеспечения определяется следующими показателями:

- целостностью;

- надежностью;
- устойчивостью;
- производительностью;
- верифицируемостью;
- сопровождаемостью;
- практичностью;
- возможностью многократного использования;
- мобильностью;
- понятностью;
- возможностью взаимодействия;
- эффективностью;
- своевременностью реакции на действия пользователя.

В основу модели *G.R. Dromey* [10] положены критерии оценки характеристик качества и их подхарактеристик. Главное предназначение этой модели заключается в оценке качества инновационного программного проекта в целом как информационной системы с учетом того обстоятельства, что оценка качества каждого такого программного проекта в силу известных причин будет отличаться от оценок качества других инновационных программных проектов. Данную модель целесообразно использовать для выявления возможного наличия ошибок в программном обеспечении и поиска тех его составляющих, в которых потенциально возможно появление ошибок. Используя отношения между характеристиками и подхарактеристиками качества, эта модель позволяет получить оценки качества отдельных свойств и характеристик как программного обеспечения и его составляющих, так и инновационного программного проекта в целом.

Большой интерес для оценки качества программного обеспечения и функционирующих на его основе информационных систем представляет разработанная в Центре обеспечения качества программного обеспечения NASA модель метрик *SATC* (Software Assurance Technology Center). Принципиальное отличие этой модели от всех рассмотренных ранее заключается в том, что в ней первоначально оценки качества определяются для каждой составляющей инновационного программного проекта в отдельности. При этом сначала оценивается качество разработки требований спецификаций, программного обеспечения и его составляющих,

документации на информационный проект, тестирования его составляющих, а также выполнение операций. Далее на основе полученных оценок формируется интегральный показатель качества разработанного инновационного программного проекта. Благодаря такому подходу эта модель обеспечивает не только оценку рисков информационного проекта, но и эффективность выполняемых операций, а значит и качество разработанного инновационного программного проекта. Для этого в модели качества *SATC* формируется набор целей, которые связаны с инновационным программным проектом и атрибутами выполняемых им операций в соответствии со структурой модели качества программного обеспечения, разработанной в рамках стандарта ISO 9126. Более детально основные положения этой модели изложены в работе [11].

Модель качества стандарта ISO 9126 определяет качество программного обеспечения как некую совокупность его характеристик, обеспечивающих решение разного набора производственных задач лицами, принимающими управленческие решения. Подобное определение качества программного обеспечения, а также разнообразные наборы решаемых на производстве с его использованием задач представлены в работах таких исследователей, как С. Stein, G. Cox и L. Etzkorn [12], N.E. Fenton и M. Neil [13], P.A. Laplante [14].

Модель качества стандарта ISO 9126 различает понятия:

- внутреннего качества, связанного с характеристиками самого программного обеспечения и его составляющих без учета их функционирования в рамках инновационного программного проекта (ISO/IEC 9126-1:2001²);
- внешнего качества, характеризующего программное обеспечение с точки зрения его функционирования на различных информационных платформах (ISO/IEC TR 9126-2:2003³);
- качества программного обеспечения с позиции его использования конкретными пользователями при различных сценариях их работы (ISO/IEC TR 9126-3:2003⁴).

Кроме того, для оценки надежности используемого программного обеспечения важным

² ISO/IEC 9126-1:2001. Software engineering – Software product quality – Part 1: Quality model.

³ ISO/IEC TR 9126-2:2003 Software engineering – Product quality – Part 2: External metrics.

⁴ ISO/IEC TR 9126-3:2003 Software Engineering – Product Quality – Part 3: Internal Metrics.

представляется качество технологических процессов его разработки (ISO/IEC TR 9126-4:2004⁵). Оценка качества всех указанных ранее характеристик осуществляется на основе введения специальных метрик, позволяющих получить их расчетные значения.

Основными показателями и их атрибутами (характеристиками и подхарактеристиками) внешнего и внутреннего качества программного обеспечения в рамках модели качества стандарта ISO 9126 являются:

1) функциональность, к атрибутам которой относятся:

- способность к взаимодействию с приложениями других информационных систем;
- функциональная пригодность использования в конкретной области знаний;
- соответствие стандартам качества функциональности;
- обеспечение согласованности получаемых результатов;
- требуемый уровень защищенности;

2) надежность, к атрибутам которой относятся:

- стабильность работы на любых информационных платформах;
- возможность оперативного восстановления утраченных данных;
- соответствие стандартам качества надежности;
- наличие устойчивости к отказам программного обеспечения;

3) удобство использования, к атрибутам которого относятся:

- соответствие стандартам качества;
- простота и удобство обучения;
- понятность и привлекательность для пользователя;
- эргономичность работы пользователя;

4) эффективность, к атрибутам которой относятся:

- соответствие стандартам качества;
- соответствие заданной эффективности работы программного обеспечения;

- эффективное использование всех информационных ресурсов для выполнения операций и решения задач;

5) сопровождаемость, к атрибутам которой относятся:

- возможность внесения изменений в программное обеспечение;
- соответствие стандартам качества;
- возможности анализа выполненных операций и тестирования программного обеспечения на наличие ошибок;
- наличие устойчивости к отказам программного обеспечения;

6) мобильность, к атрибутам которой относятся:

- адаптируемость к любым информационным платформам;
- простота внедрения программного обеспечения;
- способность к одновременной работе с приложениями других информационных продуктов;
- простота и удобство установки программного обеспечения.

Графическая интерпретация алгоритма функционирования модели оценки качества программного обеспечения в рамках стандарта ISO 9126 представлена на *рис. 1*.

Иерархическую модель оценки качества объектно-ориентированного проектирования информационных проектов (*QMOOD*) предложили J. Bansiya и C. Davis [15]. Эта модель существенно расширяет методологию оценки качества, используемую в модели G.R. Dromey. Она основывается на четырех уровнях оценки качества инновационного программного проекта:

1) на показателях качества, которые представлены в виде набора качественных атрибутов проекта и используются для описания его основных характеристик:

- функциональности;
- эффективности;
- понятности;
- расширяемости;
- возможности многократного использования;
- гибкости;

⁵ ISO/IEC TR 9126-4:2004 Software Engineering – Product Quality – Part 4: Quality In Use Metrics.

2) на объектно ориентированных свойствах инновационного программного проекта, которые могут быть определены в ходе оценки качества его структуры (внутренней и внешней), функциональности отдельных составляющих и атрибутов. В *QMOOD* в качестве структурного объектно ориентированного множества обычно используются:

- размер и иерархическая структура;
- связанность составных частей;
- полиморфизм;
- сложность обмена информацией;

3) на объектно ориентированных метриках инновационного программного проекта, которые предназначены для оценки его качества;

4) на объектно ориентированных свойствах инновационного программного проекта, которые используются для формирования его архитектуры.

Модель *QMOOD* представляет несколько иной подход к оценке качества разработанных инновационных программных проектов, который основывается на ряде новых объектно-ориентированных метрик, вводимых для его измерения.

В 2003 г. L. Bass, P. Clements, R. Kazman представили модель оценки инновационных программных проектов, основанную на двух различных подходах к выбору показателей для оценки его качества, учитывая длительность всего жизненного цикла используемого в нем программного обеспечения [16]. При этом авторы модели выделили две группы основных характеристик качества:

- 1) эффективность, безопасность, доступность и функциональность;
- 2) модифицируемость, мобильность, возможность многократного использования, наследуемость и тестируемость.

В 2005 г. исследователи K. Khosravi и Y. Gueheneuc предложили модель оценки качества программного обеспечения, основанную на многократном использовании одной из глобальных характеристик качества [17]. Кроме того, она допускает возможность многократного использования таких характеристик, как понятность, гибкость, модульность, надежность, масштабируемость и удобство использования.

Оценка качества в этой модели заключается в последовательном решении двух задач:

- 1) в выборе глобальной характеристики;
- 2) в выборе подхарактеристик, связанных с глобальной характеристикой.

В этой модели для связи показателей качества и их подхарактеристик использовались базовые определения стандартов IEEE, ISO/IEC и ряда других моделей оценки качества.

В 2008 г. исследователи C. Chang, C. Wu, H. Lin предложили подход для оценки качества программного обеспечения на основе использования методологии нечетких множеств и метода анализа иерархий [18]. Основные принципы этого подхода были ими реализованы в рамках модели оценки качества стандарта ISO 9126. В частности, ими использовались характеристики и подхарактеристики основных показателей модели оценки качества программного обеспечения стандарта ISO 9126.

В том же году ученые A. Sharma, R. Kumar и P.S. Grover [19] предложили компонентно ориентированную модель для оценки качества программного обеспечения, в которую были включены не только все характеристики и подхарактеристики модели оценки качества стандарта ISO 9126, но и предложен целый ряд новых подхарактеристик, к которым можно отнести пригодность к повторному использованию, гибкость, сложность, масштабируемость. Для оценки качества инновационных программных проектов в этой модели предложено использовать хорошо известный метод анализа иерархий.

Наконец, в 2011 г. вместо стандарта ISO/IEC 9126-1:2001. Software Engineering – Software Product Quality – Part 1: Quality Model был принят стандарт ISO/IEC 25010:2011 Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models⁶. Его аналогом в Российской Федерации стал ГОСТ Р ИСО/МЭК 25010-2015⁷.

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

- 1) модель качества в использовании, состоящая из пяти характеристик, имеющих отношение

⁶ ISO/IEC 25010:2011. Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models.

⁷ ГОСТ Р ИСО/МЭК 25010-2015. Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов.

к результатам взаимодействий продукта при его применении в заданном контексте использования. Данная модель представляет собой модель системы, применимую к полным человеко-компьютерным системам, включая как используемые информационные системы, так и используемое программное обеспечение;

- 2) модель качества информационного проекта, состоящая из восьми характеристик, имеющих отношение к статическим свойствам программного обеспечения и динамическим свойствам информационной системы. Данная модель применима как к информационным системам, так и к программному обеспечению⁸.

Качество в использовании характеризует влияние, которое информационный продукт (система или программа) оказывает на правообладателей. По мнению Е.Е. Исмаила [20], оно определяется качеством программного обеспечения, аппаратных средств и эксплуатационной среды, а также компетентностными характеристиками пользователей задач и их социальным окружением.

Модель качества информационного проекта подразделяет свойства качества информационной системы или инновационного программного проекта на восемь характеристик:

- функциональное соответствие;
- эффективность функционирования;
- совместимость;
- практичность;
- надежность;
- защищенность;
- сопровождаемость;
- мобильность.

Далее по ряду характеристик рассмотрим отличия данной модели от модели внутреннего и внешнего качества, определенной стандартом ISO/IEC 9126–1:2001, которые в тексте выделены курсивом.

Функциональное соответствие (Functional suitability) – степень обеспечения программным продуктом или системой функций, отвечающих заданным или подразумеваемым потребностям при использовании в заданных условиях.

Подхарактеристиками функционального соответствия являются *функциональная полнота (functional completeness)*, *функциональная правильность (functional correctness)*, *функциональная пригодность (functional appropriateness)*.

Эффективность функционирования (Performance efficiency) – зависимость функционирования от количества ресурсов, используемых в заданных условиях. Ресурсы могут включать другие программные продукты, программную и аппаратную конфигурацию системы, материалы.

Подхарактеристиками эффективности функционирования являются *поведение во времени (time behaviour)*, *использование ресурсов (resource utilization)*, *емкость (capacity)*.

Совместимость (Compatibility) – 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.

Подхарактеристиками совместимости являются *сосуществование (coexistence)* и *способность к взаимодействию (interoperability)*.

Практичность (Usability) – степень применимости программного продукта или системы заданными пользователями для достижения заданных целей с результативностью, эффективностью и удовлетворенностью в заданном контексте использования.

Подхарактеристиками практичности являются *распознаваемость пригодности (appropriateness recognizability)*, *обучаемость (learnability)*, *простота использования (operability)*, *защита от ошибок пользователей (user error protection)*, *эстетичность пользовательского интерфейса (user interface aesthetics)*, *доступность (accessibility)*.

Надежность (Reliability) – степень выполнения системой, программным продуктом или компонентом заданных функций в заданных условиях в течение заданного периода времени. Для программных средств ограничения надежности связаны с ошибками в требованиях, проекте и реализации или с изменениями контекста.

Подхарактеристиками надежности являются *завершенность (maturity)*, *готовность (availability)*, *устойчивость к ошибке (fault tolerance)*, *восстанавливаемость (recoverability)*.

⁸ ISO/IEC 25010:2011. Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models.

Защищенность (Security) – 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.

Подхарактеристиками защищенности являются *конфиденциальность (confidentiality)*, *целостность (integrity)*, *неопровержимость (non-repudiation)*, *идентифицируемость (accountability)*, *аутентичность (authenticity)*.

Сопровождаемость (Maintainability) – степень результативности и продуктивности модификаций программного продукта или системы запланированным персоналом сопровождения. Модификации могут включать исправления, улучшения или адаптацию программного средства к изменениям в окружающей среде, требованиях и функциональных спецификациях.

Подхарактеристиками сопровождаемости являются *модульность (modularity)*, *повторная используемость (reusability)*, *анализируемость (analysability)*, *модифицируемость (modifiability)*, *тестируемость (testability)*.

Мобильность (Portability) – степень результативности и эффективности переноса системы, программного продукта или компонента из одной аппаратной, программной или иной эксплуатационной или используемой среды в другую.

Подхарактеристиками мобильности являются *адаптируемость (adaptability)*, *простота установки (installability)*, *взаимозаменяемость (replaceability)*.

Эти характеристики и подхарактеристики дают возможность свободной абстракции по широкому кругу вопросов оценки качества. Это важно по двум взаимодополняющим направлениям:

- во-первых, путем принятия решения по качественным характеристикам, которые являются более важными для успешного функционирования инновационного программного проекта, что позволяет

сосредоточиться на множестве вопросов, связанных с решением реальных задач;

- во-вторых, поскольку эта категоризация носит комплексный характер, она снижает риск неверного решения ряда важных вопросов в связи с их взаимной зависимостью.

В целом стандарт ISO/IEC 25010:2011 достаточно подробно описывает модель оценки качества программного обеспечения с использованием большого количества характеристик и подхарактеристик показателей качества. При этом каждая характеристика или подхарактеристика описывает степень качества какого-либо показателя настолько подробно, что делает возможным количественное описание конкретной характеристики. Такой подход существенно облегчает оценку качества инновационного программного продукта, а сами характеристики достаточно полно охватывают его качественное описание, делая возможным оценку качества программных продуктов любой сложности.

Заключение

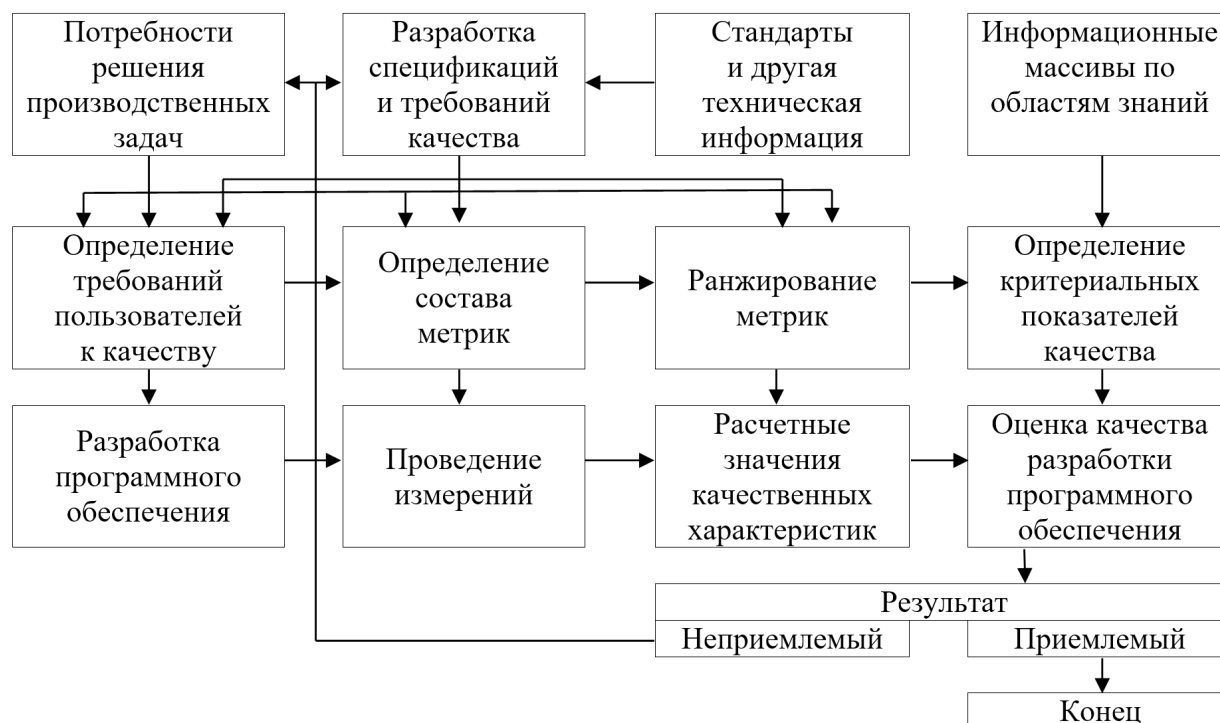
В статье был представлен краткий обзор и сравнительный анализ основных моделей качества, используемых для оценки программного обеспечения. Его результаты позволяют выявить объективную необходимость использования для оценки качества инновационных программных продуктов различных совокупностей характеристик и подхарактеристик свойств программного обеспечения, используемого для функционирования конкретных программных продуктов. Следует отметить, что для получения комплексной оценки качества инновационных программных продуктов целесообразно одновременно использовать разные модели. При этом в случае необходимости получения комплексной оценки качества конкретного инновационного программного продукта должна формироваться индивидуальная экспертная система показателей, характеристик и подхарактеристик его качественных свойств, а также метрик для их измерения.

Рисунок 1

Алгоритм функционирования модели оценки качества программного обеспечения

Figure 1

The algorithm of functioning of software quality assessment model



Источник: ISO/IEC 9126-1:2001. Software Engineering – Software Product Quality – Part 1: Quality Model

Source: ISO/IEC 9126-1:2001. Software Engineering – Software Product Quality – Part 1: Quality Model

Список литературы

1. Горбаченко И.М. Оценка качества программного обеспечения для создания систем тестирования // Фундаментальные исследования. 2013. № 6-4. С. 823–827.
2. Seffah A., Donyaee M., Kline R.B., Padda H.K. Usability Measurement and Metrics: A consolidated model. *Software Quality Journal*, 2006, vol. 14, iss. 2, pp. 159–178. doi: 10.1007/s11219-006-7600-8
3. Fitzpatrick R. Software Quality: Definitions and Strategic Issues. Staffordshire University, School of Computing Report, 1996, 34 p. Available at: <https://www.saylor.org/site/wp-content/uploads/2012/11/Software-quality-definitions-and-strategic-issues.pdf>
4. McCall J.A., Richards P.K., Walters G.F. Factors in Software Quality: Concept and Definitions of Software Quality. Final Technical Report. Vol. 1. National Technical Information Service, Springfield, 1977. Available at: <http://www.dtic.mil/dtic/tr/fulltext/u2/a049014.pdf>
5. McCall J.A., Richards P.K., Walters G.F. Factors in Software Quality: Metric Data Collection and Validation. Final Technical Report. Vol. 2. National Technical Information Service, Springfield, 1977. Available at: <http://www.dtic.mil/dtic/tr/fulltext/u2/a049015.pdf>
6. McCall J.A., Richards P.K., Walters G.F. Factors in Software Quality: Preliminary Handbook on Software Quality for an Acquisition Manager. Final Technical Report. Vol. 3. National Technical Information Service, Springfield, 1977.
7. Boehm B.W., Brown J.R., Kaspar H., Lipow M., MacLeod G.J., Merritt M.J. Characteristics of Software Quality. TRW Series of Software Technology, Amsterdam, North Holland, 1978, 166 p.

8. Grady R.B., Caswell D.L. Software Metrics: Establishing a Company-Wide Program. Prentice-Hall, 1987, 275 p.
9. Ghezzi C., Jazayeri M., Mandrioli D. Fundamental of Software Engineering. Prentice-Hall, NJ, USA, 1991, 543 p.
10. Dromey G.R. A Model for Software Product Quality. *Transactions of Software Engineering*, 1995, vol. 21, iss. 2, pp. 146–162. doi: 10.1109/32.345830
11. Hyatt L.E., Rosenberg L.H. A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality. Proceedings of Product Assurance Symposium and Software Product Assurance Workshop. Noordwijk, 1996, pp. 209–212.
12. Stein C., Cox G., Etzkorn L. Exploring the Relationship between Cohesion and Complexity. *Journal of Computer Science*, 2005, vol. 1, iss. 2, pp. 137–144. doi: 10.3844/jcssp.2005.137.144
13. Fenton N.E., Neil M. A Critique of Software Defect Prediction Models. *IEEE Transactions on Software Engineering*, 1999, vol. 25, iss. 3, pp. 675–689.
14. Laplante P.A. What Every Engineer Should Know about Software Engineering. CRC Press, 2007, p. 176.
15. Bansiya J., Davis C. A Hierarchical Model for Object-Oriented Quality Assessment. *IEEE Transactions on Software Engineering*, 2002, vol. 28, iss. 1, pp. 4–17. doi: 10.1109/32.979986
16. Bass L., Clements P., Kazman R. Software Architecture in Practice. Addison Wesley, 2003, 528 p.
17. Khosravi K., Gueheneuc Y. On Issues with Software Quality Models. Proceedings of 9th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering. 2005, pp. 70–83.
18. Chang C., Wu C., Lin H. Integrating Fuzzy Theory and Hierarchy Concepts to Evaluate Software Quality. *Software Quality Control*, 2008, vol. 16, iss. 2, pp. 263–267.
19. Sharma A., Kumar R., Grover P.S. Estimation of Quality for Software Components: An empirical approach. *ACM SIGSOFT Software Engineering Notes*, 2008, vol. 33, iss. 6, pp. 1–10. doi: 10.1145/1449603.1449613
20. Исмаил Е.Е. Современные модели качества программных средств и их особенности // МНЖ «Поиск». 2015. № 3. С. 272–282.

Информация о конфликте интересов

Мы, авторы данной статьи, со всей ответственностью заявляем о частичном и полном отсутствии фактического или потенциального конфликта интересов с какой бы то ни было третьей стороной, который может возникнуть вследствие публикации данной статьи. Настоящее заявление относится к проведению научной работы, сбору и обработке данных, написанию и подготовке статьи, принятию решения о публикации рукописи.

MODELS, METHODS, PARAMETERS, CHARACTERISTICS AND METRICS USED IN EXPERT SYSTEMS OF EVALUATING THE QUALITY OF THE DESIGN AND DEVELOPMENT OF INNOVATIVE SOFTWARE PROJECTSSergei N. LARIN^{a,*}, Larisa Yu. LAZAREVA^b, Tat'yana S. LARINA^c^a Central Economics and Mathematics Institute, RAS, Moscow, Russian Federation
sergey77707@rambler.ru^b Institute of International Accounting Standards and Management, Moscow, Russian Federation
lazarlu@rambler.ru^c Institute of International Accounting Standards and Management, Moscow, Russian Federation
lartaser@rambler.ru

* Corresponding author

Article history:Received 11 January 2017
Received in revised form
25 January 2017
Accepted 13 February 2017
Available online 15 June 2017**JEL classification:** D71, D72,
O11, P25, Q01<https://doi.org/10.24891/re.15.6.1187>**Keywords:** innovative software
project, quality assessment, expert
system, model, method**Abstract****Subject** The article takes a look at the existing expert systems to evaluate the quality of development of innovative software projects, as well as the models and methods used to obtain an assessment of their quality.**Objectives** The article aims to perform a comparative analysis of the existing models and methods used to assess the quality of the development of innovative software projects, as well as identify indicators, characteristics and metrics to quantify them when creating expert systems of evaluation**Methods** For the study, we used comparative statistical analysis methods, decomposition, the methods of expert evaluation, analytic hierarchy process, and the fuzzy sets methodology.
Results We have identified a number of the existing models used to assess the quality of modern innovative software projects, and determined the composition of indicators, their qualitative characteristics and subcharacteristics conducive to creating expert systems to evaluate the quality of their development.**Conclusions** The article concludes that, in order to obtain a comprehensive assessment of the quality of modern innovative software projects, it is appropriate to use different models simultaneously. To assess the quality of each particular innovative software project, an individual expert system of indicators, characteristics and subcharacteristics of the project's quality properties and measuring metrics should be created.

© Publishing house FINANCE and CREDIT, 2017

AcknowledgmentsThe article was supported by the Russian Foundation for Basic Research, project No. 15-06-00044a *Development of the Methodology for Examination of the Results of Exploratory Research of Projects for Innovative Products, Services and Technologies: Economic Methods, Models, Tools, and Algorithms of Information Processing.***References**

1. Gorbachenko I.M. [Assessment of the quality of the software for the creation of systems testing]. *Fundamental'nye issledovaniya = Fundamental Research*, 2013, no. 6-4, pp. 823–827. (In Russ.)
2. Seffah A., Donyaee M., Kline R.B., Padda H.K. Usability Measurement and Metrics: A consolidated model. *Software Quality Journal*, 2006, vol. 14, iss. 2, pp. 159–178. doi: 10.1007/s11219-006-7600-8
3. Fitzpatrick R. Software Quality: Definitions and Strategic Issues. Staffordshire University, School of Computing Report, 1996, April, 34 p. Available at: <https://www.saylor.org/site/wp-content/uploads/2012/11/Software-quality-definitions-and-strategic-issues.pdf>
4. McCall J.A., Richards P.K., Walters G.F. Factors in Software Quality: Concept and Definitions of Software Quality. Final Technical Report. Vol. 1. National Technical Information Service, Springfield, 1977. Available at: <http://www.dtic.mil/dtic/tr/fulltext/u2/a049014.pdf>
5. McCall J.A., Richards P.K., Walters G.F. Factors in Software Quality: Metric Data Collection and Validation. Final Technical Report. Vol. 2. National Technical Information Service, Springfield, 1977. Available at: <http://www.dtic.mil/dtic/tr/fulltext/u2/a049015.pdf>

6. McCall J.A., Richards P.K., Walters G.F. Factors in Software Quality: Preliminary Handbook on Software Quality for an Acquisition Manager. Final Technical Report. Vol. 3. National Technical Information Service, Springfield, 1977.
7. Boehm B.W., Brown J.R., Kaspar H., Lipow M., MacLeod G.J., Merritt M.J. Characteristics of Software Quality. TRW Series of Software Technology, Amsterdam, North Holland, 1978, 166 p.
8. Grady R.B., Caswell D.L. Software Metrics: Establishing a Company-Wide Program. Prentice-Hall, 1987, 275 p.
9. Ghezzi C., Jazayeri M., Mandrioli D. Fundamental of Software Engineering. Prentice-Hall, NJ, USA, 1991, 543 p.
10. Dromey G.R. A Model for Software Product Quality. *Transactions of Software Engineering*, 1995, vol. 21, iss. 2, pp. 146–162. doi: 10.1109/32.345830
11. Hyatt L.E., Rosenberg L.H. A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality. In: Proceedings of Product Assurance Symposium and Software Product Assurance Workshop. Noordwijk, 1996, pp. 209–212.
12. Stein C., Cox G., Etzkorn L. Exploring the Relationship between Cohesion and Complexity. *Journal of Computer Science*, 2005, vol. 1, iss. 2, pp. 137–144. doi: 10.3844/jcssp.2005.137.144
13. Fenton N.E., Neil M. A Critique of Software Defect Prediction Models. *IEEE Transactions on Software Engineering*, 1999, vol. 25, iss. 3, pp. 675–689.
14. Laplante P.A. What Every Engineer Should Know about Software Engineering. CRC Press, 2007, p. 176.
15. Bansiya J., Davis C. A Hierarchical Model for Object-Oriented Quality Assessment. *IEEE Transactions on Software Engineering*, 2002, vol. 28, iss. 1, pp. 4–17. doi: 10.1109/32.979986
16. Bass L., Clements P., Kazman R. Software Architecture in Practice. Addison Wesley, 2003, 528 p.
17. Khosravi K., Gueheneuc Y. On Issues with Software Quality Models. In: Proceedings of 9th ECOOP workshop on Quantitative Approaches in Object-Oriented Software Engineering. 2005, pp. 70–83.
18. Chang C., Wu C., Lin H. Integrating Fuzzy Theory and Hierarchy Concepts to Evaluate Software Quality. *Software Quality Control*, 2008, vol. 16, iss. 2, pp. 263–267.
19. Sharma A., Kumar R., Grover P.S. Estimation of Quality for Software Components: An empirical approach. *ACM SIGSOFT Software Engineering Notes*, 2008, vol. 33, iss. 6, pp. 1–10. doi: 10.1145/1449603.1449613
20. Ismail E.E. [Modern software quality models and their features]. *MNZh Poisk*, 2015, no. 3, pp. 272–282. (In Russ.)

Conflict-of-interest notification

We, the authors of this article, bindingly and explicitly declare of the partial and total lack of actual or potential conflict of interest with any other third party whatsoever, which may arise as a result of the publication of this article. This statement relates to the study, data collection and interpretation, writing and preparation of the article, and the decision to submit the manuscript for publication.