

УДК 004.05; 336.6

## Современные аспекты разработки программного обеспечения экономико-управленческих систем и процессов

М.Ю. Иванов

Братский государственный университет, ул. Макаренко 40, Братск, Россия  
nis@brstu.ru

Статья поступила 16.11.2012, принята 13.02.2013

*В ходе экономико-управленческой деятельности предприятий часто возникают проблемы поиска альтернатив и выбора единственно правильного решения. Нередки ситуации, связанные с отсутствием общепринятых норм для обозначения тех или иных финансово-хозяйственных показателей. Сложно учесть и формализовать и специфику экономических процессов в различных сферах деятельности, и взаимозависимость отдельных показателей, и инфляцию, по-разному воздействующую на всевозможные коэффициенты и процентные ставки, и т. д. Таким образом, как показывает практика, создание качественных программных продуктов, поддерживающих разработку и принятие управленческих решений, а также способных моделировать последствия принятия того или иного решения, является крайне сложной задачей. В статье приведены систематизированные материалы, определяющие особенности и подходы к разработке программного обеспечения экономических систем и процессов. Результатом исследования является формирование понятия качества программного обеспечения. Также рассмотрены методики измерения качества программного обеспечения и способы управления качеством программного обеспечения.*

**Ключевые слова:** программное обеспечение, качество программного обеспечения, измерение и управление качеством программного обеспечения, экономика.

## Modern aspects of development software economic management systems and processes

M.Y. Ivanov

Bratsk State University, 40, Makarenko str, Bratsk, Russia  
nis@brstu.ru

Received 16.11.2012, accepted 13.02.2013

*In the economic and management of companies often have trouble finding alternatives and select one correct solution. Not rare situation, due to the lack of generally accepted standards for the values of various financial and economic indicators. It is difficult to take into account the specific economic and formalize processes in various fields of activity, and the interdependence of individual indicators, and inflation effects on various factors and interest rates in different ways, etc. Thus, in practice, the creation of high-quality software products that support the development and management strategies, as well as the ability to model the consequences of the adoption of a decision, is extremely difficult. The article presents the systematic material defining features and approaches to software development and economic systems and processes. The result of the study is to develop the concept of software quality. Also examined methods of measuring the quality of software and methods of quality management software.*

**Key words:** software, software quality, measurement and quality management software, economy.

В настоящее время индустрия программного обеспечения (далее – ПО) находится примерно в таком же состоянии, что и фармацевтика в конце XIX века. Чуть ли не ежедневно появляются сообщения о том, что создан очередной «эликсир жизни», то есть замечательное новое программное решение непреодолимой проблемы (невысокой эффективности, слабого руководства, недовольных клиентов, плохой связи и т. д. и т. п.). Мощные рекламные кампании часто побуждают пользователей гоняться за миражами, способствуют провалу бизнес-проектов, в которых для разработки и принятия управленческих решений используются те или иные программные продукты [1].

Количество подобного ПО сейчас таково, что возникает вполне правомочный вопрос, а есть ли хоть доля правды в публикациях обо всех этих средствах и решениях? Иначе говоря, являются ли предлагаемые программные продукты качественными?

Определение термина «качество» нельзя назвать достаточно четким. Традиционно под качеством понимают совокупность характеристик какого-либо объекта, относящихся к его способности удовлетворять установленные или предполагаемые потребности. Иногда качество интерпретируется как «пригодность к действию», которая оценивается по функциональным, эксплуатационным характеристикам, степени безопасности, надежности продукции и т. д.

В индустрии ПО положение не лучше, и в отношении понятия «качество» существуют значительные разночтения. Особенно это актуально в том случае, когда речь заходит о ПО экономических систем и процессов [2, 3].

В самом узком смысле понятие качества ПО ассоциируется с такими критериями, как надежность, тестируемость, модифицируемость (этот подход свойственен разработчикам). Вместе с тем технические характеристики программного продукта в полной мере не могут определять качество ПО, поскольку не учитывают так называемый человеческий фактор [4]. Ведь программисты создают ПО не для себя, а для пользователя, у которого свой, часто противоположный взгляд на свойства ПО.

В свою очередь менеджеры и маркетологи под качеством ПО часто понимают выполнение требований заказчика в комплексе со своевременной поставкой и приемлемой стоимостью программного продукта (так называемое «качество, управляемое рыночными потребностями»). Да, это наглядное определение, но все же не качества ПО, а степени удовлетворенности пользователя. Безусловно, заказчик будет доволен, если вовремя получит программный продукт, соответствующий запросам (например, с дружественным интерфейсом, и не более) и по адекватной цене. Но если детально проанализировать эту формулу, то окажется, что все ее важные составляющие различны по своей природе и мало соотносятся с качеством ПО.

Вообще, удовлетворение требований пользователя – весьма абстрактная категория, наглядной иллюстрацией которой может быть операционная система Windows. Число ее критиков не поддается подсчету, но, как ни парадоксально, никто из них не отказывается от Windows в пользу другой программной платформы.

В публикациях экспертов популярно приравнивание качества ПО к отсутствию ошибок в последнем. Вместе с тем, на сегодняшний день не существует ни одного программного продукта, характеризующегося отсутствием в нем каких-либо ошибок (так называемый «совершенный код»). Всем известно, что после официального релиза того или иного ПО практически каждый день выходят многочисленные исправления, обновления, сервис-паки и пр. И эта тенденция характерна для любого ПО, начиная от операционных систем, систем автоматизированного проектирования, корпоративных информационных систем и заканчивая примитивными проигрывателями аудио- и видеофайлов.

Причиной ошибок в ПО является тот факт, что человек (программист) изначально подвержен совершению ошибок определенного рода: «систематических» и ошибок «мышления» [5]. Например, незнание тонких деталей проектирования, индексирование с ошибкой на единицу, ссылка на переменную, предшествующая ее определению, отсутствие инициализации совместно используемых переменных, пропуск важного условия в списке условий и т. п. Причем, для устранения ошибок в ПО до сих пор не выработан единый метод или инструмент типа «серебряная пуля»: не спасают ни модульное программирование, ни тестирование на корректность и отказоустойчивость. Конечно, критические ошибки устраняются, но вместе с тем остается и мно-

жество других необнаруженных багов (дефектов) ПО (например, ошибки документирования, ошибки избыточного кода, ошибки недоступных путей, алгоритмические ошибки, не влияющие на вычисления). Таким образом, качество ПО нельзя ассоциировать только с отсутствием в нем ошибок.

Еще одним подходом – можно сказать, подходом философским, умозрительным – является утверждение, что качество – это нечто неосоздаваемое и нематериальное. Приверженцы такого подхода утверждают, что у всех субъектов, связанных с ПО, свой взгляд на качество программных продуктов. Не секрет, что в большинстве случаев разработчик, не желая упускать заказ на создание того или иного ПО, действует в рамках своих возможностей и компетентности, придерживаясь тем самым принципа «лишь бы сделать». В свою очередь заказчик, не являясь специалистом в области программирования и остро нуждаясь в средстве автоматизации, оценивает программный продукт с позиции «лишь бы работал». Ни формальное составление технического задания (просто так принято), ни бесконечные согласования и прогоны не могут решить данную проблему, поскольку все заинтересованные лица преследуют разные цели. А в большинстве случаев создание ПО ведется вообще без личного общения заказчика и разработчика. Понятно, что ни о каком качестве программного продукта в этом случае и речи быть не может.

Представленные толкования понятия «качество ПО», а также определения, которые остались за рамками данной статьи, свидетельствуют о том, что очень многие из них, несомненно, неверны.

Каждое описание качества программного продукта весьма привлекательно, каждое говорит о чем-то, что имеет значение для индустрии ПО, и полемизировать на эту тему можно бесконечно. Вместе с тем, задача настоящего исследования заключается в попытке найти корректное определение качества ПО, которое отвечало бы современным тенденциям, и отбросить его ошибочные трактовки.

Между тем значительный объем работы вовсе не случайно посвящен рассмотрению различных определений качества ПО, поскольку и неверные толкования, и те суждения, в которых все же есть рациональное зерно, свидетельствуют о том, что качество программного продукта не может быть описано однозначно и нуждается в комплексном подходе.

Итак, что же сегодня понимается под качеством ПО?

Качество ПО – это совокупность следующих критериев, которыми должен обладать программный продукт:

- 1) *функциональность* – способность ПО точно решать задачи, которые соответствуют зафиксированным и предполагаемым потребностям пользователя;
- 2) *удобство использования* – возможность легкого изучения и использования ПО заказчиком (учет человеческого фактора). Также к этому критерию можно отнести и привлекательность ПО для пользователя;
- 3) *надежность* – способность ПО выполнять требуемые задачи в обозначенных условиях, самостоятельно и корректно восстанавливаться после сбоев в работе;

4) *эффективность* – способность ПО решать поставленные задачи в соответствии с выделенными ресурсами (объемом памяти, временными интервалами и т. п.);

5) *удобство сопровождения* – возможность понимания принципов работы ПО, легкость тестирования, модифицирования и адаптации;

6) *переносимость* – возможность беспрепятственного переноса ПО с одной программно-аппаратной платформы на другую.

Разные специалисты дают этим свойствам ПО разные названия. Так, например, переносимость ПО часто именуется портативностью, а некоторые авторы удобство сопровождения ПО разделяют на два отдельных критерия: тестируемость и модифицируемость и т. д. [1].

Следует отметить, что вышеперечисленные критерии качества ПО предложены в соответствии с рекомендациями международного стандарта ISO 9126-2008 Software engineering. Product quality.

Порядок перечисления приведенных выше признаков качества ПО не соответствует каким-либо приоритетам. Другими словами, нет общепринятого ранжирования по степени их важности. Исключение составит, быть может, лишь функциональность, поскольку способность ПО точно решать поставленные задачи и есть та главная цель, в соответствии с которой программный продукт, собственно, и создается.

Это, конечно, не означает, что критерии качества ПО не следует упорядочивать вовсе. Жизненно важно установить эту последовательность с самого начала для каждого отдельно взятого проекта.

Таким образом, с учетом вышеизложенного можно считать, что приемлемое определение качества программного обеспечения найдено.

Теперь, однако, возникает масса новых, не менее значимых вопросов, основными из которых являются: «На ком лежит ответственность за качество ПО?» и «Как измерить достигнутый уровень качества любого (!) программного продукта?».

Итак, кто отвечает за качество ПО? В большинстве различных публикаций и курсов, посвященных качеству программных продуктов, совершенно однозначно говорится, что это задача менеджеров, и даже приводятся абсолютно недвусмысленные тезисы: «Хороший менеджмент важнее хорошей технологии» и «Менеджмент может сделать программный продукт качественным» [1]. Так ли это? Что если менеджмент может сделать программный продукт, например, продаваемым, а не качественным?

Если посмотреть на определение качества ПО, опирающееся на совокупность свойств программного продукта, то можно увидеть и сугубо технические критерии. Так, удобство сопровождения подразумевает, что разработчик сумеет спроектировать ПО таким образом, что впоследствии его модификация не составит труда. Надежность означает, что методы создания ПО обеспечивают сведение к минимуму вероятность проникновения в него ошибок. Переносимость – свойство ПО, спроектированного таким образом, что его с минимальными затратами можно перенести с одной платформы на другую. Эти и многие другие признаки качества программного продукта требуют для их достиже-

ния глубоких знаний в области программирования. Но обладает ли этими знаниями менеджер?

Второй глобальный вопрос: «Как измерить качество ПО?». Все дело в том, что не только качество программного продукта с трудом поддается определению. То же самое можно сказать и о свойствах ПО, определяющих его качество. Практически невозможно выразить в виде числа понятность, модифицируемость, тестируемость, то есть большую часть признаков качества ПО. Да, можно определить по какой-либо шкале надежность и, до некоторой степени, эффективность, но достоверность измерений качества ПО не станет от этого выше.

Сегодня менеджеры могут сказать, что хорошие управленцы стремятся оценивать качественные показатели, а не количественные. И чем они только не руководят! И интеллектуальными процессами, и даже творческой деятельностью, совершенно не представляя себе, какими числами они должны при этом руководствоваться.

Вместе с тем, в области разработки ПО значимость измерений для менеджеров невозможно переоценить. Абсолютно ясно, что менеджеру в этом случае необходимо знать ответы на такие вопросы, как «сколько?», «когда?» и т. д. и т. п. В настоящее время в индустрии ПО образовалось целое направление работ по составлению метрик. И новые параметры программных продуктов, требующие оценки, и инструменты измерения давно известных явлений и процессов появляются для менеджеров достаточно интенсивно.

Также было организовано большое количество проектов, посвященных как выгодам, обусловленным применением метрик, так и затратам на их создание. Большинство результатов оказались позитивными. К примеру, в исследованиях NASA установлено, что текучая стоимость сбора необходимых метрических данных не превышает 3 %, обработка и анализ данных – от 4 до 6 %, что дает в сумме от 7 до 9 % общей стоимости проекта. В NASA, учитывая ценность полученных результатов по оценке качества ПО, считают такие затраты приемлемыми [6].

В истории развития средств оценки качества ПО, связанных с метриками, было много пробелов. Так, менеджеры слишком часто собирали данные о ПО, не имевшие никакого значения или обходившиеся слишком дорого. Способы сбора метрик становятся хоть сколько-нибудь рациональными лишь после появления методики GQM (Goal-Question-Metric), которую предложил В. Бэзилли. Методика GQM включает в себя три этапа: определение целей, которые необходимо достичь посредством сбора метрик; формулирование вопросов, ответив на которые, можно узнать, достигнуты ли цели; и, наконец, сбор метрик, позволяющих ответить на эти вопросы [7].

Еще одна трудность была связана с теорией программного обеспечения, которая была попыткой известного программиста М. Холстеда создать науку, подкрепляющую практику разработки ПО [8]. В конце 70-х гг. XX века эта цель казалась достойной и важной. Холстед определил и сами параметры качества программных продуктов, и способы их измерения. Однако изучение получаемых один за другим результатов по-

казало, что их ценность для индустрии ПО была либо нулевой, либо отрицательной. Некоторые эксперты тех лет даже приравнивали теорию ПО к «разновидности астрологии», в связи с чем, сбор научных данных о программных продуктах в конечном итоге приобрел сомнительную известность и был практически забыт. Разработчики, помнящие фиаско теории ПО, до сих пор склонны придавать всем метрикам негативную окраску.

Сбор метрик ПО в настоящее время производится довольно часто. Метрики ПО определены и в нормативных документах, и в публикациях специалистов [9]. Самыми популярными метриками являются: количество ошибок, обнаруженных после выхода ПО; количество изменений (или запросов на изменения); полнота (точность) документации; время, необходимое на поиск (коррекцию ошибок); распределение ошибок по типам (классам); покрытие кода тестами; сложность модуля (проекта); количество повторно использованных строк кода и т. д.

И все же эти оценки, что любопытно, не находят широкого применения. В обзорах инструментальных средств и технологий для менеджеров, работающих в индустрии ПО, средства получения метрик, как правило, упоминаются практически в последнюю очередь [10]. Конечно, есть и исключения, в частности компании IBM, Motorola, HP, обращающие на метрики пристальное внимание. Но по большей части методы оценки ПО, основанные на метриках, совершенно игнорируются. Почему? Может быть, трудно собрать какие-то необходимые данные. А может быть, менеджеры не уверены в достоверности этих измерений. Не секрет, что качество ПО они сами и определяют, а это, в свою очередь делает все оценки субъективными.

Дело не только в том, что достижение качества ПО – проблема техническая, но и в том, что сторонники ведущей роли менеджмента в решении этой задачи часто идут неправильным путем.

Менеджеры считают, что эффективность новых подходов проявится сразу, как только их начнут применять, причем в любой области, в том числе и в индустрии ПО. Это ошибочное мнение способствует тому, что на рынке ПО становится все больше программных продуктов, не всегда отвечающих предъявляемым к ним требованиям.

Нередко в роли главного врага качества программных продуктов выступает время. Таким образом, с одной стороны менеджмент мотивирует и внедряет различные методики, а с другой – устанавливает сроки, под давлением которых часто утрачивается качество ПО.

Итак, каковы результаты данного исследования?

Качество программного продукта не может быть описано однозначно и нуждается в комплексном подходе.

Методы оценки качества ПО существуют, но практически не используются.

Утверждение о том, что качество ПО должны обеспечивать управленцы, ошибочно.

Конечно, руководство играет очень важную роль в обеспечении качества программного продукта. Менеджеры могут устранить различные препятствия и создать атмосферу, в которой достижению качества присваивается высокий приоритет, могут нанять высококлассных специалистов, а это, несомненно, будет лучшим способом обеспечить качество программного обеспечения.

### *Литература*

1. Glass R. Facts and Fallacies of Software Engineering. Boston: Addison Wesley Pub Date, 2002. 224 p.
2. Иванов М.Ю. Структура и принципы функционирования экспертных систем для оценки деятельности хозяйствующего субъекта // Проблемы социально-экономического развития Сибири. 2012. № 3 (9). С. 18-22.
3. Иванов М.Ю. Экспертные системы для оценки деятельности хозяйствующего субъекта // Проблемы социально-экономического развития Сибири. 2012. № 3 (9). С. 23-27.
4. DeMarco T., Lister T. Peopleware: productive projects and teams. New York: Dorset House, 2005. 261 p.
5. Davis A. 201 Principles of Software Development. New York: McGraw: Hill, 1995. 256 p.
6. Agresti W. Manager's Handbook for Software Development. Goddard: NASA: Goddard Space Flight Center, 1990. 79 p.
7. Basili V. Software Modeling and Measurement: The Goal Question Metric Paradigm // Computer Science Technical Report Series. 1992. № 9. P. 528-532.
8. Halstead M. Elements of Software Science, Operating, and Programming Systems. Halstead. NY: Elsevier, 1977. 142 p.
9. Boehm B., Basili V. Software Defect Reduction Top 10 List // Software management. 2001. № 1. P. 135-137.
10. Jung H., Kim S., Chung C. Measuring Software Product Quality // IEEE Software. 2004. № 9-10. P. 88-92.

### *References*

1. Glass R. Facts and Fallacies of Software Engineering / R. Glass. Boston: Addison Wesley Pub Date, 2002. 224 p.
2. Ivanov M.Yu. The structure and principles of expert systems functioning to evaluate economic entity activity // Problemy sotsial'no-ekonomicheskogo razvitiya Sibiri. 2012. №3 (9). S. 18-22.
3. Ivanov M.Yu. Expert systems for economic entity activity assessment // Problemy sotsial'no-ekonomicheskogo razvitiya Sibiri. 2012. № 3 (9). S. 23-27.
4. DeMarco T., Lister T. Peopleware: productive projects and teams. New York: Dorset House, 2005. 261 p.
5. Davis A. 201 Principles of Software Development. New York: McGraw: Hill, 1995. 256 p.
6. Agresti W. and others. Manager's Handbook for Software Development. Goddard: NASA: Goddard Space Flight Center, 1990. 79 p.
7. Software Modeling and Measurement: The Goal Question Metric Paradigm // Computer Science Technical Report Series. 1992. №9. P. 528 - 532.
8. Halstead M. Elements of Software Science, Operating, and Programming Systems. NY: Elsevier, 1977. 142 p.
9. Software Defect Reduction Top 10 List // Software management. 2001. №1. P. 135-137.
10. Measuring Software Product Quality // IEEE Software. 2004. № 9-10. P. 88-92.