

УДК 004.89

Д. А. Маевский, д-р техн. наук,  
Ю. Ю. Козина, канд. техн. наук

### ГДЕ И КОГДА ФОРМИРУЕТСЯ КАЧЕСТВО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ?

**Аннотация.** Концепция проекта в области информационных технологий описывает процесс создания и сопровождение систем в виде жизненного цикла, представляя его как последовательность стадий и выполняемых процессов. При этом, большое внимание уделяется оценке качества программного обеспечения в процессе его жизненного цикла. Актуальность работы обоснована необходимостью учитывать требования к программному обеспечению с учетом закладываемых на каждом этапе жизненного цикла характеристик качества. Проведен анализ принадлежности закладываемых характеристик качества, который рекомендуется использовать в реальных условиях разработки программного обеспечения.

**Ключевые слова:** качество программного обеспечения, ISO 25010, жизненный цикл программного обеспечения, модель качества продукта, модель качества при использовании продукта, спецификации, тестирование, анализ требований, сопровождение

D. Maevskiy, ScD,  
Y. Kozina, PhD

### WHERE AND WHEN IS FORMED OF SOFTWARE QUALITY?

**Abstract.** The concept of the project in the information technology area describes the process of software development in the form of the life cycle, presenting it as a sequence of stages. In turn, give one's attention to assessing the quality of the software during its life cycle. Relevance of this article to justify the needing to considering the requirements for software with taking stock are set at each stage of quality characteristics.

The analysis of belonging characteristics of software quality is executed and it is recommended to use in real condition software development.

**Keywords:** software quality, ISO 25010, life cycle of software, quality model, quality model in using the product, specifications, testing, requirements analysis, maintenance

Д. А. Маєвський, д-р техн. наук,  
Ю. Ю. Козіна, канд. техн. наук

### ДЕ И КОЛИ ВСТАНОВЛЮЄТЬСЯ ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ?

**Анотація.** Концепція проекту в області інформаційних технологій описує процес створення і супровід систем у вигляді життєвого циклу, представляючи його як послідовність стадій і виконуваних процесів. При цьому, велика увага приділяється оцінці якості програмного забезпечення в процесі його життєвого циклу. Актуальність роботи обґрунтована необхідністю враховувати вимоги до програмного забезпечення з урахуванням характеристик якості, що закладаються на кожному етапі життєвого циклу. Проведено аналіз належності характеристик якості, який рекомендується використовувати в реальних умовах розробки програмного забезпечення.

**Ключові слова:** якість програмного забезпечення, ISO 25010, життєвий цикл програмного забезпечення, модель якості продукту, модель якості при використанні продукту, специфікації, тестування, аналіз вимог, супровід

**Введение.** Жизнь современного общества сегодня невозможно представить без информационных технологий. Разнообразные компьютерные системы настолько прочно вошли в нашу жизнь, что человечество, само того не замечая, постепенно стало заложником их надежной и безотказной работы [1]. Недостаточно качественное программное обеспечение (ПО) информационных систем сегодня может вызвать техногенные аварии, приостановку функционирования финансовой системы целых госу-

дарств, материальные потери и опасность для жизни людей. Поэтому обеспечение качества на всех этапах разработки ПО в настоящее время становится приоритетной и актуальной задачей. При этом важно понимать, какие именно характеристики качества ПО закладываются на определенных этапах его разработки. Это вызвано потребностью учесть требования к ПО, начиная с ранних этапов разработки, выстроить процесс написания кода и тестирования ПО с учетом всех возможных характеристик качества. За счет определения принадлежности закладываемых характеристик качества ПО, соответст-

вующих каждому этапу его разработки, удастся избежать возникновения непредвиденных ситуаций на конечных этапах разработки, приводящих к необходимости переписывать код (частично или полностью).

**Цель работы**, направленная на установление соответствия закладываемых характеристик качества ПО каждому этапу разработки в процессе его жизненного цикла, является актуальной.

**Постановка задачи.** Согласно стандарта ISO 25010 [2 – 4] все характеристики качества ПО описаны в двух моделях: модель качества при использовании и модель качества продукта. Модель качества при использовании согласно стандарту ISO 25010 определяет степень, с которой продукт или система могут быть использованы для конкретных пользователей, удовлетворения их потребностей для достижения конкретных целей с результативностью, эффективностью, свободой от рисков и удовлетворением в соответствии с определенными контекстами использования.

К характеристикам (подхарактеристикам) модели качества при использовании ПО относятся: эффективность, результативность, удовлетворение (полезность, доверие, радость, комфорт), отсутствие рисков (смягчение экономических рисков, смягчение рисков для здоровья и защищенности, смягчение рисков окружающей среды), контекстное покрытие (контекстная полнота, гибкость).

Модель качества продукта определяет восемь характеристик (подхарактеристик) качества: функциональная пригодность (функциональная полнота, правильность), надежность (завершенность, готовность, устойчивость к ошибке, восстанавливаемость), эффективность работы (поведение во времени, мощность), удобство использования (понятность пригодности, обучаемость, легкость использования, защита от совершения ошибок, техническая доступность), защищенность (конфиденциальность, целостность, аутентичность, доказуемость действий, подотчетность), совместимость (сосуществование, способность к взаимодействию), сопровождаемость (модульность, повторная используемость, анализируемость, изменяемость, стабильность при модификации, тестируе-

мость) и мобильность (адаптируемость, настраиваемость, взаимозаменяемость).

Вероятно, самым распространенным мотивом обращения к понятию жизненного цикла ПО является потребность в систематизации работ в соответствии с технологическим процессом. Этому назначению хорошо соответствует так называемая общепринятая модель жизненного цикла ПО, согласно которой ПО проходят в своем развитии две фазы: разработка и сопровождение [5 – 6]. Фазы разбиваются на ряд этапов (рис. 1).

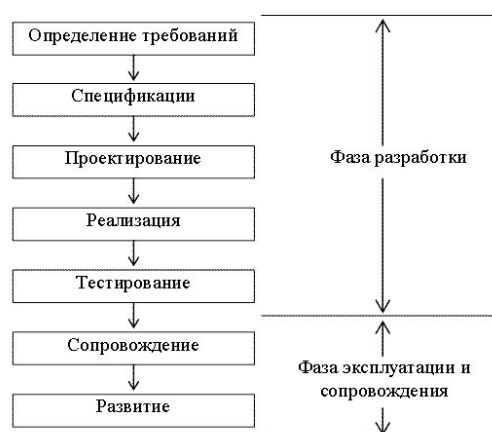


Рис. 1. Общепринятая модель жизненного цикла ПО

Модель имеет линейный вид, согласно которому разработка начинается с идентификации потребности в новом ПО, а заканчивается передачей продукта разработки в эксплуатацию. Общепринятая модель жизненного цикла является идеальной, так как только очень простые задачи проходят все этапы без каких-либо итераций – возвратов на предыдущие шаги технологического процесса [7]. При программировании, например, может обнаружиться, что реализация некоторой функции очень громоздка, неэффективна и вступает в противоречие с требуемой от системы производительностью. В этом случае требуется перепроектирование, а может быть, и переделка спецификаций. При разработке больших нетрадиционных систем необходимость в итерациях возникает регулярно на любом этапе жизненного цикла как из-за допущенных на предыдущих шагах ошибок и неточностей, так и из-за изменений внешних требований к условиям эксплуатации системы.

Таковы мотивы появления и практического использования таких моделей жизненного цикла ПО как: каскадная (водопадная), итеративная и инкрементальная, спиральная или модель Бозма [8 – 9]. В разное время, в разных источниках приводится разный список моделей и их интерпретация. Например, ранее, инкрементальная модель понималась как построение системы в виде последовательности сборок (релизов), определенной в соответствии с заранее подготовленным планом и заданными (уже сформулированными) и неизменными требованиями. Сегодня об инкрементальном подходе чаще всего говорят в контексте постепенного наращивания функциональности создаваемого продукта [10]. Может показаться, что индустрия пришла, наконец, к общей «правильной» модели. Однако, каскадная модель, подвергаясь критике теорией и практикой, продолжает встречаться в реальной жизни. Спиральная модель является ярким представителем эволюционного взгляда, и в то же время, представляет собой единственную модель, которая уделяет явное внимание анализу и предупреждению рисков.

**Решение задачи.** В целях прозрачности и простоты восприятия проводимого анализа закладываемых характеристик качества ПО, авторами сделана попытка выполнить такой анализ в рамках общепринятой линейной модели жизненного цикла ПО. При выполнении данного анализа авторами упомянуты только «существующие на практике» характеристики (подхарактеристики) качества ПО, которые закладываются на определенных этапах жизненного цикла его разработки.

«*Эффективность*», представляющая точность и полноту, с которой пользователи достигают поставленные цели, закладывается на этапе определения требований.

«*Результативность*», определяющая затраченные ресурсы в их связи с точностью и полнотой, с которой пользователи достигают поставленные цели, закладывается на том же этапе, в сопровождении с работой бизнес-аналитика. Она заключается в сборе требований к программному продукту, разработке технического задания на создание ПО, проектировании документального оформления системной и программной архитектур, постановке задач на разработку и тестирование.

«*Удовлетворение*» как степень, в которой потребности пользователя при использовании продукта в соответствии с требуемым контекстом удовлетворены, закладывается на этапе определения требований к продукту. Особое внимание также уделяется проверке соблюдения этой характеристики на этапе тестирования.

«*Контекстное покрытие*» как степень, в которой продукт или система могут быть использованы с эффективностью, результативностью, свободой от рисков закладывается на этапе определения требований к продукту, а также относится и ко всем последующим этапам жизненного цикла.

«*Функциональная пригодность*» – применяется на этапе описания функциональных требований к продукту и спецификаций. Отвечает за описание набора функций, с помощью которых пользователь в полной мере должен достигать поставленных задач.

«*Надежность*», как степень, в которой система или компонент выполняет требуемые функции в заданных условиях за определенный период времени, закладывается на этапе тестирования.

«*Мощность*» – описывается как в функциональных требованиях к ПО так и в документах и соглашениях контрактов. Нагрузка измеряется в способности приложения обрабатывать определенный объем данных в единицу времени: обработка больших массивов данных, обработка большого количества пользователей, одновременно работающих с приложением, количество действий пользователей в единицу времени и пр. Данная метрика зависит как от реализации программного кода, так и от характеристик «железа», на котором ПО работает. Большинство коммерческих приложений имеет минимальный порог мощности, который оговаривается с клиентами.

«*Удобство использования*» оценивается на этапе анализа функциональных требований, подготовки тестов и тестирования. Проверяется, является ли функциональность понятной пользователю, позволяет ли она реализовать цели пользователя за наименьшее количество действий.

«*Защищенность*» – оценка производится на этапе создания тестов. Подразумевается защита данных от атак хакеров. Например, SSL,

Passwords mask, защита от SQL инъекций и многое другое.

«Совместимость» – задается при создании спецификаций. Обозначает способность работы ПО на различных операционных системах, с различными типами аппаратного обеспечения, а также с различными форматами данных.

«Сосуществование» – оценка имеет смысл только если на одном сервере установлены несколько программных продуктов, которые напрямую не работают друг с другом, но используют одинаковые ресурсы и могут конфликтовать. Либо же, если в системе, в большинстве случаев у пользователей установлены программы другой фирмы, которые могут конфликтовать, но с которыми нет интеграции. Оценка происходит на этапе проектирования.

«Модульность и уровень повторного использования» – данная характеристика является правилом при написании любого кода в любом приложении. Программа не может являться одним сплошным набором кода и данных. Она должна делиться на модули, с целью облегчения внесения изменений в программу. Оценка данной характеристики производится техническим архитектором на этапе проектирования и создания архитектуры ПО, а также на этапе анализа кода. Отдел тестирования в данном процессе не участвует.

Выполненный анализ принадлежности закладываемых характеристик качества предложено представить в обобщенном виде на рис. 2.

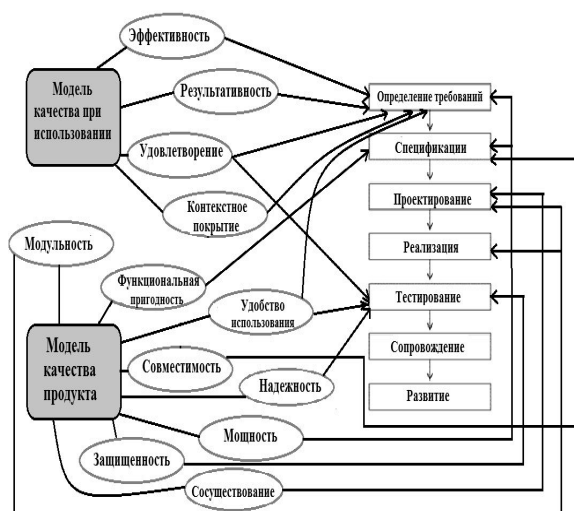


Рис. 2. Принадлежность характеристик качества этапам жизненного цикла программного обеспечения

Авторами предложено упомянуть такую характеристика качества ПО как отказоустойчивость несмотря на то, что она не входит в модель качества продукта. Эта характеристика показывает способность программы сохранять работоспособность в нештатных ситуациях, таких, например, как сбой хранилища данных при сохранении информации, нехватка дискового пространства в процессе работы и прочее. Она оценивается на этапе полного (full) тестирования.

**Выводы.** Проведенный анализ принадлежности закладываемых характеристик качества ПО, имеет практическую ценность для всех участников процесса разработки ПО – менеджеров проектов, бизнес-аналитиков, программистов и тестировщиков. Те или иные характеристики качества имеют большую или меньшую значимость в зависимости от функционального назначения программного продукта. Поэтому, выделив наиболее значимые для конкретного проектируемого ПО характеристики качества, можно выделить наиболее критичные этапы его разработки и сосредоточить на них основные усилия. Это позволит предотвратить ситуации, связанные с возвратом на предыдущие этапы жизненного цикла, снизить стоимость разработки ПО и выдержать запланированные сроки.

Список использованной литературы

- Jobs S. Thoughts on Flash [Electronic Source] (In English), available at: <http://www.apple.com/hotnews/thoughts-onflash/> (accessed 01.04.2010).
- In use Qualities from ISO/IEC 25010 [Electronic Source], *International Organization for Standardization* (In English), available at: <http://www.irit.fr/recherches/ICS/projects/twintide/upload/435.pdf> (accessed 01.03.2011).
- Денисюк А. В. Актуальные проблемы качества программного обеспечения /А. В. Денисюк, В. В. Любченко // *Электротехнические и компьютерные системы*. – К. : Техника. – 2013. – № 09 (95) – С. 142 – 148.
- ISO/IEC 25010:2011 Systems and Software Engineering – SQuaRE (Final Draft) – System and software Quality Models [Electronic Source], *International Organization for Standardization* (In English), available at:

[http://pef.czu.cz/~papik/doc/MHJS/pdf/ISOIEC\\_FDIS25010\\_\(E\).pdf](http://pef.czu.cz/~papik/doc/MHJS/pdf/ISOIEC_FDIS25010_(E).pdf) (accessed 01.03.2011).

5. Блэк Р. Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование / Р. Блэк. – М. : Лори. – 2006. – 125 с.

6. Дастин Э. Автоматизированное тестирование программного обеспечения / Э. Дастин, Д. Рэшка – М. : Лори. – 2003. – 256 с.

7. Модели жизненного цикла программного обеспечения [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/111674/> – (дата доступа 11.01.2011).

8. Общие модели жизненного цикла [Электронный ресурс]. – Режим доступа: [http://www.e-uni.ee/e-kursused/eucip/arendus\\_vk/122\\_.html](http://www.e-uni.ee/e-kursused/eucip/arendus_vk/122_.html) – (дата доступа 12.02.2012).

9. Каскадная модель жизненного цикла разработки ПО [Электронный ресурс]. – Режим доступа : <http://www.4stud.info/software-construction-and-testing/lecture6.html> – (дата доступа 13.12.2012).

10. Итеративная и инкрементальная разработка [Электронный ресурс]. – Режим доступа : <http://www.osp.ru/os/2003/09/183412/> – (дата доступа 11.10.2003).

Получено 25.03.2015

#### References

1. Jobs S., (2010), Thoughts on Flash [Electronic Source] (In English), available at: <http://www.apple.com/hotnews/thoughts-onflash/> (accessed 01.04.2010).

2. In use Qualities from ISO/IEC 25010 [Electronic Source], *International Organization for Standardization* (In English), available at: <http://www.irit.fr/recherches/ICS/projects/twintide/upload/435.pdf> (accessed 01.03.2011).

3. Denisyuk A.V., (2013), Actualnie problemi kachestva programmnoho obespecheniya [Actual Problems of Software Quality], *Elektrotehnicheskie i Computernie Sistemi Publ.*, Odessa, Ukraine, Vol. 09 (95), pp. 142 – 148 (In Russian),  
url: [http://nbuv.gov.ua/jpdf/etks\\_2013\\_9\\_22.pdf](http://nbuv.gov.ua/jpdf/etks_2013_9_22.pdf)

4. ISO/IEC 25010:2011 Systems and Software Engineering –SQuaRE (Final draft) – System and Software Quality Models [Electronic Source], *International Organization for Standardization* (In English), available at:

[http://pef.czu.cz/~papik/doc/MHJS/pdf/ISOIEC\\_FDIS25010\\_\(E\).pdf](http://pef.czu.cz/~papik/doc/MHJS/pdf/ISOIEC_FDIS25010_(E).pdf) (accessed 01.03.2011).

5. Black R. Klyuchenie processi testirovaniya. Planirovanie, podgotovka, provedenie, sovershenstvovanie [Main Processing of Testing. Planning, Preparation, Execution, Improving], (2006), Moscow, Russian Federation, *Lori Publ.*, 125 p. (In Russian).

6. Dastin E., and Reshka D. Avtomatizirovanoe testirovanie programmnoho obespecheniya [Automation Software Testing], (2003), Moscow, Russian Federation, *Lori Publ.*, 256 p. (In Russian).

7. Modeli zhiznennogo cikla programmnoho obespecheniya [Models of Software Life Cycle], (In Russian), [Electronic Source], available at: <http://habrahabr.ru/post/111674/>, (accessed: 11.01.2011).

8. Obschie modeli zhiznennogo cikla [General Models of Software Life Cycle], (In Russian) [Electronic Source], available at: [http://www.e-uni.ee/e-kursused/eucip/arendus\\_vk/122\\_.html](http://www.e-uni.ee/e-kursused/eucip/arendus_vk/122_.html) (accessed: 12.02.2012).

9. Kaskadnaya model zhiznennogo cikla razrabotki PO [Waterfall Model of Software Life Cycle Development], (In Russian) [Electronic Source], available at: <http://www.4stud.info/software-construction-and-testing/lecture6.html> – (accessed 13.12.2012).

10. Iterativnaya i incrementalnaya razrabotka [Iterative and incremental development], (In Russian) [Electronic Source], available at: <http://www.osp.ru/os/2003/09/183412/> – (accessed 11.10.2003).



Козина  
Юлия Юрьевна,  
канд. техн. наук, доцент  
каф. прикладной математики и  
информационных технологий  
Одесского нац. политехнич. ун-та,  
E-mail: yuliyak@mail.ru



Маевский  
Дмитрий Андреевич,  
д-р техн. наук, проф., зав. каф.  
теоретических основ и общей  
электротехники Одесского нац.  
политехнич. ун-та.  
E-mail: dmiry.a.maevsky@gmail.com