Место тестирования среди методов оценки качества ПО

В. В. Кулямин, О. Л. Петренко
{kuliamin, olga}@ispras.ru

Введение

В последнее десятилетие произошли революционные технологические изменения практически во всех видах деятельности, связанных с разработкой и распространением программного обеспечения. Стали активно и широко использоваться такие подходы, как

- Итеративные процессы разработки ПО
- Методы обеспечения и контроля качества ПО на всех этапах разработки, нацеленные на повышение степени удовлетворения клиентов
- Объектно-ориентированные методы анализа, проектирования и разработки ПО
- Проектирование и разработка ПО с использованием формализованных графических языков моделирования, таких как UML
- Использование инструментов CASE, поддерживающих автоматизированные преобразования из графических языков в языки программирования и обратно

В связи с такими масштабными изменениями во взглядах архитекторов, программистов и руководителей проектов требуется пересмотр традиционных представлений о месте в процессах разработки ПО различных видов деятельности, которые и составляют эти процессы. Такой пересмотр происходит, но зачастую, в отсутствии четких критериев рассмотрения сложных понятий, приводит к радикализму, неоправданному завышению или занижению роли той или иной деятельности.

Предметом данной статьи является современная оценка места тестирования в процессе производства ПО. Можно встретить совершенно противоположные мнения о значении тестирования в новых процессах разработки, и его важности для системы обеспечения качества в целом. Одни говорят, что в современных условиях тестирование является по большей части анахронизмом, что новые методы проектирования и разработки ПО в сочетании с методами обеспечения качества на ранних этапах разработки, сами по себе гарантируют получение качественного продукта, что традиционное тестирование является неэффективной затратой средств, не дающей видимого вклада в повышение качества продукта, и в то же время откладывающей выход продукта на рынок.

Другие считают, что новые методы разработки ПО и управления проектами требуют новых методов тестирования, что в новых условиях возникают новые риски и новые виды ошибок в ПО, и значение тщательного тестирования возрастает, что эффективность новейших методов обеспечения качества часто переоценивается, и указывают при этом на отмеченное пользователями общее снижение надежности коммерческого ПО в последние годы.

Данная работа представляет собой попытку исследования понятия качества ПО и того места, которое занимает тестирование среди методов обеспечения и контроля качества.

Эволюция понятия качества ПО

Что такое качественное программное обеспечение?

Если спросить об этом достаточно широкую группу людей, имеющих дело с разработкой, продажей и использованием ПО, можно получить следующие ответы:

- Легко использовать
- Хорошая производительность
- Нет ошибок
- Не портит пользовательские данные при сбоях
- Можно использовать на разных платформах

- Может работать 24 часа в сутки и 7 дней в неделю
- Легко добавлять новые возможности
- Удовлетворяет потребности пользователей
- Хорошо документировано

Все эти ответы выделяют характеристики, важные для конкретного пользователя, разработчика ПО или группы таких лиц. Однако, для повышения степени удовлетворения всех пользователей ПО, для достижения им прочного положения на рынке и повышения потенциала развития важен учет всей совокупности его характеристик, важных для всех заинтересованных сторон.
Приведенные ответы показывают, что качество ПО может быть описано большим количеством разнородных характеристик. Значит, понятие качества программы — многоплановое и может быть выражено адекватно только некоторой структурированной системой характеристик или атрибутов. Такая система характеристик называется *моделью качества.*

Качество ПО по МакКолу

Первой широко известной моделью качества ПО стала предложенная в 1977 МакКолом и др. [1] модель.
В ней характеристики качества разделены на три группы

- Факторы (factors), описывающие ПО с позиций пользователя и задаваемые требованиями.
- Критерии (criteria), описывающие ПО с позиций разработчика и задаваемые как цели.
- Метрики (metrics), используемые для количественного описания и измерения качества.

Факторы качества, которых было выделено 11, группируются в три группы по различным способам работы людей с ПО. Полученная структура изображается в виде треугольника МакКола.

Рис. 1. Треугольник МакКола.

Критерии качества — это числовые уровни факторов, поставленные в качестве целей при разработке.
Объективно оценить или измерить факторы качества непосредственно довольно трудно. Поэтому, МакКол ввел метрики качества, которые с его точки зрения легче измерять и оценивать. Оценки в его шкале принимают значения от 0 до 10. Вот эти метрики качества:

- Удобство проверки на соответствие стандартам (auditability)
- Точность управления и вычислений (accuracy)
- Степень стандартности интерфейсов (communication commonality)
- Функциональная полнота (completeness)
- Однородность используемых правил проектирования и документации (consistency)
- Степень стандартности форматов данных (data commonality)
- Устойчивость к ошибкам (error tolerance)
- Эффективность работы (execution efficiency)
- Расширяемость (expandability)
- Широта области потенциального использования (generality)
- Независимость от аппаратной платформы (hardware independence)
- Полнота протоколирования ошибок и других событий (instrumentation)
- Модульность (modularity)
- Удобство работы (operability)
- Защищенность (security)
- Самодокументированность (selfdocumentation)
- Простота работы (simplicity)
- Независимость от программной платформы (software system independence)
- Возможность соотнесения проекта с требованиями (traceability)
- Удобство обучения (training)

Каждая метрика влияет на оценку нескольких факторов качества. Числовое выражение фактора представляет собой линейную комбинацию значений влияющих на него метрик. Коэффициенты этого выражения определяются по-разному для разных организаций, команд разработки, видов ПО, используемых процессов и т.п.

Качество ПО по Боему

В 1978 Боем [2,3] предложил свою модель, по существу представляющую собой расширение модели МакКола.
Атрибуты качества подразделяются по способу использования ПО (primary use). Определено 19 промежуточных атрибутов (intermidiate construct), включающих все 11 факторов качества по МакКолу. Промежуточные атрибуты разделяются на примитивные (primitive construct), которые, в свою очередь, могут быть оценены на основе метрик.
В дополнение к факторам МакКола атрибуты качества по Боему включают следующие: ясность (clarity), удобство внесения изменений (modifiability), документированность (documentation), способность к восстановлению функций (resilience), понятность (understandability), адекватность (validity), функциональность (functionality), универсальность (generality), экономическая эффективность (economy).

Модель качества ПО ISO 9126

В 1991 году в качестве стандартной была принята модель качества ПО ISO 9126 [4,5]. Эта модель не является прямым расширением ранее предложенных. В ней оценка качества ПО основана на трехуровневом рассмотрении.

1. Цели (goals) — то, что мы хотим видеть в ПО.
2. Атрибуты (attributes) —свойства ПО, показывающие приближение к целям.
3. Метрики (metrics) — количественные характеристики степени наличия атрибутов.

Выделено 6 целей: функциональность (functionality), надежность (reliability), практичность или удобство использования (usability), эффективность (efficiency), сопровождаемость (maintainability), переносимость или мобильность (portability). Цели подразделяются на 21 атрибут качества.

В 2001 году этот стандарт был пересмотрен и расширен [6]. В него было добавлено 6 дополнительных атрибутов качества: привлекательность как атрибут практичности и степень соответствия стандартам для каждой из целей, кроме функциональности.
Ниже приведен полный список атрибутов качества ПО по стандарту ISO 9126.

- Функциональность
  - Пригодность к определенной работе(suitability)
  - Точность, правильность (accuracy)
  - Способность к взаимодействию (interoperability)
  - Соответствие стандартам и правилам (compliance)
  - Защищенность (security)
- Надежность
  - Зрелость, завершенность (обратна к частоте отказов) (maturity)
  - Устойчивость к отказам (fault tolerance)
  - Способность к восстановлению работоспособности при отказах (recoverability)
  - Соответствие стандартам надежности (reliability compliance, добавлен в 2001)
- Практичность, удобство использования
  - Понятность (understandability)
  - Удобство обучения (learnability)
  - Работоспособность (operability)
  - Привлекательность (attractiveness, добавлен в 2001)
  - Соответствие стандартам практичности (usability compliance, добавлен в 2001)
- Эффективность
  - Временные характеристики (time behaviour)
  - Использование ресурсов (resource utilisation)
  - Соответствие стандартам эффективности (efficiency compliance, добавлен в 2001)
- Сопровождаемость
  - Анализируемость (analyzability)
  - Изменяемость, удобство внесения изменений (changeability)
  - Риск возникновения неожиданных эффектов при внесении изменений (stability)
  - Контролируемость, удобство проверки (testability)
  - Соответствие стандартам сопровождаемости (maintainability compliance, добавлен в 2001)
- Переносимость, мобильность
  - Адаптируемость (adaptability)
  - Устанавливаемость, удобство установки (installability)
  - Способность к сосуществованию с другим ПО (coexistence)
  - Удобство замены другого ПО данным (replaceability)
  - Соответствие стандартам переносимости (portability compliance, добавлен в 2001)

Принятые в 2001 году части 2 и 3 стандарта ISO 9126 [7,8] определяют набор метрик качества ПО. В качестве примера таких метрик приведем следующие.

1. Полнота реализации функций. Используется для измерения пригодности.
2. Корректность реализации функций. Используется для измерения пригодности.
3. Отношение числа обнаруженных дефектов к прогнозируемому. Используется для определения зрелости.
4. Отношение числа проведенных тестов к общему их числу. Используется для определения зрелости.
5. Отношение числа доступных проектных документов к указанному в их списке. Используется для измерения анализируемости.

Заметим, что третья и четвертая метрики из приведенных показывают, что *качество ПО зависит не только от самого ПО как объекта материального мира, но и от его восприятия*

заинтересованными лицами: разработчиками, пользователями, заказчиками и др.
Действительно, не изменяя самого ПО, можно повысить его качество только за счет проведения некоторых еще не проведенных тестов, поскольку значение четвертой метрики при этом увеличится.
Сравнение моделей МакКола, Боема и ISO 9126 можно найти в [9].
Далее в этой работе мы будем использовать модель качества ПО по стандарту ISO 9126. Для полноты обзора упомянем еще две модели качества ПО.

- Модель качества FURPS [10], предложенная в 1987 году.
  Эта модель по структуре похожа на модели МакКола и Боема, но в ней не уделяется достаточное внимание переносимости ПО.
- Модель Дроми [11], предложенная в 1996 году.
  Эта модель отличается от принятого в 1991 году стандарта ISO более четким выделением связей между целями, атрибутами и метриками качества. В рамках этой модели впервые указывается на необходимость рассмотрения характеристик процесса разработки ПО для оценки качества полученного продукта.

Тестирование и другие методы оценки качества ПО

Посмотрим теперь, как соотносятся понятия тестирования и качества ПО.
В 1990 году стандартом ISO принято следующее определение тестирования.
Тестирование — это наблюдение за функционированием ПО в специфических условиях с целью определения степени соответствия ПО требованиям к нему.
Это определение показывает, что

- Тестирование само по себе не изменяет ПО, а значит, не способно влиять на те метрики качества, которые зависят только от самого ПО.
- Тестирование может служить методом контроля качества ПО, а именно тех его характеристик, которые проявляются при функционировании ПО.

Несмотря на истинность первого утверждения, *тестирование способно изменить качество ПО* в той его части, которая относится к восприятию ПО заинтересованными лицами. Некоторые метрики качества (см. примеры выше) отражают такое восприятие, и на их значение тестирование способно влиять.
Сопоставим теперь тестирование с другими методами оценки качества ПО. Для этого воспользуемся отчетами по проекту SCOPE [12,13], нацеленному на выделение средств и методов оценки различных характеристик качества. В следующем списке содержатся упоминаемые в них методы и несколько методов, по каким-то причинам не использованных в этом проекте.

- ♦ Тестирование
  - Функциональное тестирование (functional tetsing)
  - Структурное тестирование, нацеленное на покрытие кода (glass box testing)
  - Лабораторное тестирование удобства использования ПО (laboratory testing)
  - Тестирование производительности (performance testing)
  - *Нагрузочное тестирование (load testing, добавлено), стрессовое тестирование (stress testing, добавлено)*
- ♦ Изучение документов с целью поиска проблемных мест и проверки соответствия стандартам, стилям, принятым правилам и соглашениям
  - Целенаправленное изучение кода (code inspection)
  - Целенаправленное изучение документации (documents inspection)
- ♦ Формальный анализ
  - Формальное доказательство свойств ПО (formal verification)
  - Анализ алгоритмической сложности (complexity analysis)
- ♦ Анализ

- Проверка статической семантики языков программирования
- Автоматический анализ кода (static analysis)
- Анализ свойств ПО, выполняемый человеком
- Анализ архитектуры и проекта (architecture review, design review)
- *Анализ процессов разработки (process analysis, добавлен)*

♦ Измерения
- Определение метрик ПО, проекта, документации
- Измерения производительности (benchmarks)
- *Профилирование (profiling, добавлено)*

♦ Моделирование, использование моделей для оценки свойств ПО
- Модели использования (usability model)
- Модели надежности (reliability model)
- *Модели функционирования: проверка на модели (model checking, добавлено), прототипирование (добавлено)*

В отчетах SCOPE не упоминаются также применяемые иногда методы оценки качества ПО, основанные на субъективном его восприятии и интуиции экспертов.
Проанализируем перечисленные методы оценки качества, сопоставив их с атрибутами качества по стандарту ISO 9126 и с объектами применения, в качестве которых выступают различные документы, представляющие ПО. Будем рассматривать следующие виды таких документов.

- Исполнимый код
- Исходный код на языке программирования высокого уровня
- Формальные модели ПО различного рода
- Проектная документация на естественных языках
- Пользовательская документация

Атрибуты качества	**Исполнимый код**	**Исходный код**	**Модели**	**Проектная документация**	**Пользовательская документация**
Пригодность	Функциональное тестирование	Проверка кода	Проверка на модели	Изучение документов	Изучение документов
Точность	Функциональное тестирование	Анализ кода	Формальный анализ	Анализ документов	
Способность к взаимодействию	Тестирование на соответствие	Анализ кода		Изучение документов	Изучение документов
Соответствие стандартам и правилам	Тестирование на соответствие	Проверка кода		Изучение документов	Изучение документов
Защищенность	Тестирование защищенности	Анализ кода	Проверка на модели, формальный анализ	Анализ документов	
Зрелость	Нагрузочное тестирование		Проверка на модели		
Устойчивость к отказам	Стрессовое тестирование	Анализ кода	Проверка на модели, формальный анализ	Анализ документов	
Способность к восстановлению	Стрессовое тестирование	Анализ кода	Проверка на модели, формальный анализ	Анализ документов	
Соответствие стандартам надежности	Тестирование на соответствие	Анализ кода		Анализ документов	
Понятность	Тестирование		Прототипирование	Анализ документов	Изучение документов
Удобство обучения	Тестирование		Прототипирование	Анализ документов	Изучение документов
Работоспособность	Тестирование		Прототипирование	Анализ документов	Изучение документов

Привлекательность	Тестирование		Прототипирование	Анализ документов	Изучение документов
Соответствие стандартам практичности	Тестирование на соответствие			Изучение документов	Изучение документов
Временные характеристики	Тестирование и измерения производительности	Анализ кода	Формальный анализ	Анализ документов	
Использование ресурсов	Тестирование производительности, профилирование	Анализ кода, профилирование	Формальный анализ	Анализ документов	
Соответствие стандартам эффективности	Тестирование и измерения производительности			Анализ документов	
Анализируемость		Анализ кода		Анализ документов	
Изменяемость		Анализ кода		Анализ документов	
Стабильность		Анализ кода		Анализ документов	
Контролируемость		Анализ кода		Анализ документов	
Соответствие стандартам сопровождаемости		Проверка кода		Анализ документов	
Адаптируемость	Тестирование	Анализ кода		Анализ документов	Изучение документов
Устанавливаемость	Тестирование	Анализ кода		Анализ документов	Изучение документов
Способность к сосуществованию	Тестирование	Проверка кода		Анализ документов	Изучение документов
Удобство замены	Тестирование	Анализ кода		Анализ документов	Изучение документов
Соответствие стандартам переносимости	Тестирование на соответствие	Проверка кода		Анализ документов	Изучение документов

Таблица 1. Методы оценки атрибутов качества.

Таблица 1 дает сопоставление известных авторам методов оценки качества ПО с атрибутами качества и различными видами документов. Мы не претендуем на полноту представленной в ней информации, пустые клетки в таблице означают, что нам не удалось вспомнить или найти в литературе и Интернете соответствующие методы.

Эта таблица показывает, что тестирование является практически единственным способом проверки качества ПО, применимым к исполнимому коду — конечному продукту процессов разработки ПО. Только для оценки характеристик производительности кроме тестирования можно использовать другие методы.

Для подавляющей массы коммерческого ПО его исполнимый код и пользовательская документация являются единственными документами, доступными конечным пользователям. Поскольку соответствие между действительными свойствами программ и их описанием в пользовательской документации часто само по себе требует проверки, тестирование оказывается единственным надежным средством оценки качества ПО с точки зрения конечных пользователей. Оно не может быть заменено ни одним из других указанных методов контроля качества, ни каким-либо их сочетанием.

Приведенная аргументация в пользу незаменимости тестирования может быть оспорена сторонниками подхода «корректность по построению» (correctness by design). Этот подход

подразумевает применение таких методов проектирования и разработки, которые автоматически гарантировали бы корректность полученного в результате ПО. К нему тесно примыкают активно распространяющиеся сейчас методы обеспечения качества ПО на ранних этапах разработки.
Ни в коем случае не отвергая необходимости развития подобных методов и полезности их применения на практике, заметим, что все подобные подходы подразумевают, что результат будет качественным только при *правильном* применении соответствующих методов. Исключить людей из процессов разработки ПО в ближайшем будущем возможным не представляется, а людям свойственно делать ошибки даже при выполнении достаточно простой работы. Разработка же ПО является одним из самых сложных процессов, включающем в себя разнородные виды деятельности и, в частности, перевод неформальных требований к ПО в формальные модели и код, написанный на формальных языках. Последнее обстоятельство порождает множество неопределенностей в разработке ПО, способствующих появлению многочисленных и трудноопределимых ошибок.
Даже в том случае, если нужный уровень качества достигнут и в пользовательской, и в проектной документации, и в исходном коде, и подтвержден моделированием, трансляция исходного кода в исполнимый и развертывание результирующего ПО на целевой платформе, в окружении других программ, способны внести множество непредусмотренных дефектов и значительно понизить качество ПО, в его восприятии конечными пользователями.

Заключение

Данная работа показывает, что тестирование является одним из важнейших методов контроля качества ПО, и практически единственным их них, доступным конечным пользователям. Приведенная выше таблица показывает, что ни один из других известных методов, ни какая-либо их комбинация не способны полностью заменить тестирование, поскольку только тестирование позволяет получить оценки качества по исполнимому коду. Этот вывод ни в коем случае не умаляет важности других методов, зачастую способных давать необходимую информацию о качестве ПО на более ранних этапах разработки, когда тестирование не применимо.
Несмотря на активное развитие и внедрение в практику перспективных методов обеспечения качества на ранних этапах разработки ПО, в ближайшем будущем они также вряд ли будут способны сделать тестирование ненужным. Тем не менее, мы считаем, что такие методы имеют большой потенциал и их развитие и практическое использование может дать значительные результаты.
Кроме того, в силу наличия в составе понятия качества ПО характеристик, связанных с восприятием ПО различными заинтересованными лицами, само по себе проведение тестирования способно повысить качество ПО.
Можно также отметить наличие пустых клеток в Таблице 1. Такие клетки могут означать как отсутствие связи между соответствующими атрибутом качества и видом документов, так и отсутствие на данный момент методов, позволяющих оценивать этот атрибут качества по данному виду документов.
Так, например, по мнению авторов, отсутствие на данный момент методов оценки сопровождаемости на основе исполнимого кода ПО не означает невозможности проведения подобных оценок. Более того, такие методы могли бы существенно помочь при сопровождении и перепроектировании унаследованного программного обеспечения, для которого зачастую не сохраняется других документов и даже исходного кода, либо имеющиеся документы не актуальны, не соответствуют той версии ПО, которая находится в работе.

Литература

[1] J. McCall, P. Richards, G. Walters. *Factors in Software Quality*. three volumes, NTIS AD-A049-014, AD-A049-015, AD-A049-055, November 1977.

[2] B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, G. MacLeod, and M. J. Merritt. *Characteristics of Software Quality*. North Holland, 1978.

[3] B. Boehm. *Software Risk Management*. IEEE Computer Society Press, CA, 1989.

[4] International Standard ISO/IEC 9126. *Information technology – Software product evaluation – Quality characteristics and guidelines for their use*. International Organization for Standardization, International Electrotechnical Commission, Geneva, 1991.

[5] В. В. Липаев. *Обеспечение качества программных средств. Методы и стандарты*. Синтег, Москва, 2001.

[6] ISO/IEC 9126-1. 2001. *Software engineering – Software product quality – Part 1: Quality model.* Geneva, Switzerland: International Organization for Standardization.

[7] ISO/IEC DTR 9126-2. 2001. *Software engineering – Software product quality – Part 2: External metrics.* Geneva, Switzerland: International Organization for Standardization.

[8] ISO/IEC DTR 9126-3. 2000. *Software engineering – Software product quality – Part 3: Internal metric.* Geneva, Switzerland: International Organization for Standardization.

[9] L. Hyatt and L. Rosenberg. A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality. *ESA 1996 Product Assurance Symposium and Software Product Assurance Workshop*. European Space Agency, ESTEC, Noordwijk, The Netherlands, pp. 209-212.

[10] R. Grady and D. Caswell. *Software Metrics: Establishing a Company*.Wide Program, Prentice Hall, 1987.

[11] G. Dromey. *Cornering the Chimera*. IEEE Software, January, 1996, pp. 33-43.

[12] R. Bache and G. Bazzana. *Software metrics for product assessment*. McGraw Hill, International Software Quality Assurance Series, 1993.

[13] A. K. Rae, H. L. Hausen, and P. Robert (Editors). *Software Evaluation for Certification: Principles, Practice and Legal Liability*. McGraw Hill, International Software Quality Assurance Series, 1995.