

АНАЛИТИЧЕСКИЕ МЕТОДЫ В УПРАВЛЕНИИ КАЧЕСТВОМ РАЗРАБОТКИ ПРОГРАММНЫХ ПРОДУКТОВ

О.В. Пономарева*, А.А. Дубаков

Томский политехнический университет

*ООО «Сибирские информационные системы», г. Томск

E-mail: ponom-olga@yandex.ru

Предложена методология контроля качества производимого программного продукта, основанная на исследовании аналитических методологий. Проанализированы наиболее подходящие положения, применимые к деятельности IT-компании, и адаптированы для работы отдела Quality Assurance (QA). Основной идеей предложенного метода является объединение работы аналитика и инженера QA, использование аналитических методов, применительно к сфере Quality Assurance, привлечение аналитиков к тестированию, а инженеров QA – к системному анализу и к процессам принятия решений. На основе аналитического метода Quality Function Deployment (QFD) разработан собственный метод, помогающий принять решение о вступлении компании в проект. Сделан вывод об эффективности предложенной методологии.

Ключевые слова:

Контроль качества, аналитические методы, жизненный цикл программного обеспечения, принятие решения.

Главным критерием, определяющим успешность проекта, является качество полученного продукта. Определение понятия «качество продукта» можно сформулировать как отсутствие дефектов и соответствие ряду требований. Качественный программный продукт должен обладать определенными свойствами: завершенностью (готовностью к непосредственной работе), расширяемостью, возможностью повторного использования, простотой, удобством использования и т. д. Существует достаточно способов для измерения качества. Данной задачей занимается отдел Quality Assurance (QA) компании.

Для повышения эффективности процесса разработки программного обеспечения (ПО) необходимо обращать внимание на реализуемые проекты: со стороны исполнителя следует быть более требовательным к принимаемым на реализацию проектам, т. е. требовательным и избирательным по отношению к заказчику, т. к. распространенным является тот случай, когда в компании один прибыльный проект спонсирует десятки убыточных. С другой стороны, в связи с обострившейся конкуренцией на рынке IT-услуг, следует повышать уровень качества ПО, производимого в компании, чтобы быть конкурентоспособными.

Разработанный набор рекомендаций позволит оптимизировать процесс контроля качества. Одним из решений задач повышения качества, которые ставит современный рынок IT-услуг, является изменение принципа работы отдела QA. Если раньше основной упор делался на поиск и исправление ошибок, то сейчас следует занять позицию тотального контроля качества как в методологии Total Quality Management (TQM) с целью недопущения ошибок, которые пришлось бы отлавливать и исправлять.

Идея создания метода основывается на исследованиях Хиротака Такеучи и Икудзиро Нонака, их методе управления проектами Scrum, демонстрирующем, что проекты, над которыми работают небольшие кросс-функциональные команды систематически выдают лучшие результаты [1].

Методология Plan-Do-Check-Act (PDCA), предлагает объединение этапов исследования, проектирования и производства. Эта идея была частично воплощена в предлагаемой методологии.

Основной идеей предложенной методологии является **объединение работы аналитика и инженера QA, использование аналитических методов применительно к сфере QA, привлечение аналитиков к тестированию, а инженеров QA – к системному анализу и к процессам принятия решений.**

У аналитика и инженера QA есть ряд задач, которые гораздо удобнее решать совместно. Такой подход имеет ряд преимуществ. Вот лишь некоторые из них:

1. Совместное проектирование системы позволит инженеру QA заранее предусмотреть возможные сложности в обеспечении качества системы.
2. Совместное составление документации обеспечит тестирование всей проектной документации на ранних сроках разработки.
3. Помощь аналитика в тестировании выявит ошибки проектирования логики системы.
4. Инженер QA, участвовавший в разработке системы, более осведомлен о ее особенностях и возможных недостатках, поэтому будет в состоянии разработать более качественный план тестирования.
5. Методы, применяемые аналитиками в своей работе, могут быть полезными для обеспечения контроля качества.
6. Совместная работа аналитика и инженера QA делает данных специалистов в определенной мере взаимозаменяемыми.
7. Инженер QA является представителем заказчика на стороне исполнителя, поэтому может дать ценные рекомендации по проектированию интерфейсов и пользовательских функций с точки зрения удобства конечного пользователя.

8. Участие инженера QA в принятии проектно важных решений позволит избежать возможных затрат на обеспечение качества данных решений.

Проще всего представить разработанный метод, рассматривая жизненный цикл (ЖЦ) производства ПО.

В предложенной методологии работа инженера QA начинается не на этапе тестирования (как предлагается в технологии Rational Unified Process (RUP) [2]), а на этапе постановки проблемы, т. е. еще до начала работы проекта, следует на протяжении всего ЖЦ и тесно связана с работой аналитика. В результате работы было установлено, что совместная работа аналитика и инженера QA значительно улучшает результат работы всех процессов ЖЦ и способствует повышению качества производимого ПО. Следует добавить в ЖЦ этап по принятию решения о вступлении в проект, чтобы избежать неликвидных проектов.

Целью работы стоит разъяснение основных принципов разработанного метода, поэтому для более ясного понимания будем рассматривать упрощенный вариант ЖЦ разработки ПО, полагая, что в реальной ситуации, схема гораздо сложнее и требует большего анализа с применением многих аналитических методов (в равной мере уникальных для каждого проекта и универсальных для описываемого метода в принципе).

В табл. 1 изображен измененный жизненный цикл разработки ПО, представленный относительно работы по принципу «Аналитик + инженер QA» в соответствии с методологией.

Таблица 1. Распределение объема работ между аналитиком и инженером QA в процентном соотношении от общего объема проектных работ.

Этапы жизненного цикла ПО	Аналитик, %	Инженер QA, %
Постановка проблемы, принятие решения	60	40
Планирование	60	40
Системное и программное проектирование	50	50
Кодирование, тестирование, отладка	50	50
Системное документирование	50	50
Сдача в эксплуатацию	30	70

На этапе принятия решения о вступлении в проект рассмотрим пример использования аналитических методов, иллюстрирующий тот факт, что применение соответствующих аналитических методов к сфере QA оптимизирует работу компании на каждом этапе.

Ошибки, допущенные на ранних этапах разработки, обходятся исполнителю и заказчику очень дорого.

Прежде, чем браться за реализацию проекта, стоит оценить его ликвидность. Помимо анализа, проводимого финансовыми и аналитическими отделами компании, отделу контроля качества следует оценить проект в плане затрат на обеспечение качества продукта.

На основе метода QFD, используемого аналитиками при преобразовании пожеланий заказчика в требования к системе, был разработан собственный метод относительной оценки затрат на реализацию пользовательских характеристик относительно QA. Данный метод можно применять и к уже реализуемым проектам, когда встает вопрос о закрытии проекта по причине его убыточности.

Суть метода заключается в следующем. Составляется матрица вида, представленного в табл. 2.

В строках перечисляются пользовательские характеристики (ПХ), заявленные клиентом на реализацию (следует понимать, что пользовательские характеристики – это требования, которые формирует заказчик, т. е. ожидания заказчика), в столбцах – исполнитель вносит мероприятия или технические средства (МиТС), обеспечивающие качество данных характеристик (в том числе и связанные с тестированием). На пересечении строк и столбцов проставляется величина ставки пользовательской характеристики R (*Rate*) по 5-бальной шкале, отражающая зависимость характеристики от предпринятых мер (т. е. насколько сильно повлияет на обеспечение качества данной характеристики то или иное мероприятие) по принципу: чем сильнее зависимость – тем больше R .

Таблица 2. Матрица пользовательских характеристик и технических средств

ПХ \ МиТС	Написание программы по изменению времени отклика	...
Время отклика ≤ 1 с	$R_1=3$	$R_2=...$
...

Каждая позиция в столбце мероприятий имеет так же свой рейтинг P (*Price*) по 10-бальной шкале в соответствии со степенью сложности реализации или стоимостью проведения данного мероприятия (заполняется исполнителем) по принципу: чем сложнее и дороже позиция – тем выше рейтинг.

Считается величина стоимости пользовательской характеристики C (*Cost*) по формуле:

$$C_i = P_N \sum_{i=1}^N R_i P_i$$

построчно, чтобы узнать итоговую относительную стоимость каждой пользовательской характеристики, заявленной заказчиком.

Для данной матрицы считается максимальная стоимость пользовательской характеристики $C_{\max} = R_{\max} P_{\max} N$, где $R_{\max} = 5$, $P_{\max} = 10$, формула приобретает вид:

$$C_{\max} = 50N.$$

На основе результатов, полученных при подсчете всех характеристик можно сделать вывод о том, сможет ли компания гарантировать качество требуемого продукта. При большинстве $C_N \rightarrow C_{\max}$ проект считается неликвидным с точки зрения QA, т. к.

требует больших затрат, и такой проект следует брать на реализацию только в случае, если он гарантированно окупается по мнению финансистов, и выпускаемый продукт не имеет аналогов на осваиваемом рынке по мнению аналитиков [3].

Конечно, глубокий анализ проводится аналитиками при планировании системы, но следует подчеркнуть, что представленный выше метод охватывает именно анализ разработки проекта с точки зрения QA, а эту область аналитики, как правило, не продумывают при принятии решения об участии в проекте.

Благодаря проведенному анализу удастся отсеять убыточные проекты и разгрузить отдел контроля качества для более прибыльных работ, где помощь работников будет более эффективной для всего процесса реализации.

Как было сказано выше, данный метод оценки относительной стоимости пользовательских характеристик демонстрирует ситуацию, как аналитические методы применяются в сфере QA, и как сфера QA важна в принятии решения о вступлении в проект или о его продолжении.

Аналогично методологии экстремального программирования, контроль качества осуществляется на каждом этапе разработки. А именно, если было принято решение о начале реализации проекта, то, при планировании работ, в финансовый план и в график реализации следует заложить затраты по исправлению ошибок, т. к. дефекты в любом случае неизбежны. Необходимо заранее учесть в распределении средств и времени этот факт [4].

Системное и программное проектирование требует внимания инженера QA, т. к. специалист QA является представителем заказчика на стороне исполнителя и более заинтересован в детальной проверке. На данном этапе следует осуществлять вставку процедур проверки качества в модель, а также подготовить всю тестовую документацию и одобрить ее с заказчиком, чтобы избежать затем на этапе приемки временных и финансовых потерь из-за несоответствия качества полученного продукта (в документации описываются границы контроля качества продукта, т. е. оговаривается, какие функции будут проверяться в рамках договора). Работа аналитика в паре с инженером QA не заканчивается с окончанием этапа системного анализа, т. к. требования заказчика, как правило, нечеткие в начале работы. Но, при этом, требования к продукту на этапе приемки очень строгие и вполне конкретны. Аналитикам при сопровождении процесса разработки необходимо оперативно реагировать на комментарии отдела QA. В большинстве случаев требования инженера QA совпадают с требованиями заказчика при приемке.

На этапах написания программного кода следует строить работу таким образом, чтобы документация, составленная аналитиками, являющаяся постановкой задачи для программистов (специфика-

ция, техническое задание и т. д.), была протестирована специалистами из отдела QA для выявления скрытых в ней ошибок. Ошибки и неточности в формулировках могут повлечь за собой вольность интерпретации требований к системе, и как результат – неверно спроектированную систему. Инженеру QA следует заранее ознакомиться со всей документацией, чтобы вовремя спроектировать средства тестирования, подготовить набор тестовых заданий. Инженер QA, используя документацию к проектируемой системе, может заранее оценить трудоемкость и стоимость гарантии качества системы.

На этапе тестирования роль инженера QA и тестировщика традиционно заключается в контроле за техническим качеством системы, так же требуется работа аналитика по тестированию логики системы и соответствию предметной области. Инженер QA не обязан быть специалистом в предметной области, с которой связан проект, а аналитик занимается изучением специфики проекта. Но, у работников, контролирующих качество, есть опыт «универсального отлова» дефектов в любой системе. Поддержка аналитика необходима при конфигурировании автоматических средств тестирования, что немаловажно в мероприятиях QA (регрессионное тестирование, предреализационная подготовка и т. д.). Метод декомпозиции, используемый аналитиками в формировании «каркаса» проектируемой системы, может быть применен в процессе тестирования. Удобнее осуществлять тестирование системы в три этапа:

1. Поверхностное (с использованием специального документа – *check list*).
2. Позитивное (на данных, удовлетворяющих условиям нормальной работы системы, с целью проверки работы сценария в нормальных рабочих условиях).
3. Негативное (на данных, не удовлетворяющих условиям нормальной работы системы, а также на пограничных данных, с целью выявить условия, при которых система дает сбой в работе).

Особое внимание следует уделить позитивному тестированию, т. к. оно эмитирует работу конечных пользователей.

Если следовать предложенной методологии, то на этапе системного документирования аналитик и инженер QA являются наиболее осведомленными работниками проекта, т. к. участвовали в реализации каждого этапа разработки. Это значит, что написание документации следует доверить аналитикам и специалистам QA, тем более, что, благодаря контролю на всех этапах реализации, работники отдела QA менее загружены, по сравнению с обычной ситуацией, когда работа по контролю качества начиналась на этапе тестирования и отладки.

Предложенный метод основан на исследовании методологий RUP, TQM, Scrum, QFD, PDCA, ме-

тодологии экстремального программирования, метода декомпозиции. Он позволяет существенно снизить затраты на обеспечение качества продуктов. Данный метод оправдал себя в долгосрочных проектах (длительностью более 1 года), где более заметна зависимость применяемой методологии от затрат. Метод помог снизить риск компании связать себя обязательствами по отношению к убыточным проектам на этапе принятия решения о вступлении в проект. Контроль качества на каждом этапе реализации проекта снижает себестоимость проекта, т. к. позволяет отслеживать ошибки на ранних этапах, когда их стоимость значительно ниже по сравнению с ошибками, выявленными на этапе тестирования и последующих этапах.

Суть предложенного метода можно кратко изложить как объединение работы аналитика и инженера QA (возможна даже ситуация для неболь-

ших компаний, чтобы это был один человек) на каждом этапе жизненного цикла разработки программного обеспечения. Аналитик и инженер QA обладают большим запасом знаний и опытом, позволяющим осуществлять контроль за качеством системы на всех уровнях: от логики системы до опечаток программистов. Поэтому, начиная с момента принятия решения о начале реализации проекта до сдачи системы в эксплуатацию, инженер QA и аналитик должны работать в тесном сотрудничестве с обменом ролями и дополнением обязанностей друг друга. Внедрение метода предполагает определенное время на перестроение работы аналитического отдела и отдела контроля качества, а также на адаптацию многих аналитических методов к процессам контроля качества, но результаты, выраженные в финансовой и временной экономии, оправдывают такие затраты.

СПИСОК ЛИТЕРАТУРЫ

1. Schulmeyer G.G. Handbook of Software Quality Assurance. – Norwood: Artech House, 2008. – 485 p.
2. Поллис Г., Огастин Л. Разработка программных проектов на основе Rational Unified Process (RUP). – М.: Бином, 2009. – 256 с.

3. Брагин Ю.В. Путь QFD: проектирование и производство продукции исходя из ожиданий потребителей. – М.: Центр качества, 2003. – 240 с.
4. Бек К. Экстремальное программирование: разработка через тестирование. – СПб.: Питер, 2003. – 224 с.

Поступила 13.04.2009 г.

УДК 004.623

ТЕХНОЛОГИЧЕСКИЕ АСПЕКТЫ ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ ХРАНЕНИЯ И ОБРАБОТКИ БОЛЬШИХ ОБЪЕМОВ ДАННЫХ

А.В. Алубин, В.В. Грачев, С.А. Матвеев, О.Ф. Юдин, М.А. Сонькин*

ФГУП «НИИ «КВАНТ», г. Москва

*Томский политехнический университет

E-mail: sonkin@tpu.ru

Определены основные подходы к оценке производительности, проведен сравнительный анализ и протестированы системы хранения данных ведущих производителей.

Ключевые слова:

Системы хранения данных, системы обработки информации, вычислительные модули, локальная вычислительная сеть, производительность.

В настоящее время отсутствует общепринятая методика оценки производительности систем обработки информации (СОИ) специального назначения, что связано в первую очередь с отсутствием единиц для измерения количества вычислительной работы. Поэтому для оценки производительности используется широкая номенклатура величин — показателей производительности, которые и в отдельности, и в совокупности не удовлетворяют в полной мере потребностям теории и практики проектирования и эксплуатации СОИ.

Рассмотрим СОИ, состоящую из совокупности вычислительных модулей (ВМ) объединенных в локальную вычислительную сеть (ЛВС), причем каждый из ВМ занимается решением только одной выделенной задачи специального назначения. Структурная схема такой СОИ представлена на рис. 1.

Основными функциями СОИ могут быть:

- автоматическая обработка принятой информации;
- ведение архивов принятой и обработанной информации;