Определение понятия качества программного обеспечения относительно WEB-приложений

Прохоров А.С.

Международная академия образования

Что обычно подразумевается под термином «качество программного обеспечения»? Одним из примеров модели качества программного обеспечения является предложенная Hewlett-Packard идеология FURPS (функциональность, юзабилити, надежность, производительность, поддерживаемость)[1]. Хотя модель FURPS применима ко всем видам программного обеспечения, ее можно дополнить атрибутами качества, относящимися к веб-приложениям: находимость, доступность и правовое соответствие[2]. Можно согласиться с точкой зрения Нила Бевана, определяющего наличие разницы между внешним и внутренним качеством программ[3].

Внешнее качество

Клиенты или конечные пользователи приложения, сосредотачивают свое внимание на тех аспектах достоинства программы, которые имеют для них существенное значение. Согласно мнению Себастьяна Бергмана, следующие аспекты являются показателями внешнего качества приложений[4]:

- Функциональность означает, что приложение фактически может выполнять предлагаемые задачи.
- Юзабилити означает, что пользователь может работать эффективно, удобно и при этом еще получать удовлетворение от работы приложения. Доступность является частью юзабилити.
- Ре-активность (reactivity) подразумевает короткое время отклика программы. Это особенно важно, если производитель желает, чтобы пользователь был удовлетворен работой программы.
- Безопасность означает, что программа должна обеспечивать необходимый уровень защиты данных, как пользователя, так и самого кода приложения.
- Доступность и надежность особенно важны для веб-приложений с большим количеством пользователей. Приложения должны быть способны выдерживать большие нагрузки и обязаны работать даже в нестандартных ситуациях.

Все аспекты внешнего качества могут быть проверены путем тестирования приложения в целом, с использованием, так называемых, «непрерывных» (end-to-end) тестов. Применение тестирования позволяет в автоматическом режиме проверять, что программный продукт выполняет все свои функциональные требования. Еще на этапе планирования мощностей, разработчики и администраторы должны определить потенциально опасные места в приложении, если вдруг изменится программное окружение или увеличится трафик.

Внутреннее качество

Внутренне качество определяется нуждами разработчиков и администраторов приложения. Разработчики в первую очередь фокусируются на читаемости кода, который легко понять, адаптировать и расширять. Если они не сделают этого, реализация последующих требований клиента становится более сложной и, следовательно, более дорогой с течением времени. Это повышает риск того, что даже небольшие изменения в программном обеспечении могут привести к неожиданным побочным эффектам.

Внутреннее качество программного обеспечения практически незаметно для клиентов и конечных пользователей. Конечные пользователи ожидают, что программа будет удовлетворять всем или, по крайней мере, большинству их функциональных ожиданий, а также быть простой в использовании. Фактически, если при использовании продукт работает достаточно быстро, большинство клиентов будут удовлетворены.

Однако недостатки внутреннего качества программы проявляются во временной перспективе. Требуется больше времени для исправления даже тривиальных ошибок. Любые изменения или дополнения к программному обеспечению требует огромных усилий. В итоге, разработчик рано или поздно начинает просить увеличить бюджет на чистку и рефакторинг кода. А так как клиенты или начальство обычно не понимают в чем польза рефакторинга, эти запросы чаще всего остаются без внимания.

Основная цель обеспечения качества, а точнее, управления качеством программного приложения, – сделать затраты и выгоды прозрачными для всех сторон, участвующих в создании ПО. Низкое внутреннее качество программы в долгосрочной перспективе вызывают дополнительные затраты. Если эти затраты могут быть определены количественно, то нужно сделать вывод, что достижение лучшего внутреннего качества приведет к снижению затрат. Это, видимо, единственный способ заставить начальство или клиента рассмотреть вопрос о выделении бюджета для рефакторинга кода._____

[1] Robert Grady and Deborah Caswell, Software Metrics: Establishing a Company-wide Program.

[2] Klaus Franz, Handbuch zum Testen von Web-Applikationen.

[3] Nigel Bevan, “Quality in use: Meeting user needs for quality,” Journal of Systems and Software 49, Issue 1.

[4] Sebastian Bergmann, Stefan Priebsch - Real-World Solutions for Developing High-Quality PHP Frameworks and Application