Понятие качества. Основные показатели качества деятельности предприятия

Ключевым фактором обеспечения эффективного применения программных продуктов является тщательное оценивание и достижение высоких значений показателей качества. Постоянный рост сложности функций, реализованных программами в информационных системах, непосредственно приводит к увеличению их объема и трудоемкости их создания. Согласно изменениям сложности программ возрастает количество найденных и оставленных в них дефектов и ошибок, что отражается на их качестве. Программа объемом в миллионы строк кода в принципе не может быть безошибочной. Проблема нахождения и устранения ошибок обостряется по мере увеличения сложности задач и программ, которые их решают, и угрожает катастрофами в областях, которые используют программное обеспечение (ПО) [1-3].

Проблема заключается в том, чтобы обеспечить нужное качество функционирования ПО с учетом того, что некоторое неизвестное количество ошибок и дефектов всегда остается в сложных комплексах программ, и должно быть блокировано или сокращено их негативное воздействие до допустимого уровня. В связи с этим стратегическая задача в жизненном цикле современного ПО - обеспечение качества программных продуктов [4].

Качество ПО определяется качеством методов и инструментальных средств, которые применялись для обеспечения всего их жизненного цикла. На практике важно оценивать качество программ не только в завершенном виде, но и в процессе их проектирования и разработки.

Сегодня ПО является определяющей составляющей многих систем, среди которых системы критического применения, встроенные и специализированные системы различного назначения. Для указанных систем наличие ошибок и низкое качество ПО грозят катастрофами, которые приводят к человеческим жертвам, экологическим катаклизмам, экономическим потерям. Развитие современных технологий разработки ПО требует динамичного развития средств оценки качества ПО, причем уже на этапе проектирования (с точки зрения экономической и временной целесообразности).

Современная индустрия ПО характеризуется высокой конкуренцией. Для успешной работы на этом рынке софтверная компания должна разрабатывать, внедрять и сопровождать ПО быстро, укладываясь в сроки, и с удовлетворительным качеством. Поэтому многие софтверные компании вкладывают средства в модернизацию технологий разработки ПО, помня о том, что такое вложение средств обязательно окупается.

На этапе проектирования важно заложить целый ряд требований по качеству: требования к структуре программы, требования к навигации по ПО, требования к дизайну интерфейсов пользователя, требования к мультимедиа-компонентам ПО, требования к удобству применения (usability), технические требования.

На этапе проектирования формируется ответ на вопрос "Каким образом программное обеспечение будет реализовывать выдвинутые к нему требования?".

Информационные потоки этапа проектирования ПО [5]: требования к ПО, представленные информационной, функциональной и поведенческой моделями анализа. Информационная модель описывает информацию, котрую должно обрабатывать ПО по мнению заказчика. Функциональная модель определяет перечень функций обработки информации и перечень модулей ПО. Поведенческая модель фиксирует желаемую динамику системы (режимы ее работы). На выходе этапа проектирования - разработка данных, разработка архитектуры и процедурная разработка ПО.

Для получения оценки значений показателей качества по стандарту [6] используются следующие методы:

1. измерительный - базируется на использовании инструментальных, измерительных и специальных программных средств для получения информации о свойствах и характеристиках ПО (объем ПО, количество строк кода, количество операторов, число ветвей в программе, количество точек входа / выхода и др.);
2. регистрационный - основан на получении информации во время испытаний или функционирования ПО, когда регистрируются или подсчитываются определенные события (время и количество сбоев и отказов, время передачи управления от одного модуля к другому, время начала и завершения работы ПО);
3. органолептический - основан на использовании информации, полученной в результате анализа восприятий органов чувств, и применяется для определения таких показателей как удобство применения, эффективность;
4. расчетный - основан на использовании теоретических и эмпирических зависимостей (на ранних этапах разработки), статистических данных, собранных при проведении испытаний, эксплуатации и сопровождении ПО. Расчетными методами оцениваются показатели надежности, точности, устойчивости, время реакции, необходимые ресурсы и проч.;
5. экспертный - осуществляется группой экспертов. Их оценка базируется на опыте и интуиции, а не на непосредственных результатах расчетов или экспериментов. Этот метод реализуется путем просмотра программ, кодов, сопроводительных документов, описаний требований к ПО группой экспертов. Для этого устанавливаются контролируемые признаки, коррелированные с одним или несколькими показателями качества и включенные в карты опроса экспертов. Метод применяется при оценке таких показателей, как наглядность, полнота и доступность программной документации, легкость усвоения, возможность анализа, документированность, структурированность ПО и др.

Из описания методов измерения показателей (метрик) качества понятно, что на этапе проектирования ПО невозможно измерить ни одной характеристики еще не разработанного ПО, невозможно регистрировать моменты процесса выполнения еще не существующего ПО и невозможно воспринять органами чувств информацию о не разработанном ПО. Так, на этапе проектирования является возможность определять качество ПО только с применением расчетных и экспертных методов.

Согласно [7], метрика определяется как мера степени владения свойством, которая имеет числовое значение. Вообще, метрика ПО – это мера, которая позволяет получить числовое значение некоторого свойства программного обеспечения или его спецификаций. Итак, если речь идет о метрики сложности программ, то это меры, которые дают числовое значение сложности ПО, а метрики качества программ дают числовое значение качества ПО.

Показатели оценки сложности и качества проекта очень важны для получения объективных оценок по проекту. Как правило, данные показатели не могут быть вычислены на ранних стадиях работы над проектом, поскольку требуют, как минимум, детального проектирования. Однако эти показатели важны для получения прогнозных оценок длительности и стоимости проекта, поскольку непосредственно определяют его трудоемкость. Несмотря на многочисленные исследования программных метрик, в этой области остается ряд нерешенных вопросов:

1. технология измерения качества еще не достигла зрелости - только 1.5% софтверных организаций пытаются оценить качество процессов и готового продукта количественно, с помощью метрик, и только 0.5% софтверных организаций пытаются улучшить работу, руководствуясь количественными критериями качества для выпуска бездефектных продуктов;
2. отсутствуют единые стандарты на метрики - создано более тысячи метрик, каждый поставщик "измерительной" системы предлагает свои способы оценки качества и соответственно метрики;
3. существует проблема сложности интерпретации величин метрик - значения метрик, полученные с помощью "измерительных" систем, неинформативны или мало информативны для пользователя, заказчика, а часто и для программиста;
4. метрики рассчитываются только для готового ПО – все "измерительные" системы ориентированы на расчет метрик программного кода, но часто существует необходимость в расчете метрик уже на этапе проектирования - метрик сложности и качества с точными значениями для проекта ПО и метрик сложности и качества с прогнозируемыми значениями для разрабатываемого по проекту программного обеспечения.

Именно из-за нерешенности этих вопросов пока невозможно создать бездефектное высококачественное ПО. Сложность обоснования выбора и интерпретации метрик в процедурах принятия производственных решений, а также игнорирование этапов жизненного цикла ПО не позволяют полноценно использовать метрики для повышения качества ПО и уменьшения его сложности.

<http://elar.khnu.km.ua/jspui/bitstream/123456789/1515/1/ISOP_rus.pdf>

#TODO вставить источники