

УДК 65.018

The Introduction of Quality Management System in the Innovative IT- Company

Alexander K. Moskalev^{*a} and Ksenia S. Lugovaya^b

^a*Siberian Federal University*

79 Svobodny, Krasnoyarsk, 660041, Russia

^b*Ltd «Alfateam»*

20B Bauman, Krasnoyarsk, 660028, Russia

Received 14.07.2015, received in revised form 26.10.2015, accepted 24.11.2015

This paper is devoted to the results of comparison of the software produced by “Svyazkom Ltd.” the company developing software in the area of mobile communication and the Internet. The software was designed by several specialists at different time. It is shown that the application of the computer analysis permits to verify 86 indicators relevant to the quality parameters in the shell CppDepend, which meets most adequately the requirements of the assigned task. As the result of the analysis the optimized profile of the product was formed. The proposals for updating the mechanisms of quality management in the innovative IT- company based on the obtained values of quality metrics are presented.

Keywords: quality products, profile best programs, quality system in it companies, update responsibility matrix.

DOI: 10.17516/1999-494X-2015-8-8-1117-1126.

Внедрение системы менеджмента качества в инновационной IT-компании

А.К. Москалев^a, К.С. Луговая^b

^a*Сибирский федеральный университет*

Россия, 660041, Красноярск, пр. Свободный, 79

^b*ООО «Альфатим»*

Россия, 660028, Красноярск, Баумана, 20б

В статье выполнено сравнение качества программного продукта, создаваемого фирмой-разработчиком в области мобильной связи и Интернета ООО «Связьком» в разное время несколькими авторами. Показано, что использование компьютерной технологии анализа по 86 метрикам, соответствующим параметрам качества в оболочке Cppdepend, наиболее адекватно поставленной задаче. В результате анализа дан оптимизированный профиль

продукта. Изложены предложения по актуализации механизмов управления качеством в инновационной IT-компании с учетом полученных значений метрик качества.

Ключевые слова: качество программного продукта, профиль лучшей программы, система качества в IT-компании, актуализация матрицы ответственности.

Введение

Рынок современной индустрии программного обеспечения характеризуется весьма высокой степенью конкуренции, и для успешной работы IT-компания должна быстро разрабатывать, внедрять и сопровождать программное обеспечение соответствующего качества. Как следствие, предприятию необходимо создавать адекватные ситуации системы улучшение процесса управления системой качества. Статистика показывает, что улучшение процессов разработки программного обеспечения в успешных компаниях приводит к значительному повышению производительности и качества со средним уровнем возврата вложений до 8:1 [1]. В этой связи в научных исследованиях и в практике проявляется неизбежность разработки объективных показателей для оценки способностей компании производить продукцию с необходимыми качественными характеристиками, подтвержденные сертификатом соответствия.

Потребность в создании новых программных решений вызвана рядом причин. Одной из них выступает создание программных средств в сфере автоматизации производственных или коммерческих проектов. Разработка программного обеспечения в промышленных масштабах – это высокотехнологичный процесс, обеспечивающий единую схему работы большого количества разработчиков и дающий гарантии соответствия системы ожиданиям заказчика. Известно, что качество программного обеспечения – это совокупность свойств, характеризующих способность программного обеспечения удовлетворять потребностям пользователя в соответствии с предназначением [24]. Проблема обнаружения и устранения ошибок обостряется по мере увеличения сложности задач, решаемых программистами, и грозит катастрофами в информационных системах, выполняющих критические объекты или процессы [5].

Материалы и методы

Для измерения качества программного средства применяют метрики. Метрика программного обеспечения – это мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций [1]. В настоящее время в мировой практике используют несколько сотен метрик. Однако погоня за их универсальностью, игнорирование области применения разрабатываемого программного обеспечения существенно снижают эффективность их использования. При выборе метрики необходимо помнить, что она должна давать возможность отслеживать тенденцию изменений и иметь смысл как для заказчика, так и для исполнителя.

Контроль кода проверяющим лицом, отличным от его разработчика, может осуществляться вручную. Оценка кода включает в себя соответствие кода требованиям, предъявляемым к структурированию и разделению на модули, полноту и качество документирования кода, включая документирование заголовков программных модулей, прототипов функций и структур данных, комментарии к выполнению существенных операций, соответствие алго-

ритмов, реализованных в исходном коде, программной документации. Выполнение такого рода контроля – длительный и поверхностный подход, неспособный произвести расчёт метрик. Применение методов и средств автоматизированного анализа исходного кода упрощает эту задачу.

Имеется большое количество коммерческих и бесплатных статических анализаторов кода, таких как CandC++ CodeCounter, SourceMonitor, Reflector.CodeMetrics, FxCop, CppDepend, ParasoftC/C++test и т.д. Список языков, для которых существуют статические анализаторы кода, также достаточно велик (Си, Си++, С#, Java, Ada, Fortran, Perl, Ruby, ...). Подробности можно найти в монографии [1] либо на сайтах организаций – производителей соответствующего программного продукта [610]. Важно понимать, что нет идеального инструмента тестирования программ, необходимо найти тот, который больше всего подходит под определенную специфику, язык программирования и для решения поставленных задач.

Поэтому цель работы, результаты которой представлены в данной публикации, – выделить компьютерную программу автоматизированного анализа качества программного продукта и на основе главных метрик актуализировать матрицу ответственности в IT-компании.

ООО «Связьком» – разработчик инновационных решений и сервисов в области мобильной связи и Интернета, а также разработчик платформенных VAS-решений и специализированного программного обеспечения для операторов сотовой связи и сервис-провайдеров [11].

Деятельность компании разделена на два направления:

- разработка и распространение мобильного контента (картинки, мелодии, java-игры, видео, темы и др.);
- разработка платформенных VAS-решений и специализированного программного обеспечения для операторов сотовой связи и сервис-провайдеров.

Выполненный нами ABC-анализ крупных проектов организации показал, что проекты группы А занимают большую долю среди всех проектов (43 %) и составляют 79 % от всей выручки. Число проектов группы С в процентном соотношении невелико по сравнению с остальными (20 %), это означает, что менее важных проектов у компании практически нет, они равны 5 %. Все проекты приносят хороший процент в суммарную выручку компании. Наиболее выгодным и экономически эффективным проектом является проект Galaxy. Сейчас проект, по словам создателей [8], превратился в мобильную социальную сеть, причем проект «Галактика знакомств» изначально формировался, ориентируясь на пользователей мобильных телефонов. На текущий момент количество зарегистрированных пользователей проекта более 10 000 000.

Результаты

Для проведения анализа качества программной продукции компании необходимо было рассмотреть и выбрать наиболее подходящий инструмент анализа кода. Таким инструментом стал CppDepend, так как он отлично подходит для цели работы и является уникальным и обширным программным средством, содержащим в себе 86 метрик.

Оценка качества кода программного обеспечения, создаваемого в ООО «Связьком», проводилась на основе трех проектов, написанных в разное время разными программистами, но с одной целью. Комплекс модулей программного кода был написан для реализации платформы «Связьком.SmartConnect».

Платформа «Связьком.SmartConnect» предоставляет комплекс услуг для стимуляции голосового трафика в случаях, когда возникают какие-то проблемы, мешающие соединению абонентов (недоступность абонента, занятость абонента, абонент не заметил вызов, у вызывающего абонента недостаточно средств и др.). Платформа «Связьком.SmartConnect» принимает вызовы абонентов и выполняет их обработку в соответствии с параметрами переадресации, настройками в профилях вызывающего абонента и принимающего вызов.

В комплекс услуг в стандартной поставке входят:

- MissedCallsAlert («Пропущенные вызовы», МСА) — для рассылки SMS-оповещений о пропущенных вызовах и о появлении в сети ранее недоступного абонента;
- SmartVoiceMail («Голосовая почта», VM) — позволяет оставлять для недоступного абонента голосовые сообщения, которые он затем может прослушивать.

Анализ качества программного обеспечения в динамике, т.е. анализ проектов по функциональному назначению схожих, но написанных за разные промежутки времени и разными программистами, показал, что качество написанных программных модулей улучшается и упрощается их интеграция в другие проекты.

Произведен анализ качества программного кода одного из главных модулей платформы «Связьком.SmartConnect», создаваемого фирмой-разработчиком в области мобильной связи и Интернета ООО «Связьком» в разное время несколькими авторами.

Визуализированным представлением метрик кода служит матрица, которая в зависимости от вида метрики показывает величину всех составляющих проекта в сравнении. CppDepend использует TreeMap – метод отображения данных с древовидной структурой с помощью вложенных прямоугольников. Данный метод используется для отображения иерархии проекта. При наведении на определенный участок матрица, структура проекта, будь то тип, метод или класс (это зависит от выбора параметров отображения), подсвечивается другим светом на всей матрице в зависимости от того, где он вызывается другими структурами.

Ниже приведены скриншоты матриц трех обсуждаемых программ из-под CppDepend.

Первый проект был написан четыре года назад и работал нестабильно. На рис. 1 показаны относительные объемы каждого модуля и входящих в него методов и типов программы, об этом говорит ячеистая структура. Сравнение идет по количеству строк.

Данная матрица наглядно демонстрирует абсолютный и относительный объемы всех структур программы. Распределение объемов однородное. Необходимо помнить, что слишком большое количество мелких структур приводит к усложнению понимания программного кода и увеличивает вероятность появления ошибок.

Таким же образом исследовали второй проект, который был написан два года назад тем же программистом. Матрица проекта изображена на рис. 2.

Данная матрица наглядно демонстрирует абсолютный и относительный объемы всех структур программы. Распределение объемов неоднородное: libcorenproto и libvoice_grpc_grproto имеют гораздо большие объемы, чем остальные структуры.

Сравнивая два проекта, можно сделать вывод о том, что первый проект состоит из большего количества мелких структур, чем второй.

Третий проект, написанный полгода назад другим программистом, отличается и по качеству кода, и по логической структуре. Главные метрики проекта представлены на рис. 3.

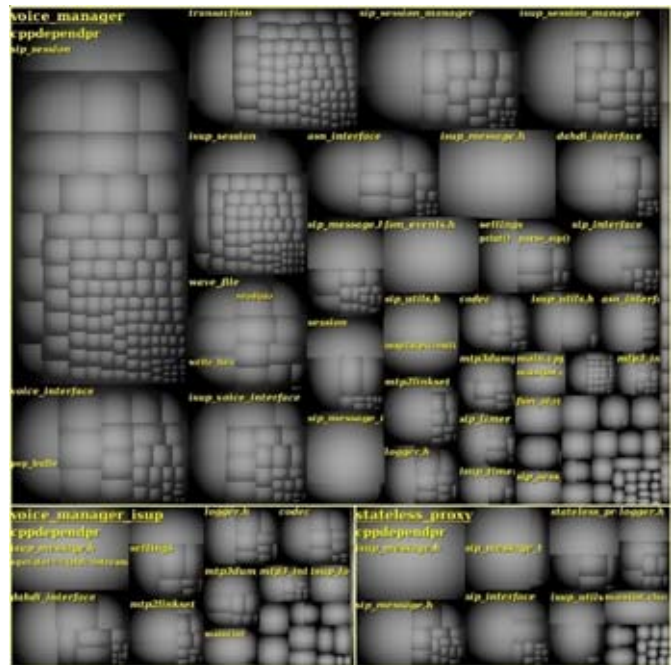


Рис. 1. Матрица проекта № 1

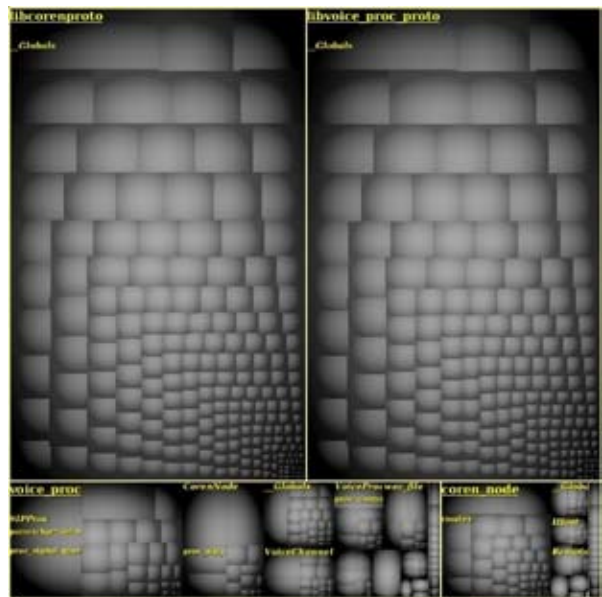


Рис. 2. Матрица проекта № 2

Данная матрица наглядно демонстрирует абсолютный и относительный объемы всех структур программы. Распределение объемов неоднородное.

В разделе качества кода, который находится в сгенерированном CppDependом отчете, у данного проекта есть одно критическое и двенадцать средней степени предупреждений.

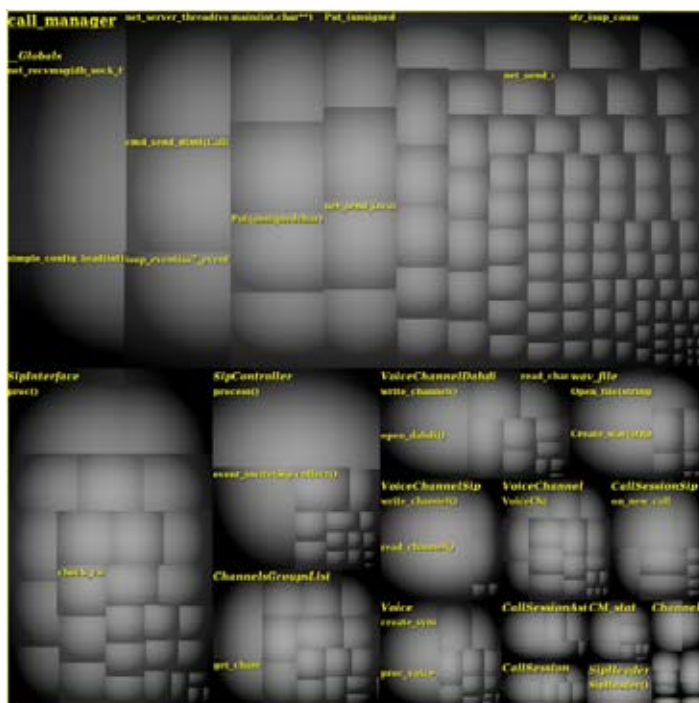


Рис. 3. Матрица проекта № 3

Сравнивая три проекта, написанных с одной целью, и анализируя отчеты, представленные СppDepend, составили конкурентный профиль всех анализируемых проектов. Конкурентный профиль отражен в табл. 1.

Обсуждение результатов

Таким образом, анализ качества программного кода трех проектов показал, что самым качественным с точки зрения программной метрики оказался проект под номером три. Об этом говорит конкурентный профиль, приведенный в табл. 1. Его код более лаконичный, структура программы проста, код прокомментирован в пределах нормы. Автор придерживался малых объемов как методов, так и типов. Данная программа более стабильна, чем остальные, и, следовательно, более управляема. Третий проект больше всего подходит для интеграции в другие системы, так как легко улучшается и модернизируется под условия заказчика.

В соответствии с задачей разработки унифицированных систем документации SMK стандарта ИСО 9001 на предприятии было разработано и написано руководство по качеству и обязательные документированные процедуры: ДП «Управление документацией»; ДП «Управление записями»; ДП «Внутренний аудит»; ДП «Управление несоответствующей продукцией»; ДП «Предупреждающие и корректирующие действия»; ДП «Управление персоналом»; ДП «Участие в тендерных торгах»; ДП «Проектирование и разработка программного обеспечения»; ДП «Техническая поддержка»; ДП «План качества»; ДП «Выбор поставщика».

Безусловно, существует взаимосвязь между метрическими параметрами программного кода, исследуемого нами, документированными процедурами системы менеджмента качества

Таблица 1. Конкурентный профиль для трех проектов

Параметры	Первый проект (шт.)	Второй проект (шт.)	Третий проект (шт.)
Общее количество строк	8 834	21 742	5 061
Количество модулей в проекте	3	4	1
Типы слишком большого объема	1 (1370 строк)	3 (1275;810;593 строки)	1 (725 строк)
Методы слишком сложные	4 (среднее:24,25)	29 (среднее:51,83)	4 (среднее:39,75)
Быстрый обзор метода рефакторинга	116	216	49
Метод слишком большого объема	76 (среднее:59,62)	116 (среднее:101,78)	40 (72,1)
Методы с большим количеством параметров	-	2 (9;9)	2 (19;11)
Слишком сложные методы	25	70	11
Плохо прокомментированные методы	89	211	47
Методы со слишком большим количеством переменных	3 (18;18;18)	39 (среднее:25,9)	8 (среднее:19;11)
Типы с большим количеством методов	9 (среднее:47)	2 (40;28)	2 (28;24)
Типы с большим количеством полей	2 (35;37)	3 (35;29;28)	2 (31;29)
Типы с плохой сплоченностью	7	5	3

и ответственными участниками проекта. Данную взаимосвязь можно представить как трехмерную матрицу. Одно из сечений являет собой стандартную процедуру и есть в соответствии с ИСО 9001 матрица ответственности.

Следующая плоскость документированные процедуры – метрики. Однако не каждая документированная процедура напрямую связана с метриками кода. В табл. 2 отражена взаимосвязь основных метрик программного продукта и документальных процедур (ДП).

Залитой зоной показана значимость метрик и их проверка на определенных этапах жизненного цикла программного средства. На каждом этапе нецелесообразно проверять и контролировать качество всех метрик. Но для того чтобы правильно скорректировать и повысить качество написания программного кода, необходимо учитывать их все на определенных этапах, как показано цветом в табл. 2.

Плоскость метрики – ответственные в матрице управления качеством программного продукта. Участники проекта на определенных этапах жизненного цикла, которые были рассмотрены выше, выставляют свою оценку качеству программного кода в зависимости от того, насколько была соблюдена та или иная метрика. Например, значение цикломатической сложности показывает, насколько программа сложна и ее можно оценить в балльной системе. Пример такого балльного подхода приведен в табл. 3.

Для лучшего восприятия программного кода программный комплекс должен быть разбит на небольшие по количеству строк блоки, их по аналогии можно оценить в баллах (табл. 4). Нестабильность программы, которая связана с центростремительным и центробежным сцеплением, можно оценить по такому же принципу.

Таблица 2. Зависимость метрик от документированных процедур

Метрики \ ДП	Количество строк	Процент комментируемости	Цикломатическая сложность	Центростремительное и центробежное сцепление (Стабильность)	Количество методов на класс, глубина дерева, количество потомков	Количество полей и функций в структуре программного кода
Проектирование и разработка						
Внутренний аудит						
Управление несоответствующей продукцией						
Предупреждающие и корректирующие действия						

Таблица 3. Пример балльной оценки цикломатической сложности программы

Сложность программы	Вероятность ошибок	Балльная оценка
1 – 10	Простая функция, ошибки маловероятны	5
11 – 20	Более сложная, средняя вероятность ошибок	4
21 – 50	Сложная, велика вероятность ошибок	3
51 и более	Нетестируемая (риск очень велик)	2

Для каждого участника проекта та или иная метрика будет иметь большее или меньшее значение. Следовательно, оценки у всех участников будут разные. После этого коэффициент качества программного кода находим по формуле

$$K = \frac{1}{E * L} \sum_{e=1}^E \sum_{l=1}^L k_{el}, \quad (1)$$

где E – количество ответственных; L – количество метрик; k_{el} – оценки, поставленные ответственными за соблюдение или несоблюдение тех или иных метрик.

Заключение

Подведем некоторые итоги работы. Анализ качества программного обеспечения в динамике, т.е. анализ проектов по функциональному назначению, схожих, но написанных за разные промежутки времени и разными программистами, показал, что качество написанных программных модулей улучшается и упрощается их интеграция в другие проекты. Об этом говорит конкурентный профиль, составленный для трех проектов, который демонстрирует, что самым качественным с точки зрения программистики оказался проект под номером три. Его код более лаконичный, структура программы проста, код прокомментирован в пределах

Таблица 4. Количество строк в структуре

Номер п/п	Количество строк в структуре программы	Балльная оценка
1	Меньше 500	5
2	499-800	4
3	799-1000	3
4	Больше 1000	2

нормы на 19 %. Цикломатическая сложность методов программы равна 56, типов 568, что по параметрам лучше остальных. Следовательно, данная программа более стабильна, чем остальные, и, следовательно, более управляема. Третий проект больше всего подходит для интеграции в другие системы, так как легко улучшается и модернизируется под условия заказчика.

Программное средство CppDepend позволило оценить качество кода больших объемов, построило граф потока управления и структуры и сгенерировало отчет.

Для системы менеджмента качества внедрение такой программы позволило улучшить качество кода, уменьшить затраты по времени на разработку последующих модулей для других комплексов. В рамках данной программы возможно отслеживание проектирования и написания кода с датами и графиками, что помогло бы менеджерам проектов выполнять свою работу, связанную с отслеживанием длительности проекта, не отвлекая программистов. Предлагается внедрить CppDepend в качестве инструмента проектирования и среды разработки для всех программистов.

Метрический синтез показал, что количество и качество программного кода зависит от логико-умственных способностей программиста, его опыта реализации программных продуктов и степени владения языком программирования. Это позволило нам установить зависимости метрик качества программного кода, обязательных документированных процедур и ответственных на каждом этапе жизненного цикла создания программного комплекса, и предложить компании СМК использовать трехмерную матрицу управления качеством.

Трехмерная матрица была рассмотрена в следующих плоскостях:

- плоскость документированные процедуры – метрики. Показана значимость метрик и их проверка на определенных этапах жизненного цикла программного средства;
- плоскость документированные процедуры – ответственные. Представлено в жизненном цикле продукции место проверки и оценки качества программного обеспечения;
- плоскость метрики – ответственные. Участники проекта на определенных этапах жизненного цикла выставляют свою оценку качеству программного кода в зависимости от того, насколько была соблюдена та или иная метрика. Совокупность данных оценок позволила найти коэффициент качества программного кода, который в своем роде является приблизительной оценкой, будь то пятибалльная система или стобалльная. Свод метрических величин в конечном итоге был приведен к одному числовому показателю – коэффициенту качества программного кода.

Качество программного кода показывает, насколько устойчива и надежна программа, профессиональный уровень работающих в компании программистов, скорость и надежность самой компании в целом.

Список литературы

[1] Черников Б.В. *Управление качеством программного обеспечения*: учебное пособие. М.: Форум, 2012, 240 с. [Chernikov B.V. *Quality management of software*. Moscow, Forum, 2012, 240 p.]

[2] ГОСТ ISO 9004-2010. *Национальный стандарт Российской Федерации*. Менеджмент для достижения устойчивого успеха организации. Подход на основе менеджмента качества. Введ. 23.11.2010. М.: Росстандарт, 2010. 47 с.

[3] ГОСТ ISO 9001-2011. *Межгосударственный стандарт*. Системы менеджмента качества. Требования. Введ. 22.12.2011. М.: Росстандарт, 2011. 32 с.

[4] ГОСТ ISO 9000-2011. *Межгосударственный стандарт*. Системы менеджмента качества. Основные положения и словарь. Введ. 22.12.2011. М.: Росстандарт, 2011. 31 с.

[5] Липаев В.В. *Проектирование и производство сложных заказных программных продуктов*. М.: СИНТЕГ, 2011, 408 с. [Lipaev V.V. *Design and manufacture of complex custom software products*. Moscow, SINTEG, 2011, 408 p.]

[6] C and C++ Code Counter, Available: <http://cccc.sourceforge.net>

[7] Source monitor. Available: www.campwoodsw.com/sourcemonitor.html (Accessed 15.03.2015).

[8] Habrahabr Available: <http://habrahabr.ru/post/111524/> (Accessed 15.04.2015)

[9] HomeCodeMetrics Available: <https://reflectoraddins.codeplex.com/wikipage?title=CodeMetrics&referringTitle=> (Accessed 20.04.2015)

[10] Developer Network Available: <https://msdn.microsoft.com/en-us/library/bb429476.aspx> / (Accessed 25.04.2015)

[11] Nashi produkty Svyaz'kom Available: <http://www.svyazcom.ru/ru/products/> (Accessed 10.05.2015)