

Б.Н. СОКОЛОВ, аспирант НТУ «ХПИ»

МОДЕЛИ УПРАВЛЕНИЯ КАЧЕСТВОМ ОБРАБОТКИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

У статті запропоновано підхід до управління якістю обробки вимог до програмного забезпечення, що базується на системному підході та стандартах ISO. Визначені цілі функціонування процесу аналізу та обробки вимог, розглянуті його характерні риси, особливості взаємодії з іншими процесами життєвого циклу програмних систем. Запропонована концепція управління якістю вхідною, вихідною інформацією та самим процесом аналізу та обробки вимог.

В статье предлагается подход к управлению качеством обработки требований к программному обеспечению, который основывается на системном подходе и стандартах ISO. Выделены цели функционирования процесса анализа и обработки требований, рассмотрены его основные свойства, особенности взаимодействия с другими процессами жизненного цикла программных систем. Предложена концепция управления качеством входной, выходной информацией и самим процессом анализа и обработки требований.

An approach to quality requirements management for software is outlined. This approach is based on system approach and ISO standards. Purposes of the system requirements analysis process activity were defined; its main properties and specific of interaction with other lifecycle processes were considered. The concept of quality control of the input, output information and the process of the system requirements analysis process was proposed.

Введение. Успех разрабатываемого программного обеспечения (ПО) в значительной степени зависит от того, насколько оно в конечном итоге удовлетворяет требования заинтересованных лиц (ЗЛ). Поэтому очень важно правильно собрать, формализовать и проанализировать требования ЗЛ по отношению к разрабатываемому продукту.

В [1] следующим образом определяются процесс программной инженерии и процесс жизненного цикла программной системы: «Под процессом программной инженерии понимается множество логически связанных видов деятельности по определению, проектированию и построению прикладных программных систем. Процесс программной инженерии имеет иерархическую архитектуру, включая множество процессов жизненного цикла программных систем. Эти процессы группируются по категориям – основные процессы, процессы поддержки, организационные процессы – и касаются аспектов не только собственно разработки программных систем, но и организации, контроля и управления разработкой».

Целью процесса анализа и обработки требований является преобразование требований, заданных заинтересованными лицами, во множество требуемых системных требований, которые будут использоваться в дисциплине проектирования [2].

Выявление системных требований к разрабатываемой системе происходит путем анализа информации об ее предполагаемом использовании. Спецификация системных требований должна включать в себя:

- Функции и возможности системы.
- Пользовательские, организационные и бизнес требования.
- Требования по безопасности, защищенности, эргономике, интерфейсу, производимым операциям, сопровождаемости.
- Архитектурные ограничения и требования по квалификации.

Ориентируясь на стандарты ISO [2,3] мы будем рассматривать разработку программного обеспечения как совокупность процессов. На рис. 1 изображены процессы, которые являются объектами исследования нашей группы.



Рис. 1. Процессный подход к управлению качеством ПО

Используя подход Деминга [4,5] получим схему управления качеством анализа обработки требований (рис. 2). Мы рассматриваем в данном случае 4 измерения качества, при этом считаем, что пятое измерение качества – качество программного продукта, по окончании полного цикла разработки, учитывать на данном этапе весьма сложно.



Рис. 2. Управление качеством анализа обработки требований

В данной статье мы рассмотрим взаимодействие процесса анализа и обработки требований с другими процессами жизненного цикла, обсудим его основные функциональные элементы и возможность оказания управляющих воздействий на них. В большей степени мы будем касаться управления качеством входа и управления качеством процесса.

2. Взаимодействие с другими процессами жизненного цикла. В соответствии со стандартом ISO/IEC 12207 [2] процесс анализа и обработки требований осуществляется после процесса определения требований и перед процессом проектирования. Процессы жизненного цикла связаны *прямой* и *обратной связью*. В случае, когда процедура контроля качества на входе одного процесса аналогична процедуре контроля качества на выходе другого процесса, принято использовать соответствующий блок контроля качества для контроля входных данных (чтобы избежать дублирования идентичных блоков).



Рис. 3. Взаимодействие процессов определения и анализа и обработки требований

На рис. 3 мы видим схему взаимодействия процессов определения требования и анализа и обработки требований. В соответствии с [1,2] в результате выполнения процесса определения требования мы получаем множество собранных требований ЗЛ и системных ограничений. При этом для получения этих данных могут быть применены различные подходы, основанные на теории распознавания текста, методах получения экспертных оценок, имитационном моделировании, психологии, социологии и т.д. При сборе требований возникает множество проблем, связанных, главным образом, с человеческим фактором. Ведь источником информации выступают заинтересованные лица, у каждого из которых могут быть свои личные интересы, которые обладают различным уровнем компетентности и т.д. Так в [6] выделяют следующие четыре типа возможных проблем при сборе требований ЗЛ: конфликты интересов ЗЛ; ограничения, связанные со способностями, компетентностью ЗЛ, а также нежелание ЗЛ сотрудничать;

неопределенность предпочтений ЗЛ; высокая сложность (например, большое количество требований, зависящих друг от друга).

На рис. 4 изображена схема взаимодействия процессов проектирования и анализа и обработки требований. В соответствии с ISO [2,3] в результате выполнения процесса анализа и обработки требований мы получаем структурированные требования заинтересованных лиц, результаты базового анализа этих требований и информацию по поддержке принятия решений по управлению проектом. Задачи принятия решений могут возникать во время каждого из этапов жизненного цикла [5,7-11]. В данном случае информация по поддержке принятия решений по управлению проектом включает в себя модели бизнес-процессов и решения по управлению проектом (основные этапы проекта, контрольные точки и т.д.). Модели бизнес-процессов могут быть использованы не только в качестве источника знаний по реализации определенного бизнес-процесса. Так в случае если разрабатываемая система имеет сервис-ориентированную архитектуру [12], модели бизнес-процессов играют важную роль для связывания различных сервисов между собой [13]. Для этих целей используются различные языки для формального описания бизнес-процессов и протоколов их взаимодействия между собой (такие как BPEL), а также системы управления бизнес-процессами [14].

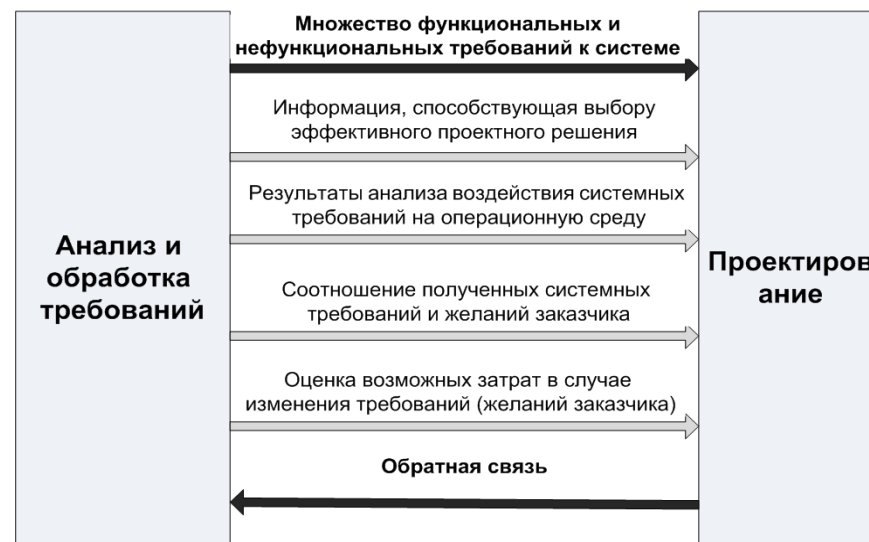


Рис. 4. Взаимодействие процессов анализа и обработки требований и проектирования

3. Основные функциональные элементы анализа и обработки требований. Изобразим основные функциональные элементы анализа и обработки требований на рис. 5. В общем случае возможна различная степень использования различных блоков (как блоков полностью, так и элементов из

которых они состоят). На выбор степени использования блоков будут влиять такие факторы, как размеры системы, ее тип (например, сервис-ориентированная), различные ресурсные ограничения. Для решения задач подобного рода может быть использован подход метод инжиниринга [15-17], который поможет на основании требований (основанных на ситуационных факторах компании) к процессу жизненного цикла и его элементам определить достаточную степень их использования.



Рис. 5. Основные функциональные элементы анализа и обработки требований

Рассмотрим подробнее основные элементы процесса анализа и обработки требований. Как мы уже говорили ранее, в результате выполнения процесса определения требования мы получаем множество собранных требований ЗЛ и системных ограничений. Эти требования поступают на вход блока анализа корректности требований ЗЛ и ограничений (рис. 6). В этом блоке осуществляется проверка полноты и корректности требований, их тестируемости, осуществимости проектирования архитектуры системы, возможности использования и сопровождаемости системы. В общем случае проверка полноты требований зависит от формата, в котором они представлены, и зачастую сводится к синтаксическим проверкам [18]. Проверка корректности подразумевает под собой анализ связей между требованиями и поиск конфликтов. Анализ осуществимости проектирования

архитектуры системы призван показать, насколько сложно будет спроектировать систему на основании собранных требований, а также оценить вероятность появления ошибок на этапе проектирования. Для оценки операционных характеристик и характеристик сопровождаемости системы существуют специфические метрики (такие как соотношение решенных проблем за месяц к возникшим проблемам за месяц) [19, 20]. Однако, метрики эти, в большей степени, применимы в случаях, когда система уже разработана (либо к завершённым компонентам, прототипам системы и т.д.). В нашем случае доступны, главным образом, различные виды анализа рисков, связанных с отклонением операционных и эргономических свойств системы от желаний заказчика; неточностью, нечеткостью и неполнотой системных требований, способными привести к частому внесению изменений в систему после ее выхода и т.д. Управление данным блоком осуществляется путем конфигурирования его настроек, связанных, по большей части, с изменением степени жесткости проверок. Очевидно, что чем жестче проверки, тем выше шанс, что требования будут возвращены на доработку на этап сбора требований, что приведет к дополнительным финансовым издержкам, однако, если условия будут слишком мягкими, повышаются риски возникновения ошибок при выполнении последующих процессов жизненного цикла.



Рис. 6. Блок анализа корректности требований заинтересованных лиц и ограничений

В блоке формирования множества функциональных и нефункциональных требований (рис. 7) осуществляется анализ и последующее преобразование требований ЗЛ и системных ограничений во множество функциональных и нефункциональных требований, поступающее на вход к процессу проектирования. Чем лучше будут произведены действия в этом блоке, тем легче будут проходить дальнейшие процессы жизненного цикла и тем меньше вероятность возникновения различных ошибок в финальном продукте. С другой стороны на функционирование блока накладываются ресурсные и временные ограничения. После того как были сформированы функциональные и нефункциональные системные требования, их, как правило, ранжируют. Довольно часто разработчик программного продукта не имеет возможности в полной мере удовлетворить требования всех ЗЛ в рамках существующих ресурсных и временных ограничений. В результате

получается задача определения подмножества требований ЗЛ, реализация которых с одной стороны вписывалась в ресурсные рамки, а с другой стороны позволяла бы создать продукт, удовлетворяющий ЗЛ. Для ранжирования требований могут использоваться как метрики, показывающие соотношение относительной важности требования к его относительной стоимости [21], так и более сложные показатели, учитывающие время на разработку отдельных модулей и т.д. [22].

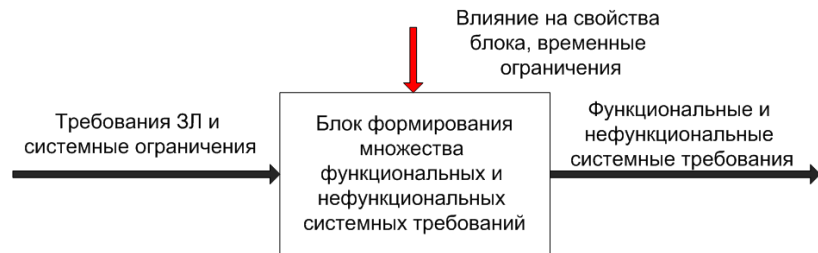


Рис. 7. Блок формирования множества функциональных и нефункциональных системных требований

Блок поддержки принятия решений по управлению проектом (рис. 8) предоставляет возможность на основании требований к системе получать модели бизнес-процессов и множество решений по управлению проектом. Для данного блока необходимы программные средства для повторного использования бизнес-процессов и решений по управлению проектом, включающие в себя соответствующие репозитории, конверторы между различными форматами и подсистемы поиска.

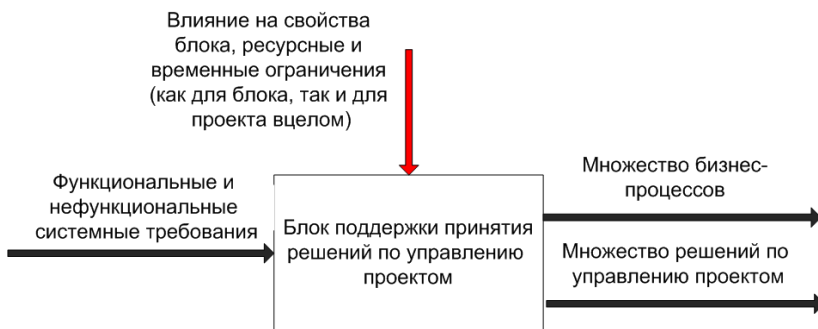


Рис. 8. Блок поддержки принятия решений по управлению проектом

Выводы. Мы рассмотрели процесс анализа и обработки требований к ПО, качество выполнения которого, так же как и других процессов жизненного цикла, в определенной степени влияет на качество получаемого программного продукта. Управление качеством данного процесса

осуществляется путем варьирования степени использования его блоков (и их элементов), наложением различных ресурсных и временных ограничений на выполнение определенных операций в блоках (и их элементах), а также заданием различных настроек для блоков (таких как уровень жесткости проверки качества входных данных).

В дальнейшем планируется разработать процедуры анализа корректности требований ЗЛ и системных ограничений, процедуру трансформации требований ЗЛ в функциональные и нефункциональные системные требования, прототип ПО для повторного использования бизнес-процессов, прототипа ПО для повторного использования решений по управлению проектом.

Список литературы: 1. Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, Е.М. Лаврищева, В.Ю. Сулов. Основы инженерии качества программных систем 2 изд. перераб. и доп. – Киев: Академперіодика, 2007. 2. ISO/IEC 12207:2008 System and software engineering – Software life cycle processes. 3. ISO/IEC 15288:2008 Systems and software engineering — System life cycle processes. 4. W. Edwards Deming. Out of the Crisis. – Cambridge: MIT Press, Center for Advanced Engineering Study, 1986. 5. V. Basili, G. Caldiera. Improve Software Quality by Reusing Knowledge and Experience, Sloan Management Review, MIT Press, vol. 37(1), 1995 – p. 55-64. 6. S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, P. Grünbacher. Value-Based Software Engineering. – Berlin: Springer-Verlag Berlin Heidelberg, 2006. 7. R.Kazman, M.Klein, P.Clemens. ATAM: Method of Architecture Evaluation. Technical Report// CMU/SEI-2000-TR-004, CMU, 2000. 8. M.Svahnberg, C.Wholin, L.Lundber. A Quality-Driven Decision-Support Method for Identifying Software// Int. Journal of Software Engineering and Knowledge Engineering, vol. 13(5), 2003 – p. 547-573. 9. R. Ocker, S. R. Hiltz, M. Turoff, J. Fjermestad. The effects of distributed group support and process structuring on software requirements development teams: results on creativity and quality// Journal of Management Information Systems, vol. 12(3), December 1995 – p. 127–153. 10. L. Liu, E. Yu, J. Mylopoulos. Security and privacy requirements analysis within a social setting// Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International – p. 151-161. 11. N.A.M. Maiden, G. Rugg. ACRE: selecting methods for requirements acquisition// Software Engineering Journal, vol. 11(3), May 1996 – p. 183-192. 12. Th. Erl. Service-Oriented Architecture: Concepts, Technology, and Design. – NJ: Pearson Education, 2005. 13. M. Wheske. Business Process Management. Concepts, Languages, Architectures. – Berlin: Springer-Verlag Berlin Heidelberg, 2007. 14. Д. Грязнов. Система управления бизнес-процессами как элемент процессного подхода к управлению бизнесом// Intelligent Enterprise, №4, 2008 – с. 12-15. 15. A.F. Harmsen. Situational Method Engineering// Doctoral dissertation. – Twente: University of Twente, 1997. 16. S. Brinkkemper, I. van de Weerd, M. Saeki, J. Versendaal. Process Improvement in Requirements Management: A Method Engineering Approach// Requirements Engineering Foundation for Software Quality. – Berlin: Springer-Verlag Berlin Heidelberg, 2008 – p. 6-22. 17. J. Ralyté, C. Rolland. An Assembly Process Model for Method Engineering// Advanced Information Systems Engineering. – Berlin: Springer-Verlag Berlin Heidelberg, 2001 – p. 267-283. 18. J. E. Burge, J. M. Carroll, R. McCall, I. Mistrik. Rationale-Based Software Engineering. – Berlin: Springer-Verlag Berlin Heidelberg, 2008. 19. St. H. Kan. Metrics and Models in Software Quality Engineering, Second Edition. – Boston: Pearson Education, 2003. 20. Kh. El Emam. The ROI from software quality. – Boca Raton, FL: Auerbach Publications Taylor & Francis Group, 2005. 21. J. Karlsson, K. Ryan. A cost-value approach for prioritizing requirements// IEEE Software, vol. 14(5), 1997 – p. 67-74. 22. Bj. Regnell, M. Höst, J. Nat, P. Beremark, Th. Hjelm. An Industrial Case Study on Distributed Prioritisation in Market-Driven Requirements Engineering for Packaged Software// Requirements Engineering, Springer-Verlag London Limited, vol. 6 (1), 2001 – p. 51-62.

Поступила в редакцию 15.12.09