

СИСТЕМНЫЙ АНАЛИЗ ПРОЦЕССА ВЫЯВЛЕНИЯ И ПРЕДСТАВЛЕНИЯ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Мысливец А.Н.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Меженная М.М. – кандидат технических наук, доцент

В работе приводится авторский подход к составлению проектной документации на основании ряда выделенных артефактов. Данные артефакты помогают аналитику облегчить обмен информацией между стейкхолдерами и командой разработки и делают коммуникацию предельно эффективной.

Целью работы является разработка системно обоснованного набора шагов и соответствующего набора артефактов для процесса выявления и представления требований к ПО. Объектом исследования выступает процесс разработки ПО, а предметом исследования – методы и средства формирования требований.

Для достижения указанной цели решены следующие задачи:

- 1) Определить основные атрибуты требований для их документирования;
- 2) Изучить основные этапы жизненного цикла ПО и определить место процессов работы с требованиями в нем;
- 3) Разработать авторский подход к созданию документации на основе существующих методик и артефактов;
- 4) Создать и апробировать демонстрационные документы на основе разработанного подхода.

Требования являются отправной точкой для определения того, что проектная команда будет проектировать, реализовывать и тестировать. Опыт индустрии информационных технологий однозначно показывает, что вопросы, связанные с разработкой требований, оказывают критически-важное влияние на программные проекты, в определенной степени – на сам факт возможности успешного завершения проектов.

По данным исследования, проведенного компанией IBM в области IT, 60% затрат времени разработчики программного обеспечения несут в результате неэффективного подхода к управлению требованиями [1]. В организациях, пренебрегающих аналитикой, проекты в 3 раза чаще заканчиваются неудачей, чем успехом. При правильном определении требований и управлении ими перерасходы по проекту можно снизить на 20% благодаря сокращению числа неточных, неполных и упущенных требований. Практика показывает, что недостаточная проработка требований зачастую проявляется лишь тогда, когда проект почти завершен, а значительная часть ресурсов, выделенных на его реализацию, уже затрачена. Устранение проблем на этапе разработки обходится гораздо дороже, чем тщательная проработка на стадии анализа (рис. 1).



Рисунок 1 – Стоимость исправления ошибки в зависимости от момента её обнаружения

Как правило, все действия по выявлению, анализу, документированию и проверке требований не удается выполнить последовательно и за один проход. Они выполняются попеременно, поэтапно и повторяются (рисунок 2).

Работа с программными требованиями – творческий процесс, и каждый аналитик использует свои инструменты для получения результата. Условно, все подходы работы с требованиями можно разделить на две группы:

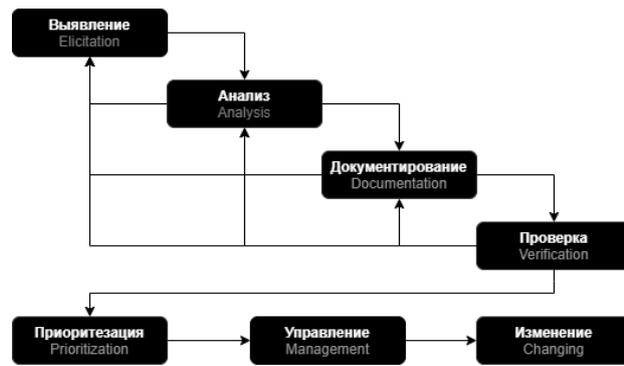


Рисунок 2 – Этапы работы с требованиями

- 1) техническое задание (ТЗ), сформированное на основе определенного стандарта;
- 2) различные форматы документов в зависимости от типа требований.

При разработке больших систем функциональности становится так много, что работать с одним только ТЗ становится крайне неудобно. Альтернативным подходом является выбор формата документа под различные типы требований. Для фиксации бизнес-требований используется один формат, для фиксации системных требований - другой, а для описания дизайна программной системы - третий.

В данной работе были проанализированы лучшие практики по созданию проектной документации и документированию требований, в частности [2,3]. В результате был разработан авторский подход к составлению проектной документации на основании следующих артефактов:

- глоссарий (Glossary);
- анализ заинтересованных сторон (Stakeholder matrix);
- матрица ответственности (RACI matrix);
- план коммуникаций (Communication plan);
- план управления требованиями (Requirements management plan);
- персоны (User personas);
- протоколы встреч (Meeting notes);
- карта пользовательских историй (User story map);
- метод мозгового штурма (Brainstorming);
- шаблон бизнес-модели (Business model Canvas);
- словарь данных (Data dictionary);
- опросные листы (Questionnaire);
- пользовательские истории (User stories);
- сценарий использования (Use cases);
- карта пути пользователя (Customer journey map);
- дорожная карта продукта (Product roadmap);
- карта планирования эффекта (Impact mapping);
- матрица трассировки требований (Traceability matrix);
- *чек-лист нефункциональных требований (Non-functional requirement checklist).*

Перечисленные выше артефакты помогают аналитику облегчить обмен информацией между стейкхолдерами и командой разработки и делает коммуникацию предельно эффективной. Они позволяют команде синхронизироваться и сводят к минимуму возникновение разночтений и недопонимания между участниками процесса [4].

Важно помнить, что не существует единственно правильного способа документирования требований и другой проектной документации. Подход определяется процессом, заинтересованными лицами, внешними требованиями и т.п. Всегда необходимо учитывать кто будет потребителем того или иного документа в проекте, а также как этот документ будет использоваться.

Список использованных источников:

11. Ляхов, Н. Как улучшить управление ИТ-проектами в организации [Электронный ресурс] / Н. Ляхов. – 2015. – Режим доступа: <https://www.infostart.ru/public/418073>. – Дата доступа: 29.02.2020.
12. Wiegers K. Software requirements / K. Wiegers, J. Beatty – Microsoft Press, 2013. – 674 p.
13. A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide) IIBA (Author), Kevin Brennan (Editor). International Institute of Business Analysis; 2nd edition (March 31, 2009). ISBN: 978-0981129211.
14. Купер, А. Психбольница в руках пациентов. Алан Купер об интерфейсах: Почему высокие технологии сводят нас с ума и как восстановить душевное равнов. Изд., испр / А. Купер. - М.: Символ, 2016. - 336 с.