

Бычков М. Ю., магистрант 2 курса,

Тверской государственный технический университет

Котлинский С. В., кандидат военных наук, доцент кафедры информационных систем, Тверской государственный технический университет

ОСНОВНОВНЫЕ ПОДХОДЫ К ОБЕСПЕЧЕНИЮ ТРАССИРУЕМОСТИ ТРЕБОВАНИЙ

Аннотация: В статье рассмотрены основные подходы трассируемости требований при разработки программного продукта(ПП).

Ключевые слова: Требования к системам, трассировка требований, обратная семантическая трассировка (ОСТ), моделирование процессов трассировки требований.

Abstract: The article discusses the main approaches to requirements traceability in software product (PP) development.

Keywords: System requirements, requirements tracing, semantic reverse tracing (OST), modeling of requirements tracing processes.

Актуальность данной статьи в том, что большая часть программных систем имеют неустойчивые или постоянно изменяющиеся требования к конечным программным продуктам(ПП) в процессе их разработки. Для совершенствования процесса разработки ПП следует различать требования на ранних этапах, используя методы определения требований, и применять различные методы их трассировки.

Объектом исследования является процесс разработки программного обеспечения и требования к его качеству.

Предметом исследования являются методы трассировки требований к программному продукту.

Целью данной работы является выявление возможных способов повышения качества разрабатываемого программного продукта за счет совершенствования методов трассировки требований.

Требования создаются итерационно [1], путем постоянного общения представителей заказчиков с аналитиками и разработчиками будущего программного продукта в целях выявления требований. Требования изменяются в зависимости от задач и условий их определения, а также постоянного уточнения на этапе заключения договора на создание ПП. Одним из методов установления зависимости между сформулированными требованиями и их изменениями является трассировка [2], т.е. поддерживается развитие и обработка требований с прослеживанием связей, которые должны быть зафиксированы по двум направлениям - от источника требований к реализации и, наоборот (Рис. 1.).

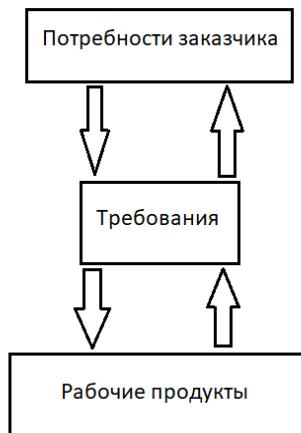


Рис. 1. Место требований между потребностями заказчика и рабочими продуктами

Если после разработки некоторого ПП возникает необходимость в преобразование конкретных требований или проследить за происхождением внесенных требований в одном из направлений данной схемы трассирования, то детализируются связи между отдельными требованиями и элементами рабочих модулей программы. В случае трассирования требований от продукта,

движение в обратном направлении, т.е. к требованиям, можно выяснить, как написана каждая строка программы этого продукта и соответствует ли она отдельным атрибутам требований. Связи трассируемости требований помогают найти незапланированные и реализованные некоторые функции или фрагменты программ, не соответствующие заданным требованиям. И наоборот, выявить нереализованные требования функциональности ПП. Взаимосвязи и зависимости между отдельными требованиями сохраняются, например, в таблице трассируемости, удаляются, обновляются при различных изменениях.

Трассировка может быть выборочной для отдельных элементов или связанной с другими элементами, а также с возможными переходами от одной модели проектирования к другой, путем проверки изменения одних элементов в другие.

Методы трассировки [3] основаны на формальных связях, между элементами требований, либо ограничиваются описаниями функций, ситуаций, контекста и возможных решений.

Основа трассировки состоит из:

- требования, изменяемые при формировании;
- некоторое описание выполняемых функций в рабочем продукте системы, которые не предполагались, не возникли в связи практической ситуации;
- связи между многочисленными модулями процесса проектирования системы на этапе ЖЦ программного продукта и принятые решения о необходимости изменения требований из-за выявления недостатков, неточностей в промежуточном этапе разработки продукта;
- сведения о согласованных атрибутах требований на разных уровнях рассмотренной схемы трассирования и сохранение ее матрице трассирования;
- системные требования, касающиеся повторного использования готовых компонентов программного продукта;
- итоги тестирования, по которым можно определить части кода, требующие обследования на наличие в них дефектов.

Процесс трассирования состоит из следующих этапов [3]:

- выбирается элемент (фрагмент, функция или некоторая часть) из матрицы трассирования требований, за которым проводится наблюдение на этапах ЖЦ;
- формируется список вопросов, по которым на каждом этапе проверяются связи при реализации требований в продукте, и, если изменяется какое-то звено в цепочке требований (рис. 1.), то может изменится процесс разработки этого элемента на последующем этапе ЖЦ;
- проводится анализ каждого требования на целостность выполнения согласно принятому плану;
- конкретизация ресурсов выполнения проекта при необходимости проведения изменений в требования и в элементы проекта.

Условием принятия решения о возможных изменениях требований и результатов промежуточного проектирования, является обновленные сведения о связях между различными элементами системы и первоначально заданными требованиями к ним.

Трассировка обеспечивает [2]:

- введение более сложных связей вместо простых, или связи частных случаев;
- использование разных путей трассировки (между элементами или иерархическими связями);
- ведение базы данных объектов трассировки и отношений между ними.

Первой из многочисленных моделей процессов трассировки требований(МПТТ) была предложена в [3], ее схема показана на рис. 2.

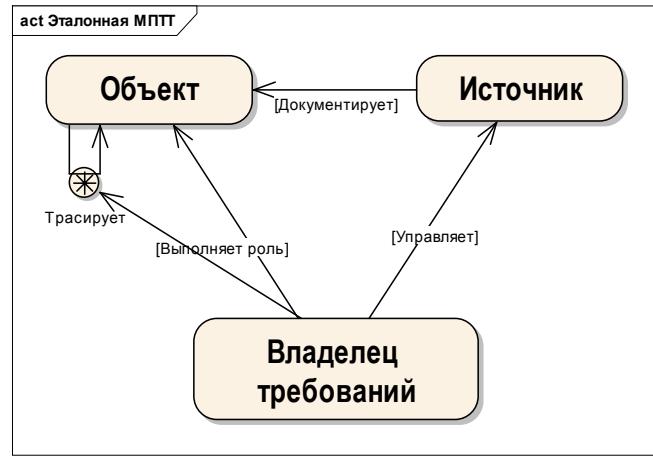


Рис. 2. Эталонная МПТТ

Основными элементами в схеме трассировки требований выступают такие сущности как: Владелец требований, Источник и Объект, которые связаны такими базовыми отношениями как: выполняет роль, трассирует, управляет и документирует.

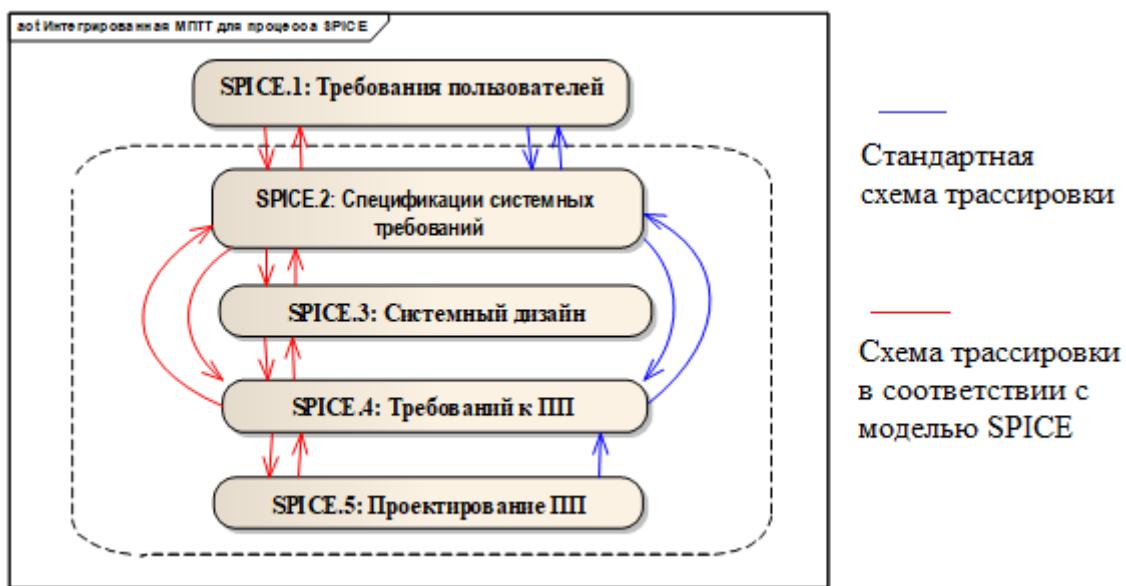
Авторы этой МПТТ [3] называют ее эталонной (reference) моделью трассировки, что предполагает возможность использования ее концептуальных компонентов (сущностей и отношения между ними) в других МПТТ, построенных на ее основе. К категории концептуальных МПТТ следует отнести и итерационную схему процесса трассировки требований в гибкой (agile) разработке ПП [4], которая приведена на рис. 3.



Рис. 3. Итерационная МПТТ

Непосредственное влияние процессов трассировки и технологий обеспечения качества программного продукта, в частности, подхода SoftwareProcessImprovementCapabilitydEtermination (SPICE), отражает МПТТ, концептуальная схема которой показана на рис. 4.

Основное преимущество этой модели, по мнению ее авторов[2,3], заключается в том, что она интегрирована в многоуровневую модель всего процесса разработки ПП по методологии SPICE, в соответствии с которой, на первом этапе на основе требований пользователей (SPICE.1) вначале выполняется этап разработки общесистемных спецификаций системных требований (SPICE.2), затем – этап высокоуровневого (архитектурного) проектирования всей целевой системы (SPICE.3), далее, с учетом его итогов, происходит переход к этапу разработки требований к ПП (SPICE.4), и только после этого выполняется этап непосредственного проектирования (SPICE.5).



По первичным требованиям создается тестовая платформа для разработки.

Проводится пересечение всех требований, реализуются модели тестирования частных случаев.

По результатам пересечений формируется матрица трассировки требований (таблица 1).

Таблица 1. Матрица трассировки требований

	Требование 1	Требование 2	Требование 3
Тест-кейс 1	+		
Тест-кейс 2		+	
Тест-кейс 3		+	
Тест-кейс 4			+

Результат доносится до заказчика с возможными дополнениями. После принятия всех изменений в разработке ПП, заказчиком будут сгенерированы новые требования и так до тех пор, пока все требования не будут удовлетворять заказчика.

Заключение.

В работе были описаны основные подходы к обеспечению трассируемости требований для контроля качества при разработке ПП.

Библиографический список:

1. Лаврищева Е, Петрухин В. Методы и средства инженерии программного обеспечения <https://intuit.ru/studies/courses/2190/237/info>.
2. ISO/IEC 9126, Information Technology – Software quality characteristics and metrics 1997.
3. Ramesh B. Toward Reference Model for Requirements Traceability [Текст] / B. Ramesh, M. Jarke. – IEEE Software Engineering – 2001. – #27(1) – P.58–93.
4. Tkachuk M. V. Models and Tools for Effectiveness Increasing of Requirements Traceability in Agile Software Development [Текст] / TkachukM.V.,

R.O.Gamzayev, H.C.Mayr, V.O. Bolshutkin – Проблемы программирования. – К.: НАНУкраїни. – 2012. – № 2–3 (спец. выпуск). – С.252 – 260.