

## АНАЛИЗ ТРЕБОВАНИЙ НА ОСНОВЕ UML-МОДЕЛЕЙ И ДИНАМИЧЕСКИХ HTML-ПРОТОТИПОВ

Н.Б. КИРЕЕВ

*Институт Информационных Технологий  
Белорусского государственного университета информатики и радиоэлектроники (ИИТ БГУИР)  
ул. Козлова, 28, г. Минск, 220013, Республика Беларусь  
info@webmax.by; www.webmax.by*

При разработке информационных программных систем функциональные требования являются наиболее сложными с точки зрения их выявления, детализации и управления, но именно они определяют архитектуру и влияют на процесс разработки программы. Предлагаемая итерационная методика позволяет выявлять и детализировать функции системы посредством построения визуальных UML-моделей с размещением последних в среде Axure Pro и генерацией динамических HTML-прототипов. Данная методика может быть использована в рамках стандартного унифицированного процесса разработки программного обеспечения, а также при неопределенных и изменяющихся требованиях к системе в условиях «гибкой» методологии «SCRUM».

*Ключевые слова:* модель UML, HTML-прототип, диаграмма вариантов использования.

В основе любой программной системы, а также процесса её разработки лежат требования к ПО. Карл Вигерс в [1] предложил их следующую классификацию (рис. 1).

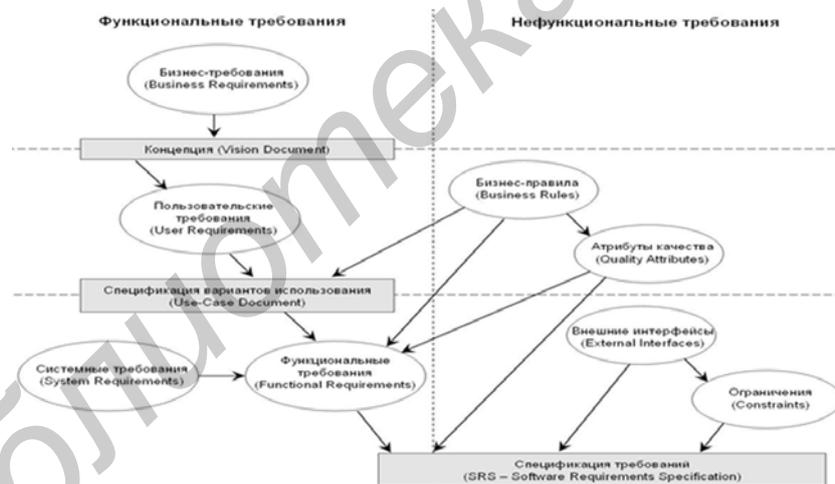


Рис. 1. Классификация требований к ПО, предложенная К. Вигерсом

При этом к группе функциональных требований относятся:

*Бизнес-требования (Business Requirements)* – определяют высокоуровневые цели организации/клиента (потребителя, заказчика разрабатываемого ПО) или его бизнеса;

*Пользовательские требования (User Requirements)* – описывают цели/задачи пользователей системы, которые должны достигаться/выполняться пользователями в рамках поддерживаемого бизнес-процесса.

*Функциональные требования (Functional Requirements)* – определяют функциональность (поведение) программной системы, которая должна обеспечить пользователям исполнение их обязанностей в контексте пользовательских требований.

В приведенной выше классификации, модель вариантов использования создается на основе пользовательских требований, соответственно, на их основе производится разработка архитектуры системы. Данный подход является классическим, но обладает существенным недостатком. Часто такие варианты использования оказываются слишком абстрактными и с трудом могут быть использованы для дальнейшего проектирования архитектуры системы.

Другой подход – создавать модель вариантов использования на основе функциональных сервисов (функций бизнес-логики), которые иницируются через графический интерфейс пользователя (GUI). Проектировать архитектуру, реализующую конкретную функциональность предпочтительнее, чем для более абстрактных пользовательских требований. Однако, до внедрения методик и программ, генерирующих динамические прототипы GUI, сложность такого подхода состояла в том, что этот функционал разрабатывался и тестировался на этапе кодирования. Использование программных продуктов, таких как Axure Pro 6.5 или Prototyper 5.6, позволяет уже на этапе анализа требований детализировать и согласовать с заказчиком функции бизнес-логики, иницируемые GUI разрабатываемой системы, и на их основе проектировать ее архитектуру.

При использовании гибкой модели процесса разработки Scrum прототипы экранных форм могут быть разработаны в самом начале процесса и использоваться для составления product backlog'a – некоего аналога списка требований.

Возможности моделирования функциональности, структур данных и процессов, одновременно с разработкой прототипов экранных форм, позволяют разработчикам выявлять скрытую функциональность, детально прорабатывать и согласовывать с пользователями потоки событий в сценариях взаимодействия с программной системой.

Иерархическая браузерная организационная структура проекта, создаваемого в программном средстве Axure Pro 6.5, позволяет вводить пользовательские виджеты, соответствующие нотации языка визуального моделирования UML и строить диаграммы на отдельных страницах проекта, которые не включаются в HTML-прототип. Кроме того, благодаря простому механизму UML-диаграммы, созданные в других приложениях (Magic Draw UML, Visio, Rational Rose и т.д.), можно легко интегрировать непосредственно в HTML-прототип, например, для пояснения сценария взаимодействия пользователя и программной системы.

Таким образом, визуальное моделирование с одновременным созданием динамических HTML-прототипов, является мощным средством анализа программных систем, позволяющим не только определить функционал, но и детализировать его, а также согласовать с пользователями сценарии взаимодействия.

#### Список литературы

1. Карл И. Вигерс Разработка требований к программному обеспечению «Русская Редакция». М., 2004.