

## Разработка через сценарии

Е. А. Рубцов, email: rea@inistek.ru

Московский авиационный институт (национальный исследовательский университет), кафедра 319

***Аннотация.** Работа посвящена описанию предлагаемого автором метода разработки программного обеспечения через сценарии, сочетающего в себе элементы таких методов разработки, как объектное моделирование через варианты использования с применением UML, трехэтапное проектирование (концептуальное, логическое, физическое), итеративная модель разработки, предметно-ориентированное проектирование, псевдокод.*

***Ключевые слова:** методы разработки программного обеспечения, варианты использования, сценарии использования, модель процесса разработки.*

### Введение

В настоящее время существует множество подходов к разработке программного обеспечения. Некоторые из них касаются организации процесса разработки в целом, некоторые больше сфокусированы на одном или нескольких этапах разработки. Одна из ключевых задач таких подходов – организация качественного перехода от концепции программного обеспечения к его физической реализации.

Данная работа посвящена описанию предлагаемого автором метода разработки программного обеспечения, в основе которого лежит идея последовательного написания сценариев на всех этапах трехэтапной модели проектирования [5].

На этапе концептуального проектирования варианты и сценарии использования помогают выявить потребности целевой аудитории, ответить на вопрос «что программа должна позволять делать пользователям» и «как программа должна позволять это делать с точки зрения пользователей». Описание применения сценариев использования на этапе анализа требований можно найти в [4].

На этапе логического проектирования после разбиения программы на логические части предлагается при помощи сценариев описать порядок взаимодействия этих частей друг с другом.

На этапе физического проектирования предлагается использовать сценарии для описания последовательности вызовов функций, относящихся к верхним уровням абстракции.

К плюсам предлагаемого подхода можно отнести получение на выходе отслеживаемых цепочек переходов от вариантов использования программы через сценарии трех уровней к конкретным фрагментам кода с возможностью обратного перехода от фрагментов кода к вариантам использования. Так же к плюсам можно отнести упрощение процессов написания кода и проверки качества программного обеспечения.

К минусам можно отнести затраты времени на написание сценариев и необходимость наличия у разработчиков навыков написания и чтения таких сценариев.

За последние полгода предлагаемый метод был использован для создания нескольких веб-приложений.

Метод разработки через сценарии может выступить дополнением к методу разработки через тестирование, к минусам которого можно отнести отсутствие предварительно проработанной проектной документации.

## 1. Модель процесса разработки

Предлагаемая в данной работе модель процесса разработки программного обеспечения состоит из нескольких этапов.

1. Выявление проблемы, которую должно решить программное обеспечение.

2. Выявление целевой аудитории, для которой разрабатывается программное обеспечение.

3. Выявление вариантов использования разрабатываемого программного обеспечения.

4. Разработка **сценариев использования** для выявленных вариантов использования программного обеспечения.

5. Разработка **сценариев взаимодействия** для выявленных вариантов использования программного обеспечения на основании сценариев использования.

6. Разработка **сценариев выполнения** для выявленных вариантов использования программного обеспечения на основании сценариев взаимодействия.

7. Реализация сценариев выполнения.

8. Валидация программного обеспечения.

9. Верификация программного обеспечения.

10. Дальнейшее развитие программного обеспечения путем повторения шагов с 3-го по 9-й.

Первый шаг, который рекомендуется сделать в самом начале процесса разработки – сформулировать проблему, которую программное обеспечение должно решать. Определение проблемы – это описание сути проблемы с пользовательской точки зрения без каких-либо намеков на возможные пути ее решения [1]. На последующих этапах важно удерживать внимание на решаемой проблеме.

## **2. Варианты использования**

Для описания представителей целевой аудитории и вариантов использования ими программного обеспечения предлагается использовать диаграммы вариантов использования, входящих в состав набора диаграмм унифицированного языка моделирования [2]. При помощи диаграмм вариантов использования можно наглядно отобразить информацию, перечисленную ниже.

1. Представители целевой аудитории (акторы, действующие лица).
2. Основные варианты использования.
3. Связи представителей целевой аудитории с основными вариантами использования.
4. Включаемые варианты использования.
5. Расширяющие варианты использования.
6. Отношения обобщения между представителями целевой аудитории.
7. Отношения обобщения между вариантами использования.

В первую очередь необходимо выявить варианты использования, которые можно включить в минимально жизнеспособный продукт. В сфере разработки программного обеспечения есть правило: восемьдесят процентов успеха и ценности любой программы заложены в двадцати процентах ее функциональных возможностей [3]. Как следствие, в минимально жизнеспособный продукт рекомендуется включать варианты использования из тех двадцати процентов, реализация которых обеспечит восемьдесят процентов возможностей.

## **3. Сценарии**

В рамках процесса проектирования предлагается использовать три вида сценариев.

1. Сценарии использования – описание программного обеспечения с точки зрения пользователей. Сценарии пишутся в виде диалогов пользователей с программой и отвечают на вопрос «как пользователи используют программу?».

2. Сценарии взаимодействия – описание программного обеспечения с точки зрения проектировщика. Сценарии пишутся в виде общения различных компонентов программы друг с другом и отвечают

на вопрос «как разные части программы взаимодействуют друг с другом?»).

3. Сценарии выполнения – описание программного обеспечения с точки зрения программиста. Сценарии пишутся в виде последовательности вызовов процедур/функций/методов и отвечают на вопрос «как будет работать программа?»).

Каждый из трех видов сценариев соответствует одному из трех этапов проектирования [5]: концептуальное, логическое, физическое. Сценарии использования позволяют определить концепцию взаимодействия пользователей с программой. Сценарии взаимодействия позволяют определить логику взаимодействия различных частей программы друг с другом. Сценарии выполнения позволяют описать структуру программы на физическом уровне. На концептуальном уровне сценарии рекомендуется создавать без учета технологий и языков программирования. На логическом уровне сценарии рекомендуется создавать с учетом выбранных технологий, но без учета языков программирования. На физическом уровне сценарии рекомендуется создавать с учетом выбранных языков программирования.

Последовательная разработка трех видов сценариев помогает проложить «мост» между концепцией программного обеспечения и его физической реализацией, что упрощает процессы реализации, валидации и верификации.

В процессе реализации программисты пишут код, опираясь на сценарии выполнения. Валидация программы выполняется путем проверки возможности решения программой выявленной на первом этапе проблемы [6]. Верификация программы выполняется путем проверки соответствия поведения программы сценариям.

В дальнейшем в рамках предлагаемого метода возможны следующие пути развития программы:

- добавление новых вариантов использования;
- добавление альтернативных сценариев;
- корректировка существующих сценариев.

#### **4. Пример**

В качестве примера рассмотрим этапы процесса разработки через сценарии программы для учета посещаемости занятий студентами.

Этап 1. Выявление проблемы.

Несмотря на то, что в начальной формулировке речь идет о студентах, проблему можно сформулировать, в более общей форме: решение проблемы учета присутствия учащихся на занятиях.

Этап 2. Выявление целевой аудитории.

Целевой аудиторией являются преподаватели, ведущие занятия в группах.

Этап 3. Выявление вариантов использования.

Диаграмма выявленных вариантов использования отображена на рисунке.

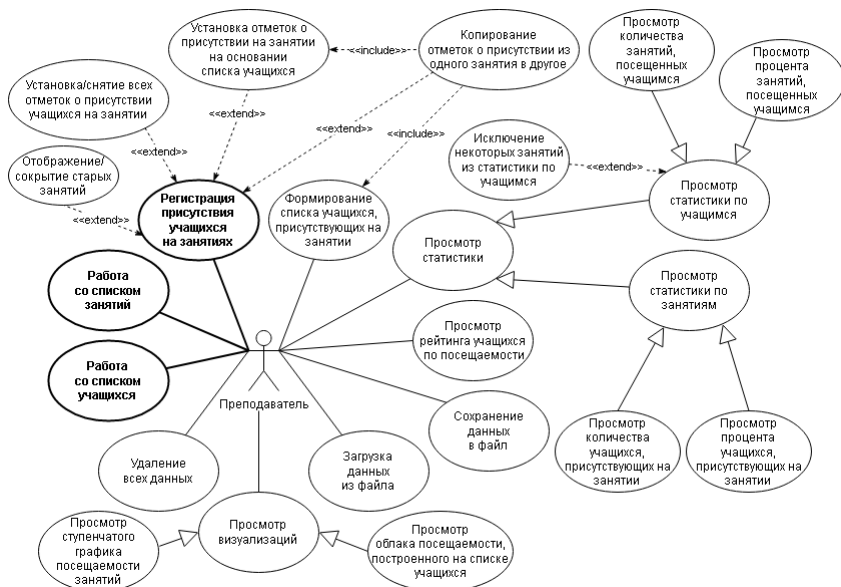


Рисунок. Диаграмма вариантов использования

Выделенные полужирным шрифтом варианты использования вошли в состав минимального жизнеспособного продукта. Далее в примере рассматриваются только эти три варианта использования.

Этап 4. Разработка сценариев использования.

Основной сценарий использования для варианта «Работа со списком учащихся»:

1. Преподаватель переходит в раздел для работы со списком учащихся.
2. Программа отображает раздел для работы со списком учащихся.
3. Преподаватель вводит/корректирует список учащихся.
4. Преподаватель нажимает «Сохранить».
5. Программа сохраняет список учащихся.
6. Программа обновляет данные в разделе для учета посещаемости занятий.

Основной сценарий использования для варианта «Работа со списком занятий»:

1. Преподаватель переходит в раздел для работы со списком занятий.
2. Программа отображает раздел для работы со списком занятий.
3. Преподаватель вводит/корректирует список занятий.
4. Преподаватель нажимает «Сохранить».
5. Программа сохраняет список занятий.
6. Программа обновляет данные в разделе для учета посещаемости занятий.

Основной сценарий использования для варианта «Регистрация присутствия учащихся на занятиях»:

1. Преподаватель переходит в раздел для учета посещаемости.
2. Программа отображает раздел для учета посещаемости.
3. Преподаватель устанавливает/снимает отметки о присутствии учащихся на занятиях.
4. Преподаватель нажимает «Сохранить».
5. Программа сохраняет данные о присутствии учащихся на занятиях.

Этап 5. Разработка сценариев взаимодействия.

В качестве технологий были выбраны web-технологии, но без использования объектно-ориентированного подхода и без использования клиент-серверной архитектуры. В качестве компонентов программы были выбраны три компонента: Пользователь, Браузер, ХранилищеДанных.

Основной сценарий взаимодействия для варианта «Работа со списком учащихся»:

1. Пользователь работает со списком учащихся.
2. Пользователь просит Браузер сохранить список учащихся.
3. Браузер просит ХранилищеДанных сохранить список учащихся.
4. Браузер запрашивает у ХранилищаДанных список учащихся, список занятий, данные о присутствии учащихся на занятиях.
5. Браузер обновляет данные в разделе для учета посещаемости занятий.

Основной сценарий взаимодействия для варианта «Работа со списком занятий»:

1. Пользователь работает со списком занятий.
2. Пользователь просит Браузер сохранить список занятий.
3. Браузер просит ХранилищеДанных сохранить список занятий.
4. Браузер запрашивает у ХранилищаДанных список учащихся, список занятий, данные о присутствии учащихся на занятиях.

5. Браузер обновляет данные в разделе для учета посещаемости занятий.

Основной сценарий взаимодействия для варианта «Регистрация присутствия учащихся на занятиях»:

1. Пользователь работает с разделом для учета посещаемости занятий.

2. Пользователь просит Браузер сохранить информацию о присутствии учащихся на занятиях.

3. Браузер просит ХранилищеДанных сохранить информацию о присутствии учащихся на занятиях.

В случае использования объектно-ориентированного подхода в качестве взаимодействующих частей программы можно было использовать объекты конкретных классов.

Этап 6. Разработка сценариев выполнения.

Для разработки программы был выбран язык программирования JavaScript. Так как разработку программы предполагалось вести без использования клиент-серверной архитектуры, в сценариях выполнения отсутствуют какие-либо скрипты, выполняющиеся на сервере.

Основной сценарий выполнения для варианта «Работа со списком учащихся» отображен в листинге 1.

Листинг 1

#### *Работа со списком учащихся*

```
button.saveStudentList.onclick
  saveStudentList ()
  showAttendanceRecords ()
```

Основной сценарий выполнения для варианта «Работа со списком занятий» отображен в листинге 2.

Листинг 2

#### *Работа со списком занятий*

```
button.saveLessonList.onclick
  saveLessonList ()
  showAttendanceRecords ()
```

Основной сценарий выполнения для варианта «Регистрация присутствия учащихся на занятиях» отображен в листинге 3.

Листинг 3

#### *Регистрация присутствия учащихся на занятиях*

```
button.saveAttendanceRecords.onclick
  saveAttendanceRecords ()
```

Этап 7. Реализация.

Результаты реализации можно посмотреть здесь:  
<http://artemidos.ru/app/attrec>.

Этап 8. Валидация.

Спроектированное и реализованное программное обеспечение позволило решить проблему учета присутствия учащихся на занятиях.

Этап 9. Верификация.

Поведение реализованной программы соответствует разработанным ранее сценариям.

Этап 10. Дальнейшее развитие.

В дальнейшем к трем вошедшим в состав минимального жизнеспособного продукта вариантам использования были добавлены еще несколько вариантов. Со сценариями, описывающими все варианты использования можно ознакомиться здесь:  
<http://artemidos.ru/app/attrec/doc.html>.

### **Заключение**

В рамках данной работы был предложен метод разработки программного обеспечения через сценарии. Была описана модель процесса разработки, были отдельно описаны его этапы, был приведен пример использования метода и предоставлена возможность ознакомиться с результатами его применения.

### **Список литературы**

1. Макконнел, С. Совершенный код. Мастер класс. / С. Макконнел. – М.: Русская редакция, 2010. – 896 с.
2. Unified Modeling Language Specification [Электронный ресурс] : Object Management Group. – Режим доступа: <https://www.omg.org/spec/UML>
3. Сазерленд, Д. Scrum. Революционный метод управления проектами / Д. Сазерленд. – М.: Манн, Иванов и Фебер, 2016. – 288 с.
4. Rosenberg, D. Use Case Driven Object Modeling with UML. Theory and Practice. / D. Rosenberg, M. Stephens. – New York: Apress L. P., 2008. – 440 p.
5. Уилсон, С. Принципы проектирования и разработки программного обеспечения. / С. Уилсон, Б. Мэйплс, Т. Лэндгрейв. – 2-е изд., испр. – М.: Русская Редакция, 2002. – 736 с.
6. ГОСТ Р ИСО/МЭК 15288–2005. Информационная технология. Системная инженерия. Процессы жизненного цикла систем [Текст]. – Введ. 2005-12-29. – М: Стандартинформа, 2006. – 53 с.