УДК 004.021

# Поиск причин дефектов при разработке веб-приложения в рамках AFD-1

**Е. С. Григорьев**

evgen-grit@mail.ru

ФГБОУ ВО «Уфимский государственный авиационный технический университет» (УГАТУ)

**Аннотация.** В данной статье рассмотрена задача обеспечения качества программных продуктов с помощью инструментов анализа дефектов в рамках методологии AFD-1. Рассмотрен пример применения диверсионного анализа для устранения дефектов при создании веб-приложения. На основе рассмотренного примера автором предлагается включение диверсионного анализа в качестве инструмента повышения качества программных продуктов.

**Ключевые слова:** веб-приложение; диверсионный анализ; Anticipatory Failure Determination; AFD; анализ дефектов.

## ВВЕДЕНИЕ

Четвертая промышленная революция в последние несколько лет привела к увеличению спроса на программные продукты, к их усложнению и увеличению масштабов проектов по их реализации[1]. И хоть программные продукты раньше играли лишь вспомогательную роль, сегодня во многих отраслях невозможно представить эффективное принятие решений без их помощи. Один из способов повышения качества программных продуктов – с помощью проактивного подхода постараться предусмотреть как можно большее число возможных дефектов в программных продуктах еще на стадиях проектирования и реализации в программных кодах.

Говоря о повышении качества программных продуктов с помощью пред усмотрения возможных дефектов, нельзя не упомянуть о модели швейцарского сыра – идее, заключающейся в том, что некоторая последовательность латентных дефектов способна привести к появлению новых дефектов.

Одним из возможных подходов к реализации модели швейцарского сыра является диверсионный анализ (*англ.* Anticipatory Failure Determination, AFD) – методология, позволяющая находить и устранять дефекты. AFD содержит в себе несколько инструментов, среди которых следует отметить два: AFD-1, который предназначен для поиска причин дефектов, которые уже произошли, и AFD-2, который применяется для поиска дефектов, которые еще не дали о себе знать[3]. Например, согласно AFD-1, мы не задаемся вопросом почему именно произошел дефект, но напротив, пытаемся добиться этого дефекта и усилить его, определяем необходимые ресурсы и порядок действий для достижения этого дефекта.

**Обоснование выбора методологии.** Поскольку веб-приложения на сегодняшний день широко распространенное явление из-за высокого спроса на них, при их создании и сопровождении необходимо каким-то образом предусматривать и устранять проявившиеся дефекты. Для анализа дефектов представляется возможным использовать такие методы, как FTA, FMEA и HAZOP. Данные методы предназначены для анализа и контроля качества продуктов и производственных процессов, причем FMEA и HAZOP мало применимы при создании программных продуктов. FTA же несмотря на то, что позволяет детально представить разрабатываемую систему и найти корневые причины дефектов, требует значительных затрат средств и времени, так как увеличение детальности рассматриваемой инфраструктуры приводит к геометрическому увеличению числа влияющих событий.

В отличии от описанных выше методов, AFD-1 позволяет уложить процесс поиска причин дефекта в 7 последовательных шагов. Также, AFD-1 позволяет рассматривать только те события, которые непосредственно влияют на дефект. Поэтому, для описанного ниже случая будем применять 7 шагов AFD-1.

Рассмотрим часть процесса создания веб-приложения, предназначенного для вычисления некоторого статистического показателя на основании выборочных данных. После размещения приложения на сервере, загрузка приложения в окне браузера пользователя происходит значительная временная задержка (более 2 секунд[4]). Также, при нажатии пользователем кнопки выдачи результатов вычислений происходит задержка вывода (более 1.5 секунды[4]).

### ШАГ 1. ПОСТАНОВКА ЗАДАЧИ

Необходимо создать веб-приложение, калькулятор, предназначенный для вычисления некоторого статистического значения на основе входных. Логика работы приложения реализовано на языке JavaScript, предназначенного для работы на клиентской стороне, т.е. после загрузки приложения в окне браузера пользователя обращения к серверу минимальны или полностью отсутствуют.

### ШАГ 2. ОПРЕДЕЛЕНИЕ СЦЕНАРИЕВ УСПЕХА

Определим сценарии успешной работы приложения (табл. 1)

Таблица 1

**Сценарии успеха**

| *Операция* | *Результат* |
|---|---|
| Открытие приложения | Приложение загружено с сервера и отображается в окне браузера |
| Ввод исходных значений | Отображение введенных значений |
| Вывод результатов | Численное значение, полученное в результате вычислений |

### ШАГ 3. ЛОКАЛИЗАЦИЯ ОТКАЗА

Во-первых, при первом открытии приложения в окне браузера клиента временная задержка составляет больше двух секунд. Это происходит не часто, но достаточно, чтобы оказаться заметным. Кроме того, задержка в 2 секунды и более является крайне нежелательной при работе современных веб-сайтов и приложений. Во-вторых, при нажатии кнопки вычисления и вывода результата, происходит временная задержка вывода от полутора секунд и более.

### ШАГ 4. ФОРМУЛИРОВКА И УСИЛЕНИЕ ИНВЕРТИРОВАННОЙ ПРОБЛЕМЫ

4.1 Необходимо создать временную задержку размером не менее двух секунд. Также необходимо обеспечить задержку в выдаче результатов от полутора секунд и более.

4.2 Необходимо обеспечить максимально возможную временную задержку при загрузке приложении и выдаче результатов работы.

### ШАГ 5. ПОИСК РЕШЕНИЙ

#### 5.1 Поиск очевидных решений

Подобное поведение сайтов и веб-приложений замечено в нескольких случаях, среди которых следует выделить наиболее часто встречающиеся:

Низкая пропускная способность сервера, на котором расположен сайт/веб-приложение из-за выбранного на хостинге тарифа [5];

Сетевые ограничения скорости[5];

Территориальная удаленность сервера, на котором располагается приложение[5];

Использование излишнего числа изображений, CSS файлов и скриптов, которые значительно увеличивают общий размер приложения и увеличивают время загрузки.

Оба этих случая, по разным причинам, приводят к одному и тому же результату: медленная загрузка html-страниц, из которых состоит веб-приложение.

Однако они не приводят к задержке вывода результата вычислений. Подобные задержки при вычислениях часто встречаются при большом числе циклов и рекурсий в программном коде, а также при использовании излишнего ОЗУ.

#### 5.2 Определение ресурсов

Нам доступны следующие ресурсы:

- Алгоритм работы приложения;
- Исходные данные;
- Техническое задание на разработку веб-приложения;
- Перечень хостингов и серверами

на них, а также различные тарифные планы на каждом из хостингов;

- Первичные знаний и навыки программирования;

- Первичные знания и навыки тестирования программ.

## ШАГ 6. ФОРМУЛИРОВАНИЕ ГИПОТЕЗ И ПРОЕКТНЫХ ИСПЫТАНИЙ ДЛЯ ПРОВЕРКИ

Гипотеза: при использовании хостинга, по тем или иным причинам не способном обеспечить должный уровень скорости, а также при использовании излишнего числа изображений, CSS и JavaScript файлов, сайт и веб-приложение будет медленно загружаться в окне браузера пользователя. Кроме того, нерациональное использование циклов, рекурсий и использование большого числа переменных и массивов, приводит к появлению значительной временной задержке при выводе результата работы приложения.

Гипотеза легко поддается проверке путем размещения приложения на хостинге, сервера которого находятся на другом континенте, а также путем добавления большого числа изображений и JavaScript файлов.

## ШАГ 7. ИСПРАВЛЕНИЕ ДЕФЕКТА

Выяснив причины появления временных задержек, произведем следующие манипуляции с веб-приложением:

Выберем другой хостинг, способный обеспечить высокий уровень скорости интернет-соединения, либо тариф, с более подходящим набором услуг;

Минимизируем число изображений, CSS и JavaScript файлов.

## ЗАКЛЮЧЕНИЕ

Применение методологии AFD может помочь в обеспечении качества программных продуктов, однако, на сегодняшний день наблюдается малое число наработок и инструментов, позволяющих более эффективно применять идею диверсионного анализа.

Таким образом, диверсионный анализ в дальнейшем может дополнить уже имеющийся инструментарий анализа дефектов, и использоваться для повышения качества программных продуктов, но для этого необходимо дальнейшее развитие методологии и соответствующих инструментов.

## СПИСОК ЛИТЕРАТУРЫ

1. **Шваб К.** Четвертая промышленная революция. Эксмо, 2019. 288 с. [ K. Schwab, *Fourth Industrial Revolution* (in Russian), Eksmo, 2019. ]

2. **Down M., Brozowski L., Hisham Y.** Potential Failure Mode & Effects Analysis FMEA Reference Manual (4th Edition). AIAG; 2008. 151 с. [ M. Down, L. Brozowski, Y. Hisham, *Potential Failure Mode & Effects Analysis FMEA Reference Manual (4th Edition)*, AIAG, 2008 ]

3. **Kaplan S., Visnepolschi S., Zlotin B. and Zusman A.** New Tools for Failure and Risk Analysis: An Introduction to Anticipatory Failure Determination (AFD) and the Theory of Scenario Structuring [Электронный ресурс]. URL:http://whereinnovationbegins.net/build/publication.asp#New_Tools.pdf (дата обращения 1.02.2020). [ S. Kaplan, S. Visnepolschi, B. Zlotin and A**.** Zusman, *New Tools for Failure and Risk Analysis: An Introduction to Anticipatory Failure Determination (AFD) and the Theory of Scenario Structuring*. [Online]. Available: http://whereinnovationbegins.net/build/publication.asp#New_Tools. **]**

4. **Купер А., Рейман Р., Кронин Д., Носсел К.,** Интерфейс. Основы проектирования взаимодействия. изд.: ПИТЕР, 2020, 720с. [ A. Cooper, R. Reimann, D. Cronin, C. Noessel, *About Face: The Essentials of Interaction Design*, (in Russain). “Piter”, 2020. ]

5. **Олифер В. Г., Олифер Н. А.** Компьютерные сети. Принципы, технологии, протокол. изд.: ПИТЕР, 2019. 992 с. **[** V. Olifer and N. Olifer, *Computer Networks: Principles, Technologies and Protocols*, (in Russian). “Piter”, 2019. **]**

## ОБ АВТОРАХ

**ГРИГОРЬЕВ Евгений Сергеевич**, магистрант 1-го курса факультета информатики и робототехники, кафедры технической кибернетики, группы ИПОАС-107м.

## METADATA

**Title:** Using AFD-1 method to search for causes of defects

**Author:** E. S. Grigoryev

**Affiliation:**

Ufa State Aviation Technical University (UGATU), Russia.

**Email:** evgen-grit@mail.ru,

**Language:** Russian.

**Source:** Molodezhnyj Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), no. 1 (22), pp. 45-47, 2020. ISSN 2225-9309 (Print).

**Abstract:** This article discusses the task of ensuring the quality of software products using defect analysis tools in the framework of the AFD-1 methodology. An example of the use of anticipatory failure determination analysis to eliminate defects when building a web application is considered. Based on the considered example, the author proposes the inclusion of anticipatory failure determination as a tool to improve the quality of software products.

**Key words:** web application; Anticipatory Failure Determination; AFD; root-cause analysis.

**About author:**

**GRIGORYEV, Evgenii Sergeevich,** Postgrad. (PhD) Student, Dept. of Automated Systems. Master of Technics & Technology (UGATU, 2010).