

**МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ВЫЧИСЛИТЕЛЬНЫХ МАШИН, КОМПЛЕКСОВ
И КОМПЬЮТЕРНЫХ СЕТЕЙ (ТЕХНИЧЕСКИЕ НАУКИ)**

УДК 004.05:004.514

**МОДЕЛЬ АВТОМАТИЗИРОВАННОГО ПОСТРОЕНИЯ НАВИГАЦИОННОГО ГРАФА
ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА МОБИЛЬНОГО ПРИЛОЖЕНИЯ**

Статья поступила в редакцию 20.05.2020, в окончательном варианте – 04.09.2020.

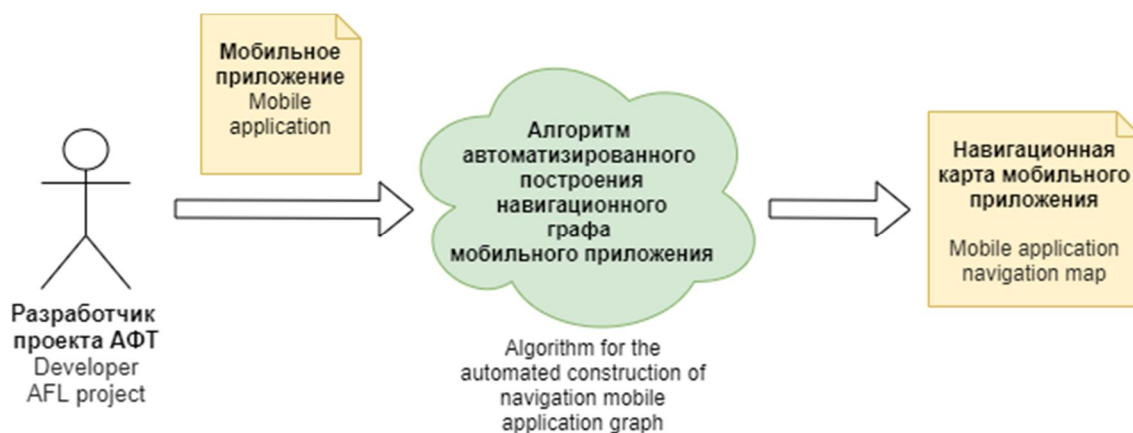
Горкавенко Владимир Сергеевич, Астраханский государственный технический университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 16, аспирант, e-mail: slipmetal@mail.ru

Шуршев Валерий Федорович, Астраханский государственный технический университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 16, доктор технических наук, профессор, e-mail: v.shurshev@mail.ru

Рассмотрен новый подход к созданию наглядной информации о мобильном приложении. Данную информацию можно создать с помощью автоматизированного построения навигационного графа пользовательского интерфейса мобильного приложения при разработке проекта автоматизации функционального тестирования (АФТ). Полученный результат может быть приложен к проектной документации. Актуальность данной работы обусловлена растущим рынком мобильных устройств и приложений для них. В связи с этим растет необходимость проведения качественного автоматизированного тестирования мобильных приложений. На текущий момент в сфере коммерческой разработки мобильных приложений, на начальных этапах разработки проекта автоматизации функционального тестирования (АФТ) пользовательского интерфейса, разработчик проекта автоматизации функционального тестирования, как правило, не имеет полной информации о тестируемом программном обеспечении, чтобы начать покрывать его функциональность автоматизированными скриптами тестирования. В частности, заказчиком не предоставляется достаточная информация о том, как логически реализованы переходы от одной информационной единицы на экране к другой, какие элементы находятся на экране и с какими из них пользователь может взаимодействовать, а также как реализована навигация в приложении. В работе представлено описание структуры абстрактного пользовательского интерфейса мобильного приложения, обосновано понятие логической связи экранов мобильного приложения. Формализована структура экранов и их взаимодействие между собой, описан алгоритм автоматизированного построения навигационного графа пользовательского интерфейса мобильного приложения. Определена мера сложности пользовательского интерфейса мобильного приложения. Это дает возможность перейти к реализации программного обеспечения, которое будет выполнять автоматизированное построение каталога пользовательского интерфейса мобильного приложения.

Ключевые слова: пользовательский интерфейс, UI, функциональное тестирование, автоматизация тестирования, тестирование UI, контроль качества, мобильное приложение, каталог интерфейса, навигационный граф

Графическая аннотация (Graphical annotation)



MODEL OF AUTOMATED BUILDING OF THE NAVIGATION GRAPH OF THE USER INTERFACE OF THE MOBILE APPLICATION

The article was received by the editorial board on 20.05.2020, in the final version – 04.09.2020.

Gorkavenko Vladimir S., Astrakhan State Technical University, 16 Tatishchev St., Astrakhan, 414056, Russian Federation,

graduate student, e-mail: slipmetal@mail.ru

Shurshev Valery F., Astrakhan State Technical University, 16 Tatishchev St., Astrakhan, 414056, Russian Federation,

Doct. Sci. (Engineering), Professor, e-mail: v.shurshev@mail.ru

A new approach to creating visual information about a mobile application is considered. This information can be created using the automated construction of the navigation graph of the user interface of a mobile application when developing a functional testing automation project. The result can be attached to the design documentation. The relevance of this work is due to the growing market for mobile devices and applications for them. In this regard, the need for high-quality automated testing of mobile applications is growing. At the moment in the field of commercial development of mobile applications, at the initial stages of development of a user interface functional testing automation project, the functional testing automation project developer, as a rule, does not have complete information about the software being tested in order to start covering its functionality with automated test scripts. In particular, the customer does not provide sufficient information about how the transitions from one information unit on the screen to another are logically implemented, which elements are on the screen and which of them the user can interact with, as well as how navigation is implemented in the application. The paper presents a description of the structure of an abstract user interface of a mobile application, substantiates the concept of a logical connection between screens of a mobile application. The structure of the screens and their interaction with each other are formalized, an algorithm for the automated construction of the navigation graph of the user interface of a mobile application is described. A measure of the complexity of the user interface of a mobile application has been determined. This makes it possible to proceed to the implementation of software that will perform automated building of the mobile application user interface catalog.

Keywords: user interface, UI, functional testing, test automation, UI testing, quality control, mobile app, interface catalog, navigation graph

Введение. На текущий момент информационные технологии развиваются вместе с растущими потребностями современного человека. Многие повседневные и необходимые действия, связанные с профессиональной деятельностью, личной коммуникацией, а также получением новой информации, как правило, выполняются через программное обеспечение (ПО) на смартфоне или на компьютере: просмотр новостных сводок, ведение личной переписки, выполнение видеозвонков, оплата счетов за ЖКХ, за обучение и т.д.

Сложность разработки и поддержки ПО, которое позволяет выполнять вышеперечисленные действия, также отражается на процессах оценки качества программного продукта. Оценка качества является одним из важнейших этапов в процессе разработки ПО, так как в процессе его выполнения производится проверка соответствия требований к разрабатываемому ПО и тем, что было реализовано.

Различным аспектам этой проблемы посвящено достаточно много работ. В статье [1] рассматриваются основные проблемы автоматизации тестирования: демонстрация соответствия требованиям, выявление ситуаций нежелательного поведения программы. Автор предлагает их решить с помощью генетических алгоритмов. В работе [2] достаточно подробно рассмотрены различные этапы работы, необходимые для построения системы автоматизированного тестирования ядра библиотеки, предназначенной для узкопрофильной операционной системы. Автором предложены различные подходы для создания тестовых приложений. В материале статьи [6] отражен математический подход к созданию специфических тестовых сценариев на основе реальных пользовательских сценариев использования приложения. В работе [12] рассмотрена задача оценки издержек и потерь, связанных со степенью защищенности внешних параметров объекта, которая достигается в том числе за счет регулярного проведения мероприятий по тестированию мер противодействия угрозам. В статье [14] автор рассматривает подход автоматизации процесса тестирования программного обеспечения с использованием метода разбиения по эквивалентности и анализа граничных значений, а также средства модульного тестирования JUnit.

Однако малоисследованным остается вопрос автоматизированного создания наглядной информации о мобильном приложении, в частности каталога пользовательского интерфейса.

Общая характеристика проблематики работы. Одним из этапов процесса оценки качества программного продукта является этап проведения тестирования [10, 14, 15]. Тестирование – это

проверка качества программного продукта, т.е. его точности и корректности реализации набора согласованных и утвержденных требований к программному продукту [17].

Согласно статистике, в 2019 году 63,5 % всех пользователей мобильных телефонов получают доступ к интернету со своего смартфона. На долю мобильной электронной торговли к 2021 г. будет приходиться 2,32 триллиона долларов, или около 67,2 % от общего объема продаж в сфере электронной коммерции. Диаграмма на рисунке 1 также подтверждает тенденцию к устойчивому росту процента мобильных телефонов среди общего трафика пользователей интернета в мире, что косвенно подтверждает рост использования смартфонов относительно других персональных компьютеров с возможностью использования интернета [16].

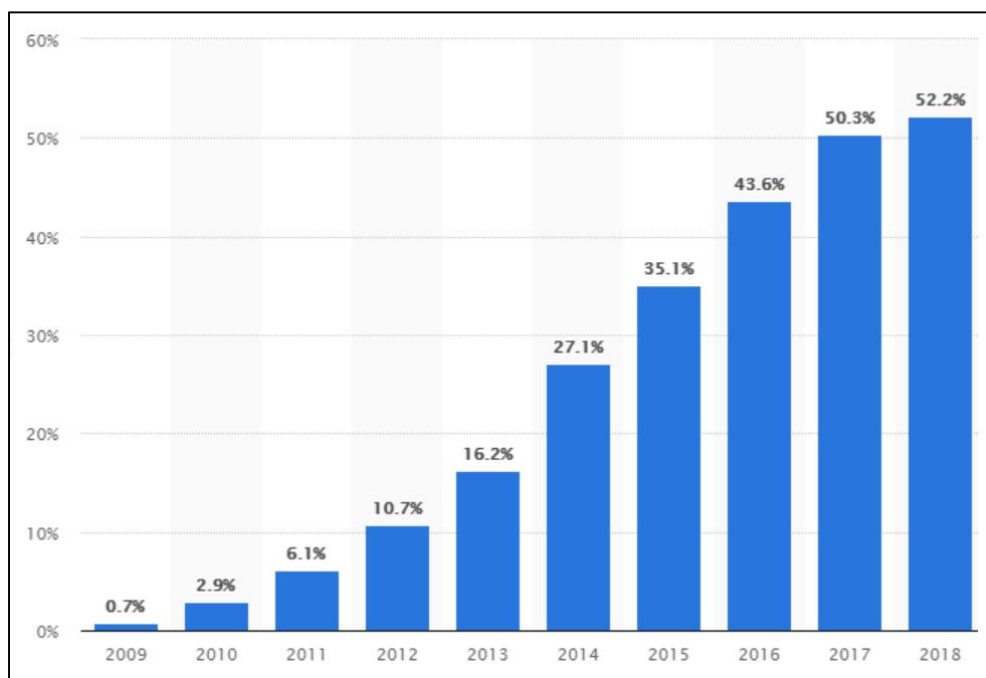


Рисунок 1 – Доля мобильного трафика в мире (по горизонтали – год; по вертикали – процент мобильного интернет-трафика от общего объема трафика)

Вместе с развитием пользовательских мобильных приложений и растущим интересом пользователей к мобильным устройствам также развивается и процесс тестирования. Одним из результатов такого развития стало появление новой сферы тестирования – автоматизированное тестирование ПО. Основными его преимуществами являются:

- повторяемость – любой написанный тест всегда будет выполняться однообразно, то есть исключен «человеческий фактор»;
- быстрое выполнение – автоматизированный скрипт быстрее человека сверяется с инструкциями и документацией, а также выполняет необходимые действия над ПО в процессе выполнения теста, это сильно экономит время выполнения.

Автоматизация тестирования – процесс построения и разработки систем, подсистем, скриптов для реализации выполнения тестовых сценариев с помощью программных средств. На текущий момент автоматизация процесса тестирования является одним из самых перспективных направлений развития информационных технологий в коммерческой разработке программного обеспечения [1–4, 6, 8–10, 12]. Существует множество видов автоматизированного тестирования, одним из которых является UI-тестирование [7]. Данный вид тестирования также называется сквозным, так как при его реализации специалист по тестированию или программное средство взаимодействуют с тестируемым продуктом на самом высоком уровне – на уровне User Interface (UI). Суть такого тестирования заключается в том, что аудитор пытается найти достаточно подтверждений тому, что структура и содержание описанной в проектной документации системы соответствует ее структуре и содержанию на практике [1–3].

На текущий момент в сфере коммерческой разработки мобильных приложений, на начальных этапах разработки проекта автоматизации функционального тестирования (АФТ) пользовательского интерфейса, разработчик проекта АФТ, как правило, не имеет полной информации

о тестируемом ПО, чтобы начать покрывать его функциональность автоматизированными скриптами тестирования. В частности, заказчиком не предоставляется достаточная информация о том, как логически реализованы переходы от одной информационной единицы на экране к другой, какие элементы находятся на экране и с какими из них пользователь может взаимодействовать, а также как реализована навигация в приложении [1–4, 6, 8–10].

Как правило, документация, которую предоставляет заказчик проекта АФТ, не дает подобной структурированной информации. Одним из возможных решений может являться разработка алгоритма построения иерархии экранов пользовательского интерфейса мобильного приложения. На данный момент ПО (в том числе платного), способного выполнять описанные выше действия, не было найдено.

В связи с этим конкретной *целью данной работы* стала разработка модели автоматизированного построения навигационного графа пользовательского интерфейса мобильного приложения. Разработке моделей различных предметных областей, связанных с данной темой, также посвящены работы [7, 11, 18].

Описание структуры абстрактного пользовательского интерфейса мобильного приложения. Существует большое количество мобильных приложений, которые сильно отличаются друг от друга. Это связано прежде всего с тем, что они направлены на различную пользовательскую аудиторию с различными потребностями. Есть простые приложения с упрощенным UI, которые имеют только одно представление (View). В то же время есть приложения с гораздо более сложной и структурированной навигацией и несколькими View [19–21]. Однако можно сказать, что практически любое мобильное приложение состоит из View двух категорий: верхнего уровня и уровня детализации. Визуализация такой структуры представлена на рисунке 2.

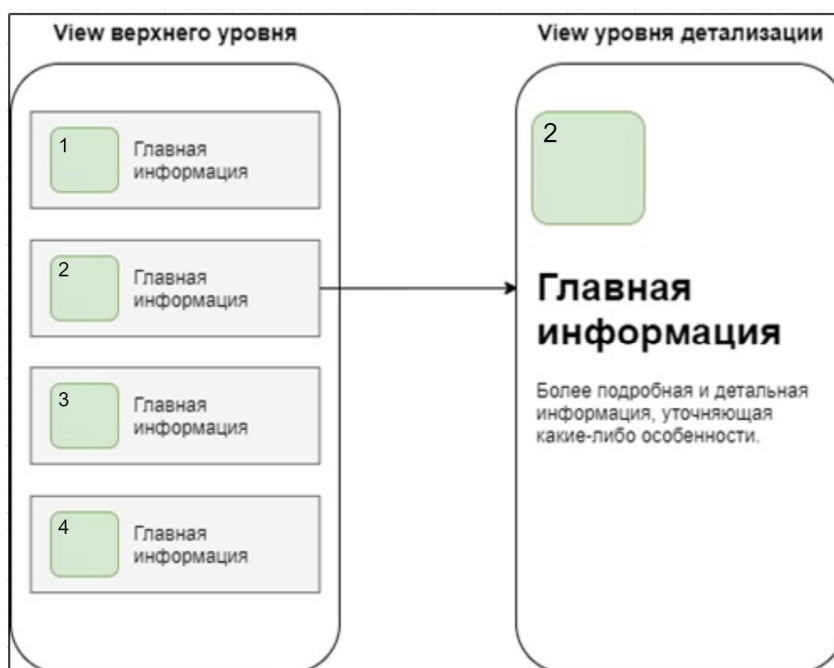


Рисунок 2 – Визуализация структуры мобильного приложения

View верхнего уровня – это область «посадки» мобильного приложения, т.е. первое, что видит пользователь после его запуска. На данном уровне, как правило, находятся элементы экрана, которые отражают главную информацию о каком-либо информационном объекте интереса пользователя. При интерактивном взаимодействии с этим элементом пользователь может получить дополнительную информацию на более низком уровне с точки зрения иерархии структуры пользовательского интерфейса.

View уровня детализации – это представление низкого уровня относительно View верхнего уровня, где пользователь может напрямую взаимодействовать с данными, доступными для просмотра или редактирования. В зависимости от типа компонента, который использует разработчик для отображения подробной информации пользователю, появляется возможность реализовать типовые решения, которые упрощают взаимодействие пользователя с приложением. На этом уровне разработчик может реализовать интуитивное перемещение по приложению, чтобы повысить удобство использования приложения (к примеру, используя типовое решение «смахивания» – Swipe) [19–21].

В связи с вышеизложенным можно утверждать, что View верхнего уровня и уровня детализации представляют из себя логически связанные экраны мобильного приложения различной степени архитектурной сложности (глубины дерева элементов) экрана пользовательского интерфейса, а также навигационной вложенности. Также можно сделать вывод о ограниченности навигационных возможностей мобильного интерфейса, так как одновременно на экране смартфона может отображаться только один экран приложения, в отличие от приложений, разработанных для десктопа.

Логическая связь экранов обуславливается возможностью перехода от одного экрана мобильного приложения к другому посредством интерактивного взаимодействия пользователя с мобильным приложением в целом или с отдельными элементами на экране. Такая логическая связь позволяет каталогизировать экраны мобильного приложения с целью дальнейшего построения иерархии, которая может быть представлена в виде древовидной структуры (на более верхнем уровне – в виде карты переходов мобильного приложения).

Для предполагаемой в будущем программной реализации проекта автоматизации построения навигационного графа экранов мобильного приложения была разработана соответствующая модель.

Характеристика модели. Пусть базовой (корневой) единицей модели является экран – $s_i \in \{Scr\}$. Тогда дочерними единицами экрана являются элементы экрана – $e_j \in s_i, e_j \in \{Elm\}$.

Составим множество $\{Elm\}$ основных элементов экрана, которые встречаются в пользовательском интерфейсе мобильного приложения: e_1 – кнопка; e_2 – текст; e_3 – изображение; e_4 – поле ввода.

Каждый из элементов (e_j) имеет свойства сущности ($p_k \in e_j$). У свойств $\{P\}$ есть параметры значений $\{C\}$. Каждое установленное значение свойства имеет соответствующий ему тип данных.

Составим множество свойств $\{P\}$ элементов $\{Elm\}$ с соответствующим типом параметров значений $\{C\}$:

- уникальный идентификатор (строковый тип);
- текст (название) – строковый тип;
- фактическое представление элемента на экране (булев тип);
- фактическое присутствие в дереве элементов экрана (булев тип);
- доступность действия $\{A\}$:
 - a_1 нажатие (булев тип);
 - a_2 ввод символов (булев тип).

Действия $\{A\}$ вызывают соответствующие события $\{O\}$.

При выполнении действия a_1 происходит одно или несколько событий ($a_1(e_j) \rightarrow \{O\}$):

- изменение свойств элементов, расположенных на текущем экране;
- переход на другой экран.

При выполнении действия a_2 происходит одно или несколько событий ($a_2(e_j) \rightarrow \{O\}$): изменение свойств элементов, расположенных на текущем экране.

Формализованная структура экранов мобильного приложения представлена в графическом виде на рисунке 3.

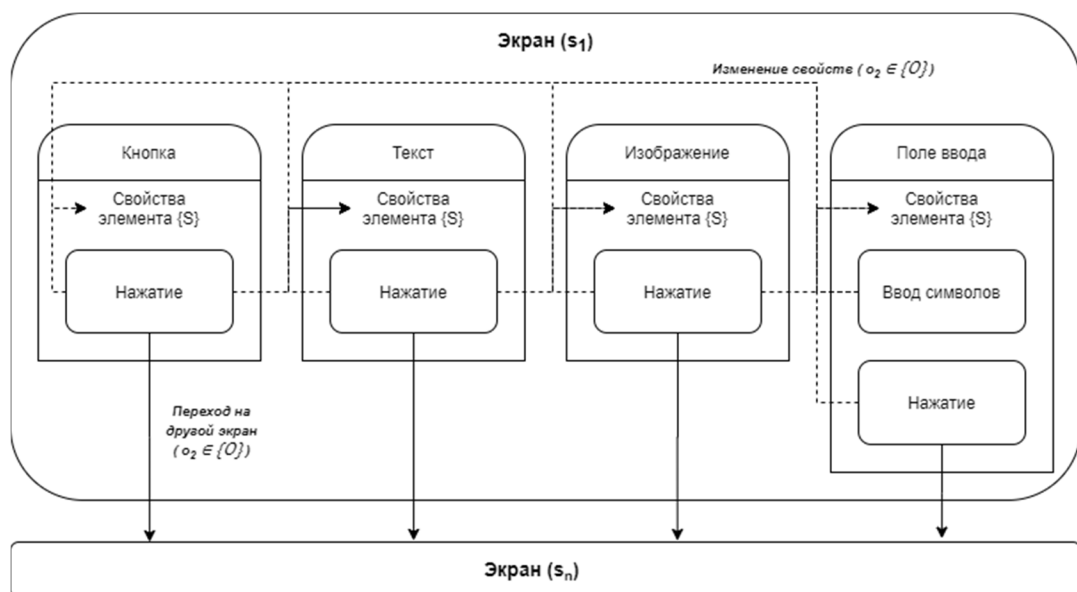


Рисунок 3 – Графическое представление структуры экранов мобильного приложения

Модель, выполняющую построение навигационного графа пользовательского интерфейса мобильного приложения, можно представить в следующем виде (рис. 4).



Рисунок 4 – Модель автоматизированного построения навигационного графа UI в виде «черного ящика»

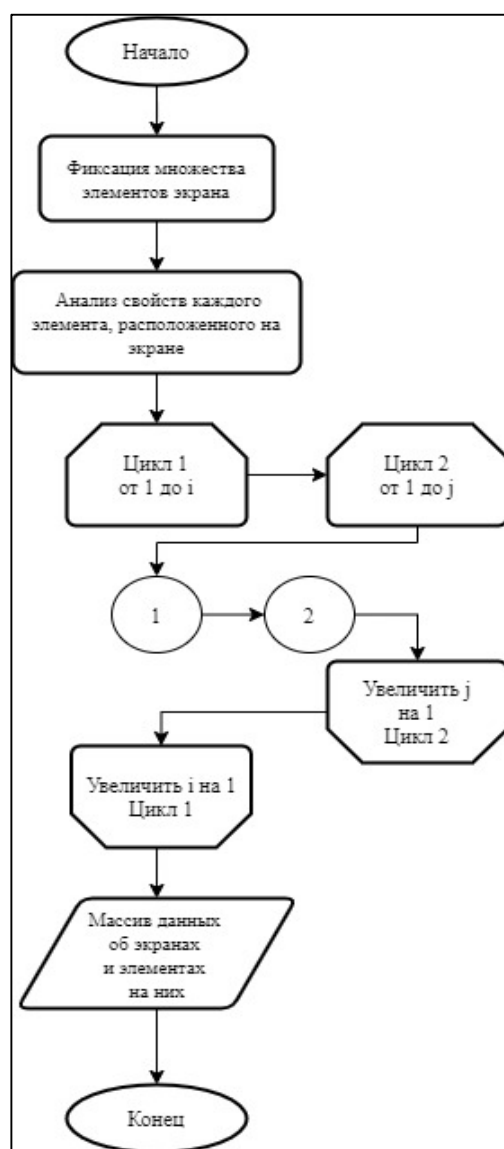


Рисунок 5.1 – Блок-схема этапов построения навигационного графа пользовательского интерфейса мобильного приложения (часть 1)

На рисунке 4 входным значением является множество экранов $\{Scr\}$, а выходным – результирующий массив данных (R), содержащий информацию о логических переходах между экранами, а также о свойствах элементов и действий, которые влияют на их изменение. Чтобы получить

результатирующий массив, необходимо, чтобы в «черном ящике» автоматизировано выполнялся алгоритм преобразования информации, которую содержат экраны. Так как на вход «черного ящика» подается массив данных, то необходимо, чтобы алгоритм автоматизированного построения навигационного графа позволял бы обработать все элементы массива.

Тогда каждую его итерацию можно представить в виде последовательности следующих этапов:

- 1) фиксация множества $\{Elm\}$ и анализ свойств $\{P\}$ каждого элемента e_j , расположенного на экране s_i ;
- 2) последовательное выполнение доступных действий из множества $\{A\}$ над каждым элементом экрана e_j ;
- 3) фиксация множества событий $\{O\}$ после выполнения действия a_i ;
- 4) анализ каждого произошедшего события o_i :
 - если произошедшее событие o_i – это переход к другому экрану, то логически связать экран s_i , на котором выполнялось действие a_i , с экраном s_j (с обязательным указанием через какой элемент e_j выполнялся переход);
 - иначе – логически связать событие o_i с элементами, свойства которых изменились после того, как оно произошло (с обязательным указанием того, какой элемент e_j и действие над ним a_i вызвало это событие).

Описанный выше алгоритм можно представить в виде блок-схемы (рис. 5.1–5.3). Результатом его выполнения будет являться массив данных, который хранит в себе информацию о логических переходах между экранами, а также о свойствах элементов и действий, которые влияют на их изменение.

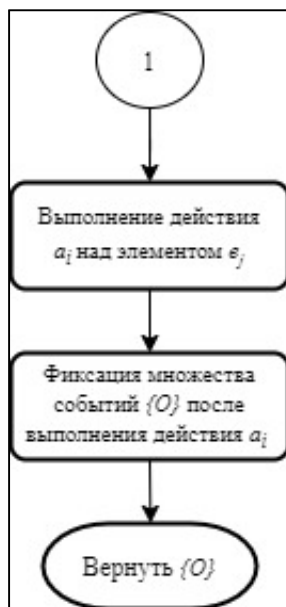


Рисунок 5.2 – Продолжение блок-схемы этапов построения навигационного графа пользовательского интерфейса мобильного приложения (часть 2)

Такой массив можно представить в виде графовой структуры. Согласно определению [5], графом $G(V, E)$ называется совокупность двух множеств – непустого множества V (множества вершин) и множества E его двухэлементных подмножеств множества V (E – множество ребер). Данное определение можно представить в следующем виде (1):

$$G(V, E) = [V; E], V \neq \emptyset, E \subset 2^V. \quad (1)$$

При этом вершинами V будут являться экраны $\{Scr\}$ мобильного приложения, а ребрами E будут являться выполненные действия a_i над элементами $e_i \in s_i$, $e_i \in \{Elm\}$, которые приводят к множеству событий $\{O\}$, каждое o_i из которых может содержать как изменение свойств элемента $e \in s_i$ без перехода на другой экран, так и непосредственный переход на экран $s_j \in \{Scr\}$. Так как все действия $\{A\}$ являются однонаправленными и могут порождать петли, можно сделать вывод, что такой граф будет являться сильно связным ориентированным псевдографом.

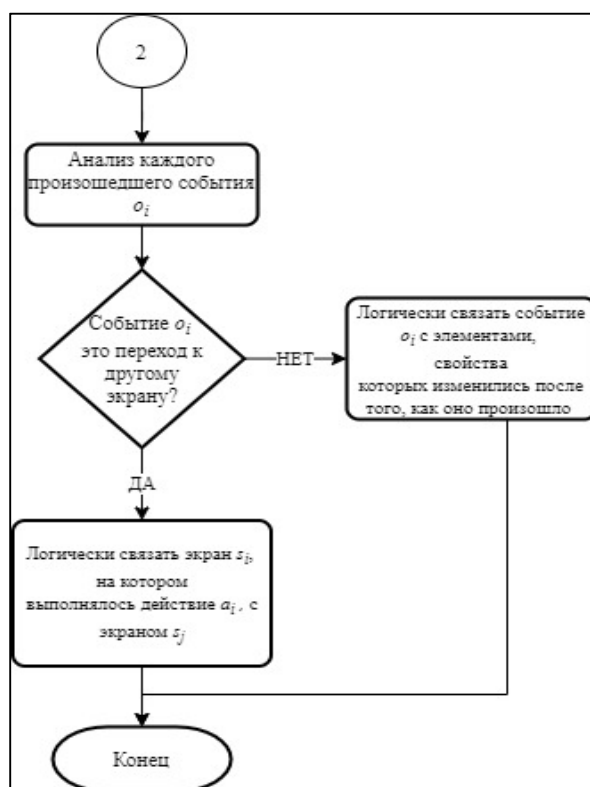


Рисунок 5.3 – Продолжение блок-схемы этапов построения навигационного графа пользовательского интерфейса мобильного приложения (часть 3)

Графически такую структуру можно представить в следующем виде (рис. 6).

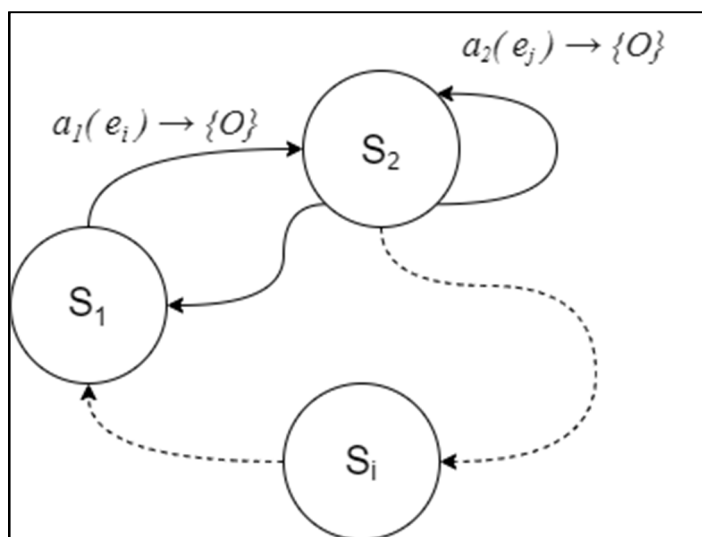


Рисунок 6 – Навигационный граф пользовательского интерфейса мобильного приложения

Полученный в результате автоматизированной обработки мобильного приложения навигационный граф пользовательского интерфейса возможно использовать как «карту переходов». Также результат выполнения возможно приложить к документации тестируемого мобильного приложения как один из проектных артефактов.

С помощью полученного графа становится возможным определение сложности пользовательского интерфейса мобильного приложения. В качестве основы для такого показателя можно использовать разработанную Томасом Дж. Маккейбом структурную (топологическую) меру – «Цикломатическая сложность программы».

Определение сложности интерфейса мобильного приложения. Так как данная мера используется для вычисления сложности части программного кода, необходимо адаптировать ее определение для вычисления сложности пользовательского интерфейса мобильного приложения.

Согласно [23], цикломатическая сложность части программного кода – это количество линейно независимых маршрутов через программный код. Математически цикломатическая сложность программы определяется с помощью ориентированного графа, узлами которого являются блоки программы, соединенные рёбрами, если управление может переходить с одного блока на другой. Тогда сложность программного кода определяется как (2):

$$M = E - N + 2P, \quad (2)$$

где M – цикломатическая сложность; E – количество рёбер в графе; N – количество узлов в графе; P – количество компонент связности.

Для сильносвязного графа (когда точка выхода соединена с точкой входа) сложность будет определяться как (3):

$$M = E - N + P. \quad (3)$$

Исходя из вышесказанного, можно сделать вывод, что цикломатическая сложность пользовательского интерфейса мобильного приложения – это количество линейно независимых маршрутов через пользовательский интерфейс мобильного приложения. Тогда сложность пользовательского интерфейса в отдельном независимом пользовательском бизнес-процессе приложения будет определяться как (4):

$$M_{ui} = |\{A\}| - |\{Scr\}| + 2P, \quad (4)$$

где M_{ui} – цикломатическая сложность пользовательского интерфейса; $|\{A\}|$ – количество действий над элементами, приведших к какому-либо событию; $|\{Scr\}|$ – количество экранов мобильного приложения; P – количество логических входов в отдельный независимый пользовательский бизнес-процесс.

Автоматизированный сбор информации об экранах мобильного приложения – это циклический и замкнутый процесс, где точкой входа и выхода является один и тот же экран. Так как результатом выполнения алгоритма построения навигационного графа пользовательского интерфейса мобильного приложения является сильносвязный ориентированный граф, справедливо утверждать, что сложность пользовательского интерфейса мобильного приложения будет определяться как (5):

$$M_{ui} = |\{A\}| - |\{Scr\}| + 1, \quad (5)$$

Предположим, имеется мобильное приложение, состоящее из 3 экранов, на экранах S_1 , S_2 , S_3 имеется по 2 элемента. Нажатие на каждый из этих элементов на соответствующих экранах приводит к событию перехода на один из экранов или изменению свойства элемента на текущем. Описанные входные данные можно представить в следующем виде (6):

$$\begin{aligned} \{Scr\} &= \{S_1, S_2, S_3\}, \\ |\{Scr\}| &= 3, \\ e_1, e_2 &\in S_1, \\ e_3, e_4 &\in S_2, \\ e_5, e_6 &\in S_3, \\ a_1(e_1) &\rightarrow \{S_2\}, \\ a_1(e_2) &\rightarrow \{S_3\}, \\ a_1(e_3) &\rightarrow \{S_1\}, \\ a_1(e_4) &\rightarrow \{S_3\}, \\ a_1(e_5) &\rightarrow \{S_1\}, \\ a_2(e_6) &\rightarrow \{S_3\} \\ |\{A\}| &= 6. \end{aligned} \quad (6)$$

Согласно данным выше, возможно построить ориентированный сильносвязный граф (рис. 7).

Сложность пользовательского интерфейса мобильного приложения из примера будет равна:

$$M_{ui} = 6 - 3 + 1 = 4.$$

Полученный результат является приемлемым, согласно предложениям МакКейба [23], так как не превышает 10.

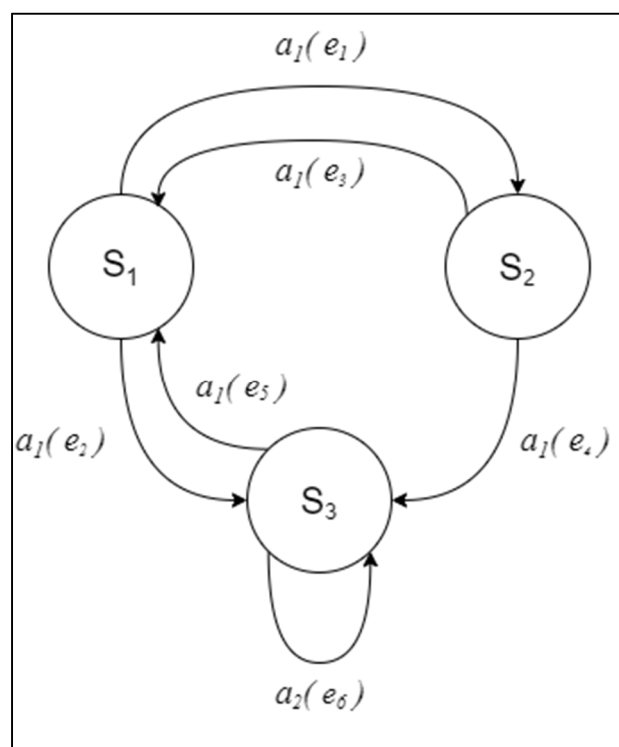


Рисунок 7 – Навигационный граф пользовательского интерфейса мобильного приложения на основе данных из примера

Вывод. В работе описана и визуализирована структура абстрактного пользовательского интерфейса мобильных приложений, рассмотрены основные компоненты, обоснована логическая связь экранов. Это позволило перейти к формализации модели и построению графического представления структуры экранов. Представлена модель автоматизированной каталогизации UI в виде «черного ящика». Эта модель содержит поэтапный алгоритм построения навигационного графа пользовательского интерфейса. Благодаря этому удалось применить метод оценки сложности интерфейса мобильного приложения на примере описанных входных данных. Полученные результаты позволяют применить их непосредственно в процессе реализации проекта автоматизированного тестирования мобильного приложения с целью структурирования информации о том, как реализована навигация в мобильном приложении.

Разработанная модель позволяет перейти к проектированию и реализации ПО, выполняющего автоматизированное построение каталога пользовательского интерфейса мобильного приложения.

Библиографический список

1. Артюхова А. С. Проблемы автоматизации тестирования и подходы к их решению / А. С. Артюхова // CETERIS PARIBUS. – 2016. – № 10. – Режим доступа: <https://cyberleninka.ru/article/n/problemy-avtomatizatsii-testirovaniya-i-podhody-k-ih-resheniyu>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 19.05.2020).
2. Барладян Б. Х. Проблемы автоматизации тестирования программной реализации OpenGL Sc / Б. Х. Барладян, Л. З. Шапиро, А. А. Хлупина // Новые информационные технологии в автоматизированных системах. – 2018. – № 21. – Режим доступа: <https://cyberleninka.ru/article/n/problemy-avtomatizatsii-testirovaniya-programmnoy-realizatsii-opengl-sc>, (дата обращения: 19.05.2020).
3. Бедняк С. Г., Тиханов А. А. Автоматизация процессов тестирования web-приложений // МНИЖ. 2013. – № 11-1 (18). – Режим доступа: <https://cyberleninka.ru/article/n/avtomatizatsiya-protsssov-testirovaniya-web-prilozheniy>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 19.05.2020).
4. Веселов А. О. Автоматизация тестирования телекоммуникационных приложений / А. О. Веселов, А. С. Иванов, Б. В. Тютин, В. П. Котляров // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика, телекоммуникации и управление. – 2009. – № 3 (80). – Режим доступа: <https://cyberleninka.ru/article/n/avtomatizatsiya-testirovaniya-telekommunikatsionnyh-prilozheniy>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 25.04.2020).
5. Гладких О. Б. Основные понятия теории графов / О. Б. Гладких, О. Н. Белых. – Елец : ЕГУ им. И.А. Бунина, 2008. – 175 с.
6. Дробинцев П. Д. Автоматизация тестирования на основе покрытия пользовательских сценариев / П. Д. Дробинцев, В. П. Котляров, И. Г. Черноруцкий // Научно-технические ведомости Санкт-Петербургского

государственного политехнического университета. Информатика, телекоммуникации и управление. – 2012. – № 4 (152). – Режим доступа: <https://cyberleninka.ru/article/n/avtomatizatsiya-testirovaniya-na-osnove-pokrytiya-polzovatel'skih-stsenariiev>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 19.05.2020).

7. Иконников В. В. Разработка системы автоматизации регрессионного тестирования / В. В. Иконников, А. А. Лебедев // Объектные системы. – 2011. – № 1 (3). – Режим доступа: <https://cyberleninka.ru/article/n/razrabotka-sistemy-avtomatizatsii-regressionnogo-testirovaniya>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 19.05.2020).

8. Квятковский К. И. Проектирование информационных систем для органов государственной власти / К. И. Квятковский, В. Ф. Шуршев // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. – 2011. – № 1. – Режим доступа: <https://cyberleninka.ru/article/n/proektirovanie-informatsionnyh-sistem-dlya-organov-gosudarstvennoy-vlasti>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 25.04.2020).

9. Ким В. Ю. Особенности разработки дизайна пользовательского интерфейса для мобильного приложения / В. Ю. Ким // Новые информационные технологии в автоматизированных системах. – 2015. – № 18. – Режим доступа: <https://cyberleninka.ru/article/n/osobennosti-razrabotki-dizayna-polzovatel'skogo-interfeysa-dlya-mobilnogo-prilozheniya>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 25.04.2020).

10. Колмогоров К. А. Роль автоматического тестирования в процессе разработки программного обеспечения / К. А. Колмогоров // КИО. – 2006. – № 3. – Режим доступа: <https://cyberleninka.ru/article/n/rol-avtomaticheskogo-testirovaniya-v-protseesse-razrabotki-programmnogo-obespecheniya>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 19.05.2020).

11. Кравченко П. П. Автоматизация тестирования программных интерфейсов приложений: метод и средства / Кравченко П. П., Бирюков С. В. // Известия ЮФУ. Технические науки. – 2012. – № 3. – Режим доступа: <https://cyberleninka.ru/article/n/avtomatizatsiya-testirovaniya-programmnyh-interfeysov-prilozheniy-metod-i-sredstva>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 19.05.2020).

12. Магомедов Ш. Г. Построение моделей описания рисков охранных действий по защите внешних периметров организации / Ш. Г. Магомедов, В. Ф. Шуршев, Г. А. Попов, А. Ф. Дорохов, М. Ф. Руденко // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. – 2017. – № 3. – Режим доступа: <https://cyberleninka.ru/article/n/postroenie-modeley-opisaniya-riskov-ohrannyh-deystviy-po-zaschite-vneshnih-perimetrov-organizatsii>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 19.05.2020).

13. Мартюкова Е. С. О внедрении процесса автоматизации тестирования в различных методологиях разработки программного обеспечения / Е. С. Мартюкова // Новые информационные технологии в автоматизированных системах. – 2015. – № 18. – Режим доступа: <https://cyberleninka.ru/article/n/o-vnedrenii-protseessa-avtomatizatsii-testirovaniya-v-razlichnyh-metodologiyah-razrabotki-programmnogo-obespecheniya>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 19.05.2020).

14. Нагаев Р. А. Автоматизация процесса тестирования программного обеспечения с применением JUnit / Р. А. Нагаев, И. С. Полевщиков // Вестник науки и творчества. – 2016. – № 5 (5). – Режим доступа: <https://cyberleninka.ru/article/n/avtomatizatsiya-protseessa-testirovaniya-programmnogo-obespecheniya-s-primeneniem-junit>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 19.05.2020).

15. Блэк Рэкс. Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование / Блэк Рэкс. – Москва : Лори, 2011. – 544 с.

16. Стерлягов С. П. Применения User Experience/User Interface (UX/UI) в разработке мобильных приложений / С. П. Стерлягов, Е. П. Селютина // Международный научно-исследовательский журнал. – 2017. – № 08 (62), ч. 3. – С. 69–73. – Режим доступа: <https://research-journal.org/technical/primenenie-user-experienceuser-interface-modelirovaniya-dlya-razrabotki-mobilnogo-prilozheniya/>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 25.04.2020).

17. Статистика Интернета 2019: трафик, сайты и блоги, домены, социальные медиа, онлайн-реклама и электронная коммерция – свежие цифры и факты // sdvv.ru – Режим доступа: <https://sdvv.ru/articles/elektronnaya-kommertsiya/statistika-interneta-2019-trafik-sayty-i-blogi-domeny-sotsialnye-media-onlayn-reklama-i-elektronnaya/>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 12.05.2020).

18. Тестирование. Фундаментальная теория // habr.com. – Режим доступа: <https://habr.com/ru/post/279535/>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 12.05.2020).

19. Шуршев В. Ф. Автоматизированная система обработки информации во время формирования топологии сети мониторинга природной территории / В. Ф. Шуршев, Э. Р. Чишиев, А. А. Сорокин, Л. В. Галимова, В. Н. Есауленко // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. – 2017. – № 3. – Режим доступа: <https://cyberleninka.ru/article/n/avtomatizirovannaya-sistema-obrabotki-informatsii-vo-vremya-formirovaniya-topologii-seti-monitoringa-prirodnoy-territorii>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 19.05.2020).

20. Francesco Azzola. Android UI Design Basics / Francesco Azzola. – Java Code Geeks, 2016. – 132 p.

21. Lee Copeland. A Practitioner's Guide to Software Test Design. – Artech House, 2004. – 300 p.

22. Ron Patton. Software Testing / Ron Patton. – Sams Publishing, 2005. – 408p.

23. McCabe. A Complexity Measure / McCabe // IEEE Transactions on Software Engineering. – 1976. – December. – P. 308–320.

References

1. Artyukhova A. S. Problemy avtomatizatsii testirovaniya i podkhody k ikh resheniyu [Test automation problems and approaches to their solution]. *CETERIS PARIBUS*, 2016, no. 10. Available at: <https://cyberleninka.ru/article/n/problemy-avtomatizatsii-testirovaniya-i-podhody-k-ih-resheniyu> (accessed 19.05.2020).
2. Barladyan B. H., Shapiro L. Z., Khlupina A. A. Problemy avtomatizatsii testirovaniya programmnoy realizatsii OpenGL Sc [Test automation problems for software implementation of OpenGL Sc]. *Novye informatsionnye tekhnologii v avtomatizirovannykh sistemakh* [New information technologies in automated systems], 2018, no. 21. Available at: <https://cyberleninka.ru/article/n/problemy-avtomatizatsii-testirovaniya-programmnoy-realizatsii-opengl-sc> (accessed 19.05.2020).
3. Bednyak S. G., Tikhonov A. A. Avtomatizatsiya protsessov testirovaniya web-prilozheniy [Automation of web application testing processes]. *Mezhdunarodnyy nauchno-bssledovatel'skiy zhurnal* [International Research Journal], 2013, no. 11–1 (18). Available at: <https://cyberleninka.ru/article/n/avtomatizatsiya-protsessov-testirovaniya-web-prilozheniy> (accessed 19.05.2020).
4. Veselov A. O., Ivanov A. S., Tyutin B. V., Kotlyarov V. P. Avtomatizatsiya testirovaniya telekommunikatsionnykh prilozheniy [Automation of testing telecommunication applications]. *Nauchno-tekhnicheskie vedomosti Sankt-Peterburgskogo gosudarstvennogo politekhnicheskogo universiteta. Informatika, telekommunikatsii i upravlenie* [Scientific and technical bulletin of the St. Petersburg State Polytechnic University. Computer science, telecommunications and management], 2009, no. 3 (80). Available at: <https://cyberleninka.ru/article/n/avtomatizatsiya-testirovaniya-telekommunikatsionnykh-prilozheniy> (accessed 25.04.2020).
5. Gladkikh O. B., Belykh O. N. *Osnovnye ponyatiya teorii grafov* [Basic concepts of graph theory]. Yelets, Yelets State University named after I.A. Bunin, 2008. 175 p.
6. Drobintsev P. D., Kotlyarov V. P., Chernoruckiy I. G. Avtomatizatsiya testirovaniya na osnove pokrytiya polzovatel'skikh stsensariyev [Automation of testing based on the coverage of user scenarios]. *Nauchno-tekhnicheskie vedomosti Sankt-Peterburgskogo gosudarstvennogo politekhnicheskogo universiteta. Informatika, telekommunikatsii i upravlenie* [Scientific and technical statements of the St. Petersburg State Polytechnic University. Computer science, telecommunications and management], 2012, no. 4 (152). Available at: <https://cyberleninka.ru/article/n/avtomatizatsiya-testirovaniya-na-osnove-pokrytiya-polzovatel'skikh-stsenariyev> (accessed 19.05.2020).
7. Ikonnikov V. V., Lebedev A. A. Razrabotka sistemy avtomatizatsii regressionnogo testirovaniya [Development of a regression testing automation system]. *Obektnye sistemy* [Object systems], 2011, no. 1 (3). Available at: <https://cyberleninka.ru/article/n/razrabotka-sistemy-avtomatizatsii-regressionnogo-testirovaniya> (accessed 19.05.2020).
8. Kvyatkovskiy K. I., Shurshev V. F. Proektirovanie informatsionnykh sistem dlya organov gosudarstvennoy vlasti [Designing information systems for government authorities]. *Vestnik Asrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika i informatika* [Bulletin of ASTU. Series: Management, Computer Engineering and Informatics], 2011, no. 1. Available at: <https://cyberleninka.ru/article/n/proektirovanie-informatsionnykh-sistem-dlya-organov-gosudarstvennoy-vlasti> (accessed 25.04.2020).
9. Kim V. Yu. Osobennosti razrabotki dizayna polzovatel'skogo interfeysa dlya mobilnogo prilozheniya [Features of the development of user interface design for a mobile application]. *Novye informatsionnye tekhnologii v avtomatizirovannykh sistemakh* [New information technologies in automated systems], 2015, no. 18. Available at: <https://cyberleninka.ru/article/n/osobennosti-razrabotki-dizayna-polzovatel'skogo-interfeysa-dlya-mobilnogo-prilozheniya> (accessed: 25.04.2020).
10. Kolmogorov K. A. Rol avtomaticheskogo testirovaniya v protsesse razrabotki programmnoy obespecheniya [The role of automatic testing in the process of software development]. *Kompyuternye instrumenty v obrazovanii* [Computer instruments in Education], 2006, no 3. Available at: <https://cyberleninka.ru/article/n/rol-avtomaticheskogo-testirovaniya-v-protsesse-razrabotki-programmnoy-obespecheniya> (accessed 19.05.2020).
11. Kravchenko P. P., Biryukov S. V. Avtomatizatsiya testirovaniya programmnykh interfeysov prilozheniy: metod i sredstva [Automation of testing of application programming interfaces: method and means]. *Izvestiya Yuzhnogo federal'nogo universiteta. Tekhnicheskie nauki* [Izvestiya SFedU. Technical science], 2012, no. 3. Available at: <https://cyberleninka.ru/article/n/avtomatizatsiya-testirovaniya-programmnykh-interfeysov-prilozheniy-metod-i-sredstva> (accessed 19.05.2020).
12. Magomedov Sh. G., Shurshev V. F., Popov G. A., Dorokhov A. F., Rudenko M. F. Postroenie modeley opisaniya riskov okhrannykh deystviy po zashchite vneshnykh perimetrov organizatsii [Building models for describing the risks of protective actions to protect the external perimeters of the organization]. *Vestnik Asrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika i informatika* [Bulletin of ASTU. Series: Management, Computer Engineering and Informatics], 2017, no. 3. Available at: <https://cyberleninka.ru/article/n/postroenie-modeley-opisaniya-riskov-okhrannykh-deystviy-po-zaschite-vneshnykh-perimetrov-organizatsii> (accessed 19.05.2020).
13. Martyukova E. S. O vnedrenii protsessa avtomatizatsii testirovaniya v razlichnykh metodologiyakh razrabotki programmnoy obespecheniya [On the implementation of the test automation process in various methodologies of software development]. *Novye informatsionnye tekhnologii v avtomatizirovannykh sistemakh* [New information technologies in automated systems], 2015, no.18. Available at: <https://cyberleninka.ru/article/n/o-vnedrenii-protsessa-avtomatizatsii-testirovaniya-v-razlichnykh-metodologiyakh-razrabotki-programmnoy-obespecheniya> (accessed 19.05.2020).
14. Nagaev R. A., Polevshchikov I. S. Avtomatizatsiya protsessa testirovaniya programmnoy obespecheniya s primeneniem JUnit [Automation of software testing process using JUnit]. *Vestnik nauki i tvorchestva* [Bulletin

of science and creativity], 2016, no. 5 (5). Available at: <https://cyberleninka.ru/article/n/avtomatizatsiya-protsesta-testirovaniya-programmnogo-obespecheniya-s-primeneniem-junit> (accessed 19.05.2020).

15. Blek Reks. *Klyuchevye protsessy testirovaniya. Planirovanie, podgotovka, provedenie, sovershenstvovanie* [Key testing processes. Planning, preparation, implementation, improvement]. Moscow, Lori Publ., 2011. 544 p.

16. Sterlyagov S. P., Selyutina E. P. *Primeneniya User Experience/User Interface (UX/UI) v razrabotke mobilnykh prilozheniy* [Applications of User Experience/User Interface (UX/UI) in the development of mobile applications]. *Mezhdunarodnyy nauchno-issledovatel'skiy zhurnal* [International Research Journal], 2017, no. 08 (62), part 3, pp. 69–73. Available at: <https://research-journal.org/technical/primenenie-user-experienceuser-interface-modelirovaniya-dlya-razrabotki-mobilnogo-prilozheniya/> (accessed 25.04.2020).

17. Statistika Interneta 2019: trafik, sayty i blogi, domeny, socialnye media, onlayn-reklama i elektronnyaya kommertsiya – svezhie tsifry i fakty [Internet statistics 2019: traffic, sites and blogs, domains, social media, online advertising and e-commerce – fresh numbers and facts]. *sdvv.ru*. Available at: <https://sdvv.ru/articles/elektronnaya-kommertsiya/statistika-interneta-2019-trafik-sayty-i-blogi-domeny-sotsialnye-media-onlayn-reklama-i-elektronnaya/> (accessed 12.05.2020).

18. Testirovanie. Fundamentalnaya teoriya [Testing. Fundamental theory]. *habr.com*. Available at: <https://habr.com/ru/post/279535/> (accessed 12.05.2020).

19. Shurshev V. F., Chishiev E. R., Sorokin A. A., Galimova L. V., Esaulenko V. N. *Avtomatizirovannaya sistema obrabotki informatsii vo vremya formirovaniya topologii seti monitoringa prirodnoy territorii* [Automated information processing system during the formation of the topology of the natural area monitoring network]. *Vestnik Asrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika i informatika* [Bulletin of ASTU. Series: Management, Computer Engineering and Informatics], 2017, no. 3. Available at: <https://cyberleninka.ru/article/n/avtomatizirovannaya-sistema-obrabotki-informatsii-vo-vremya-formirovaniya-topologii-seti-monitoringa-prirodnoy-territorii> (accessed 19.05.2020).

20. Francesco Azzola. *Android UI Design Basics*. Java Code Geeks, 2016. 132 p.

21. Lee Copeland. *A Practitioner's Guide to Software Test Design*. Artech House, 2004. 300 p.

22. Ron Patton. *Software Testing*. Sams Publishing, 2005. 408 p.

23. McCabe. A Complexity Measure. *IEEE Transactions on Software Engineering*, 1976, December, pp. 308–320.