МИНИСТЕРСТВО СВЯЗИ И ИНФОРМАТИЗАЦИИ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«БЕЛОРУССКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ СВЯЗИ»

Обособленное подразделение «Колледж информационно-коммуникационных технологий»

ОТЧЁТ

О ПРОХОЖДЕНИИ ПРЕДДИПЛОМНОЙ ПРАКТИКИ

|  |  |  |  |
| --- | --- | --- | --- |
| Ф.И.О. учащегося |  | | |
|  |  | | |
| Группа |  | Подпись учащегося |  |

|  |  |
| --- | --- |
| Наименование предприятия |  |
| Сроки практики |  |

Руководитель практики

от организации \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (Ф.И.О.)

М.П.

Дифференцированный зачет по практике \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

(отметка) (дата)

Руководитель практики от колледжа \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (Ф.И.О.)

Минск 2023

СОДЕРЖАНИЕ

[Введение 3](#_Toc136282127)

[1 Вопросы трудового законодательства, охраны труда и техники безопасности 4](#_Toc136282128)

[1.1 Общие требования по охране труда 4](#_Toc136282129)

[1.2 Нормативная документация по организации группы ТО РУП «Белтелеком» 7](#_Toc136282130)

[2 Знакомство со структурой базовой организации 9](#_Toc136282131)

[2.1 Организационная структура РУП «Белтелеком» и выбранного подразделения 9](#_Toc136282132)

[2.2 Политика РУП «Белтелеком», миссия и философия 10](#_Toc136282133)

[3 Изучение процессов тестирования и разработки программного обеспечения 14](#_Toc136282134)

[4 Тестирование документации и требований 16](#_Toc136282135)

[5 Тестирование программного обеспечения 20](#_Toc136282136)

[6 Чек-листы, тест-кейсы, наборы тест-кейсов 24](#_Toc136282137)

[6.1 Чек-лист 24](#_Toc136282138)

[6.2 Тест-кейс 25](#_Toc136282139)

[6.3 Набор тест-кейсов 27](#_Toc136282140)

[7 Отчеты о дефектах 29](#_Toc136282141)

[8 Автоматизация тестирования 34](#_Toc136282142)

[9 Индивидуальное задание 36](#_Toc136282143)

[Заключение 37](#_Toc136282144)

[Приложение А](#_Toc136282145) [Тест-план для мобильного приложения «Белтелеком» 38](#_Toc136282146)

[Приложение Б](#_Toc136282147) [Тест-кейсы для тестирования пользовательского интерфейса 41](#_Toc136282148)

[Приложение В](#_Toc136282149) [Тест-кейсы для тестирования удобства использования 43](#_Toc136282150)

[Приложение Г](#_Toc136282151) [Тест-кейсы для расширенного тестирования 44](#_Toc136282152)

[Приложение Д](#_Toc136282153) [Добавление тестовой документации в github репозиторий 49](#_Toc136282154)

## ВВЕДЕНИЕ

В современном мире очень важную роль играют информационные технологии. Они внедряются буквально во все сферы деятельности человека. Большие компании стараются автоматизировать с помощью программного обеспечения как можно больше процессов для экономии времени и ресурсов.

Не удивительно, что для пользования всеми возможностями различного программного обеспечения необходим доступ к сети Интернет. В Беларуси одним из наикрупнейших поставщиков услуг электросвязи является РУП «Белтелеком».

Республиканское унитарное предприятие «Белтелеком» – оператор Республики Беларусь по предоставлению услуг электросвязи.

Создано 3 июля 1995 г. в форме Республиканского государственного объединения. 1 августа 2004 г. преобразовано в Республиканское унитарное предприятие (РУП) с преобразованием предприятий, входивших в состав объединения, в филиалы путём присоединения.

2 сентября 2022 года «Белтелеком» стал единым оператором республиканской системы мониторинга общественной безопасности согласно указу № 69 от 25 февраля 2022 года. В соответствии с указом «Белтелеком» отвечает за создание и функционирование республиканской системы мониторинга общественной безопасности, а также хранение полученной информации.

Целью данной технологической практики является усвоение и применение всех знаний, приобретенных в ходе обучения по специальности.

Для проведения тестирования и выполнения индивидуального задания было выбрано мобильное приложение «Белтелеком».

Внедрение различных технологий благоприятно влияет не только на улучшение экономических показателей и расширение фонда активных абонентов, но и на повышение авторитета среди других лидирующих филиалов большого предприятия РУП «Белтелеком».

Созданное программное обеспечение для абонентов должно соответствовать наивысшим стандартам качества и не нести урон репутации предприятия. Лучший способ повысить качество продукта – протестировать его. Именно это является целью преддипломной практики.

## 1 ВОПРОСЫ ТРУДОВОГО ЗАКОНОДАТЕЛЬСТВА, ОХРАНЫ ТРУДА И ТЕХНИКИ БЕЗОПАСНОСТИ

## 1.1 Общие требования по охране труда

Техника безопасности – это деятельность по предотвращению [несчастных случаев](https://laboratoria.by/stati/neschastnyj-sluchaj-na-proizvodstve), оценке и снижению рисков, связанных с человеческими ошибками.

Все работники на предприятии должны выполнять требования пожарной безопасности на предприятии, знать план эвакуации, а также местонахождение первичных средств: пожаротушения и оповещения и связи, уметь использовать их.

Работник обязан:

− поддерживать свое рабочее место, оборудование и приспособления в исправном состоянии, порядке и чистоте;

− немедленно сообщать работодателю о любой ситуации, угрожающей жизни или здоровью работающих и окружающих, несчастном случае, произошедшем на производстве, оказывать содействие работодателю в принятии мер по оказанию необходимой помощи потерпевшим и доставке их в организацию здравоохранения;

− проходить в установленном законодательством порядке медицинские осмотры, обучение, стажировку, инструктаж и проверку знаний по вопросам охраны труда;

− не загромождать пути эвакуации;

− следить за безопасной работой компьютерного и осветительного оборудования;

− услышав сигнал пожарной сигнализации или сигнал оповещения о пожаре, срочно отключить компьютер от сети, взять верхнюю одежду, личные вещи и покинуть здание согласно плану эвакуации, прибыть к месту сбора, сообщить об этом руководителю подразделения;

− заботиться о личной безопасности и личном здоровье, а также о безопасности окружающих в процессе выполнения работ либо во время нахождения на территории организации.

В помещениях предприятия и на рабочих места запрещается:

− включать в электросеть одновременно два и более электронагревательных прибора (электрочайник, микроволновая печь, обогреватель);

− использовать электронагревательные приборы, не имеющие устройств тепловой защиты (автоматического отключения), при отсутствии в них или неисправности терморегуляторов;

− оставлять без надзора включенные в электросеть электроприборы (в том числе и ПЭВМ), кроме электрооборудования непрерывной эксплуатации.

Эксплуатирующий средства вычислительной техники и периферийное оборудование персонал может подвергаться опасным и вредным воздействиям, которые по природе действия подразделяются на следующие группы:

– поражение электрическим током;

– механические повреждения;

– электромагнитное излучение;

– инфракрасное излучение;

– опасность пожара;

– повышенный уровень шума и вибрации.

Для снижения или предотвращения влияния опасных и вредных факторов необходимо соблюдать Санитарные правила и нормы, гигиенические требования к видео дисплейным терминалам, персональным электронно-вычислительным машинам и организации работы:

– на рабочем месте монитор должен устанавливаться таким образом, чтобы исключить возможность отражения от его экрана в сторону пользователя источников общего освещения помещения;

– расстояние от экрана монитора до глаз пользователя должно составлять от 50 до 70 см;

– монитор должен быть установлен прямо перед пользователем и не требовать поворота головы или корпуса тела;

– рабочий стол и посадочное место должны иметь такую высоту, чтобы уровень глаз пользователя находился чуть выше центра монитора. На экран монитора следует смотреть сверху вниз, а не наоборот;

– рабочий стол и посадочное место должны иметь такую высоту, чтобы уровень глаз пользователя находился чуть выше центра монитора. На экран монитора следует смотреть сверху вниз, а не наоборот;

– при работе с компьютерной мышью рука не должна находиться на весу. Локоть руки или хотя бы запястье должны иметь твердую опору. Если предусмотреть необходимое расположение рабочего стола и кресла затруднительно, рекомендуется применить коврик для мыши, имеющий специальный опорный валик;

– недопустимо размещать на системном блоке, мониторе и периферийных устройствах посторонние предметы;

– все питающие кабели и провода должны располагаться с задней стороны компьютера и периферийных устройств. Их размещение в рабочей зоне пользователя недопустимо;

– запрещается производить какие-либо операции, связанные с подключением, отключением или перемещением компонентов компьютерной системы без предварительного отключения питания;

– компьютер не следует устанавливать вблизи электронагревательных приборов и систем отопления.

Наниматель обязан:

– рационально использовать труд работников;

– обеспечивать производственно-технологическую, исполнительскую и трудовую дисциплину;

– вести учет фактически отработанного работником времени;

– выплачивать заработную плату в сроки и в размерах, установленных законодательством, коллективным договором, соглашением или трудовым договором;

– обеспечивать на каждом рабочем месте условия труда, соответствующие требованиям по охране труда, соблюдать установленные нормативными правовыми актами, в том числе техническими нормативными правовыми актами, требования по охране труда, а при отсутствии в нормативных правовых актах, в том числе в технических нормативных правовых актах, требований по охране труда принимать необходимые меры, обеспечивающие сохранение жизни, здоровья и работоспособности работников в процессе трудовой деятельности;

– принимать необходимые меры по профилактике производственного травматизма, профессиональных и других заболеваний работников; контролировать знание и соблюдение работниками требований инструкций по охране труда и пожарной безопасности; своевременно и правильно проводить расследование и учет несчастных случаев на производстве;

– обеспечивать работников в соответствии с установленными нормами специальной одеждой, специальной обувью и другими средствами индивидуальной защиты, организовывать надлежащее хранение и уход за этими средствами;

– коллективными договорами, соглашениями, другими локальными правовыми актами и трудовыми договорами;

– своевременно оформлять изменения в трудовых обязанностях работника и знакомить его с ними под роспись, создавать условия для ознакомления работника с локальными правовыми актами, затрагивающими его права и обязанности;

– обеспечивать профессиональную подготовку, повышение квалификации, переподготовку и стажировку работников в соответствии с законодательством;

– исполнять другие обязанности, вытекающие из законодательства, локальных правовых актов и трудовых договоров.

## **1.2 Нормативная документация по организации группы ТО РУП «Белтелеком»**

Основные нормативные правовые акты по вопросам осуществления административных процедур:

– директива Президента Республики Беларусь от 27 декабря 2006 г. № 2 «О дебюрократизации государственного аппарата и повышении качества обеспечения жизнедеятельности населения»;

– закон Республики Беларусь от 28 октября 2008 г. № 433.3 «Об основах административных процедур»;

– указ Президента Республики Беларусь от 26 апреля 2010 г. № 200 «Об административных процедурах, осуществляемых государственными органами и иными организациями по заявлениям граждан»;

– постановление Совета Министров Республики Беларусь от 17 февраля 2012 г. № 156 «Об утверждении единого перечня административных процедур, осуществляемых государственными органами и иными организациями в отношении юридических лиц и индивидуальных предпринимателей, внесении дополнения в постановление Совета Министров Республики Беларусь от 14 февраля 2009 г. № 193 и признании утратившими силу некоторых постановлений Совета Министров Республики Беларусь»;

– постановление Совета Министров Республики Беларусь от 6 мая 2020 г. № 271 «Об изменении постановлений Совета Министров Республики Беларусь по вопросам осуществления административных процедур»;

– постановление Министерства связи и информатизации Республики Беларусь «О составлении протоколов об административных правонарушениях» от 19.02.2021 № 4.

Основным нормативным документом, определяющим гигиенические требования к организации труда на компьютере, являются Санитарные правила и нормы СанПиН 2.2.2/2.4.1340.03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

Все гигиенические требования к организации труда с применением компьютеров условно можно разделить на три группы:

– требования к помещениям, где организована и осуществляется подобная трудовая деятельность;

– требования к оборудованию индивидуального рабочего или учебного места;

– требования к организации труда с использованием компьютерной техники.

Каждое производственное помещение, в котором осуществляется работа с компьютерами, должно иметь комбинированное, естественное и искусственное освещение. Все подобны производственные помещения должны быть оснащены принудительной приточно-вытяжной вентиляцией и увлажнителями воздуха, заправляемыми ежедневно дистиллированной или прокипяченной питьевой водой.

В течение рабочего дня для всех пользователей компьютеров, непосредственно и постоянно работающих с этой техникой, помимо обязательного обеденного перерыва, устанавливаемого законодательством о труде, «для обеспечения оптимальной работоспособности и сохранения здоровья профессиональных пользователей, на протяжении рабочей смены должны устанавливаться регламентированные перерывы».

Важнейшие гигиенические требования:

– расположите монитор компьютера к окну боком, чтобы свет на него падал слева;

– перед началом и по окончании работы, а также в обеденный перерыв проводите аэрацию (проветривание) помещения, где работает компьютер;

– при непрерывной работе с компьютером каждые 2 часа делайте перерыв на 15 минут для отдыха и выполнения комплекса физкультурно-оздоровительных упражнений;

– при работе с компьютером расстояние от глаз пользователя до монитора должно составлять 600-700 мм, но не менее 500 мм;

– следите за осанкой: спина должны быть прямая, руки в локтях должны быть согнуты под прямым углом.

## 2 ЗНАКОМСТВО СО СТРУКТУРОЙ БАЗОВОЙ ОРГАНИЗАЦИИ

Республиканское унитарное предприятие электросвязи **«Белтелеком»** – ведущая телекоммуникационная компания с многолетней историей, персонал которой обеспечивает и развивает важные для государства, общества, частных и корпоративных клиентов технологии связи.

**«Белтелеком»** был создан 3 июля 1995 года как Республиканское государственное объединение. 1 августа 2004 года компания прошла преобразование в Республиканское унитарное предприятие электросвязи.

## 2.1 Организационная структура РУП «Белтелеком» и выбранного подразделения

Сегодня компания «Белтелеком» включает в себя 9 филиалов и 3 производства в составе головного структурного подразделения предприятия. Филиалы представляют каждую из областей Беларуси и Минск:

– брестский филиал;

– витебский филиал;

– гомельский филиал;

– гродненский филиал;

– могилевский филиал;

– минский филиал;

– филиал «Минская городская телефонная сеть»;

– филиал «Междугородная связь»;

– филиал «Подсобное сельское хозяйство».

Производства в составе головного структурного подразделения:

– международный центр коммутации;

– производство Информационно-расчетный центр;

– производство Минская телефонно-телеграфная станция.

Информационно-производственная практика в организации проходит на производстве ИРЦ, более подробная его структура представлена на  
рисунке 2.1.

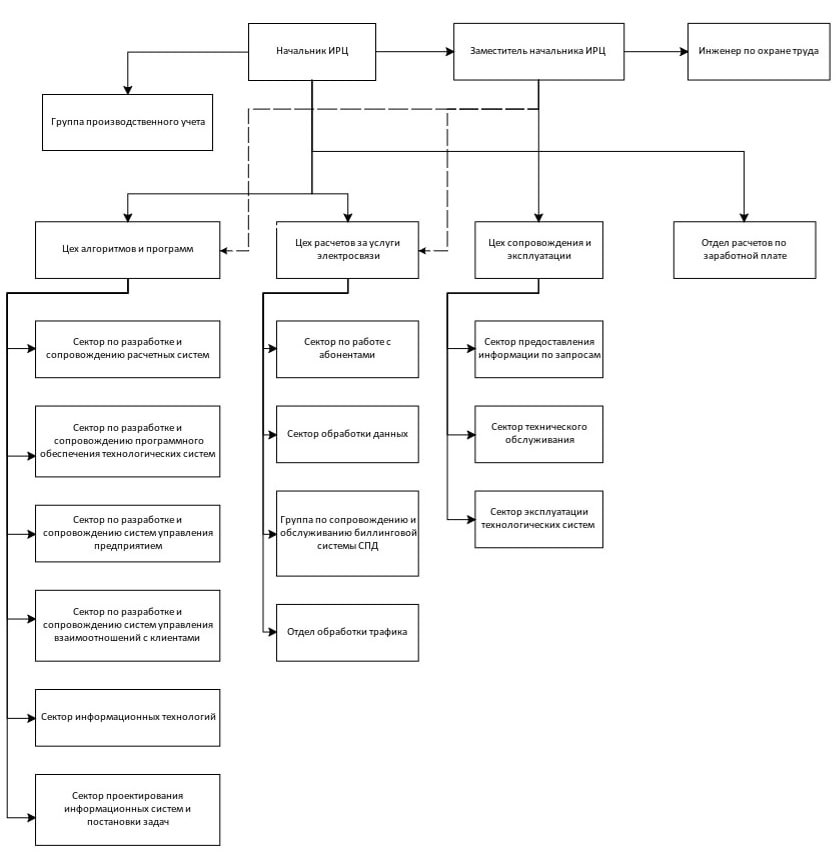
****

Рисунок 2.1 – Организационная структура производства ИРЦ

Непрерывной линией на схеме отображено прямое подчинение, пунктиром – подчинение в вопросах технического обеспечения и технической эксплуатации оборудования.

## 2.2 Политика РУП «Белтелеком», миссия и философия

Для полной характеристики организации необходимо рассмотреть основные принципы, которыми она руководствуется в своей работе: по отношению к клиентам, партнерам, конкурентам, качеству продукта, охране труда и многим другим вопросам. В связи с этим разрабатывается стратегическая цель, для ее достижения ставится ряд задач, которые в свою очередь дополняются правилами (принципами) выполнения. Все это исходит из миссии и философии, без которых не может существовать ни одна уважающая себя организация.

Миссия организации представляет собой основную суть существования, ради чего эта самая организация создавалась.

Философия организации – совокупность внутриорганизационных принципов, моральных и административных норм, систему ценностей и убеждений, разделяемую всеми сотрудниками и посвященную глобальной цели организации.

Стратегическая цель предприятия – обеспечение долгосрочной конкурентоспособности и повышение экономической эффективности компании путем своевременного и полного удовлетворения спроса на современные услуги электросвязи.

Для достижения стратегической цели РУП «Белтелеком» ставит перед собой следующие задачи:

– качественно оказывать услуги электросвязи;

– расширять присутствие компании на рынке за счет внедрения новых услуг, предлагать удобные для потребителей тарифные планы, применять эффективные рыночные стратегии;

– непрерывно улучшать производственную базу компании путем внедрения технологических инноваций и совершенствования бизнес-процессов;

– развивать и укреплять кадровый потенциал компании.

Принципы выполнения поставленных задач:

– применение инструментов результативного менеджмента, в первую очередь системы менеджмента качества, процессного подхода с мышлением на основе рисков;

– соответствие применимым требованиям и постоянное улучшение системы менеджмента качества;

– ориентация на потребителей, выполнение их требований и повышение их удовлетворенности;

– создание высокопроизводительных рабочих мест, развитие компетенций и обеспечение высокого уровня профессионализма персонала компании;

– лидерство руководителей всех уровней, привлечение персонала в процессы повышения качества, точное распределение ответственности и полномочий;

– признание личных заслуг сотрудников, вознаграждение за вклад в создание услуг высокого качества и совершенствование системы менеджмента качества;

– взаимовыгодное долгосрочное сотрудничество с партнерами компании на национальном и международном уровнях.

Высшее руководство РУП «Белтелеком» принимает обязательство неукоснительно следовать изложенным принципам, берет на себя ответственность за обеспечение эффективного управления предприятием, создание необходимых организационных условий, выделение финансовых, кадровых и материальных ресурсов.

Цель политики в области охраны труда:

– проведение планомерной работы по устранению причин производственного травматизма и профессиональных заболеваний;

– внедрение современных технологических решений, направленных на минимизацию производственных рисков.

Основные задачи в области охраны труда:

– создание системы идентификации производственных опасностей, оценки рисков. Обеспечение постоянного контроля над рисками, проведение предупреждающих и корректирующих действий, направленных на снижение вероятности возникновения профессиональных заболеваний и несчастных случаев на производстве;

– регулярный анализ целей и задач в области охраны труда, повышения эффективности проводимых мероприятий с целью постоянного улучшения функционирования системы управления охраной труда;

– соблюдение требований действующего законодательства в области охраны труда, производственной санитарии, пожарной и промышленной безопасности;

– вовлечение всего персонала предприятия в деятельность по охране труда;

– обеспечение информированности заинтересованных сторон.

Пути достижения поставленных задач:

– поддержание в рабочем состоянии на предприятии процедуры для постоянной идентификации опасностей, оценки рисков и обеспечения необходимых мер по контролю в соответствии с требованием СТБ 18001-2009, как основы всей системы управления охраной труда;

– внедрение на предприятии новых современных технологий, проведение своевременного обслуживания и ремонтов технологического оборудования;

– обеспечение финансовой и материально-технической поддержки запланированных мероприятий по охране труда, выполнение требований ТНПА по безопасности и охране труда, приказов и предписаний вышестоящих и контролирующих организаций;

– постоянный анализ законодательной и нормативно-правовой базы в области охраны труда;

– изучение и внедрение передового опыта в области охраны труда;

– взаимодействие и тесное сотрудничество с профессиональными союзами и другими организациями в реализации государственной политики в области охраны труда, содействие общественному контролю за соблюдением прав и интересов работников;

– проведение внутренних проверок для выявления несоответствий и их устранения.

Руководство РУП «Белтелеком» обязуется предоставлять средства и ресурсы для реализации настоящей политики, доводить ее до ведения каждого работника и требовать ее неукоснительного исполнения.

Миссия компании «Белтелеком»: объединять людей, предоставляя свободу общения и получения информации.

Философия компании: «Белтелеком» – лидер в сфере телекоммуникационных услуг.

Слоган, девиз: на связи с миром.

«Мир – это не только планета Земля. Мир – это все, что нас окружает. И в первую очередь – это люди. Люди, с которыми и для которых мы работаем, их чувства, переживания, желания и стремления, их отношение друг к другу и к обществу. Быть на связи с миром – значит быть в согласии с самим собой и тем, что окружает, понимать и принимать друг друга».

Основная цель:

– формирование команды профессионалов, объединенной общими целями;

– качественное предоставление клиентам полного набора современных услуг;

– последовательное расширение своего присутствия на рынке услуг электросвязи.

Ценности:

– уважение, внимание, профессиональное развитие с сотрудниками;

– доброжелательность и взаимоуважение с потребителями;

– надежность и открытость с клиентами;

– этичность с конкурентами;

– открытость и честность в своих обязательствах с государством.

Для организации главной ценностью являются люди, как работающие на предприятии, так и те, для кого работают.

Во главе отношений с клиентами – их интересы. Организация стремится к тому, чтобы любой клиент, обратившийся к ней, нашел заинтересованный отклик на его потребности, получил содействие и помощь в решении любых вопросов. Центральное место в отношении с клиентами занимает команда компетентных, образованных и коммуникабельных сотрудников.

Политика в отношении партнеров и конкурентов:

– ориентация на долгосрочные взаимовыгодные отношения;

– открытость для партнерства и взаимодействия как в рамках отдельных проектов, так и на долгосрочную перспективу;

– уважительное отношение к нашим конкурентам, добросовестная конкуренция;

– честные и открытые отношения с конкурентами и партнерами.

## 3 ИЗУЧЕНИЕ ПРОЦЕССОВ ТЕСТИРОВАНИЯ И РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Тестирование программного обеспечения представляет процесс проверки того, насколько поведение реального программного обеспечения соответствует требованиям и ожиданиям. Проводимые для этого тесты основаны на задокументированных требованиях заказчика.

Разработка программного обеспечения – это процесс разработки, определения, проектирования, программирования, документирования, тестирования и исправления ошибок, связанных с созданием и обслуживанием приложений, сред или других компонентов программного обеспечения.

В первую очередь стоит отметить, что процесс тестирования программного обеспечения тесно связан непосредственно с процессом разработки. Разработка через тестирование предлагает больше, чем просто проверку корректности, она также влияет на дизайн программы. Изначально сфокусировавшись на тестах, проще представить, какая функциональность необходима пользователю. Таким образом, разработчик продумывает детали интерфейса до реализации.

Тесты заставляют делать свой код более приспособленным для тестирования. Например, отказываться от глобальных переменных, одиночек, делать классы менее связанными и легкими для использования.

Жизненный цикл разработки состоит из следующих этапов:

– анализ требований;

– дизайн;

– разработка;

– тестирование и дебаггинг;

– эксплуатация и поддержка.

Жизненный цикл тестирования программного обеспечения (рисунок 3.1) является процессом, которого нельзя избежать. Он непрерывен, продолжителен и требует наличия команды QA, достаточно опытной для того, чтобы произвести полный цикл тестирования. Эта неотъемлемая часть современного процесса разработки программного обеспечения помогает заказчику, команде разработчиков, а также конечному пользователю получить продукт высокого качества.

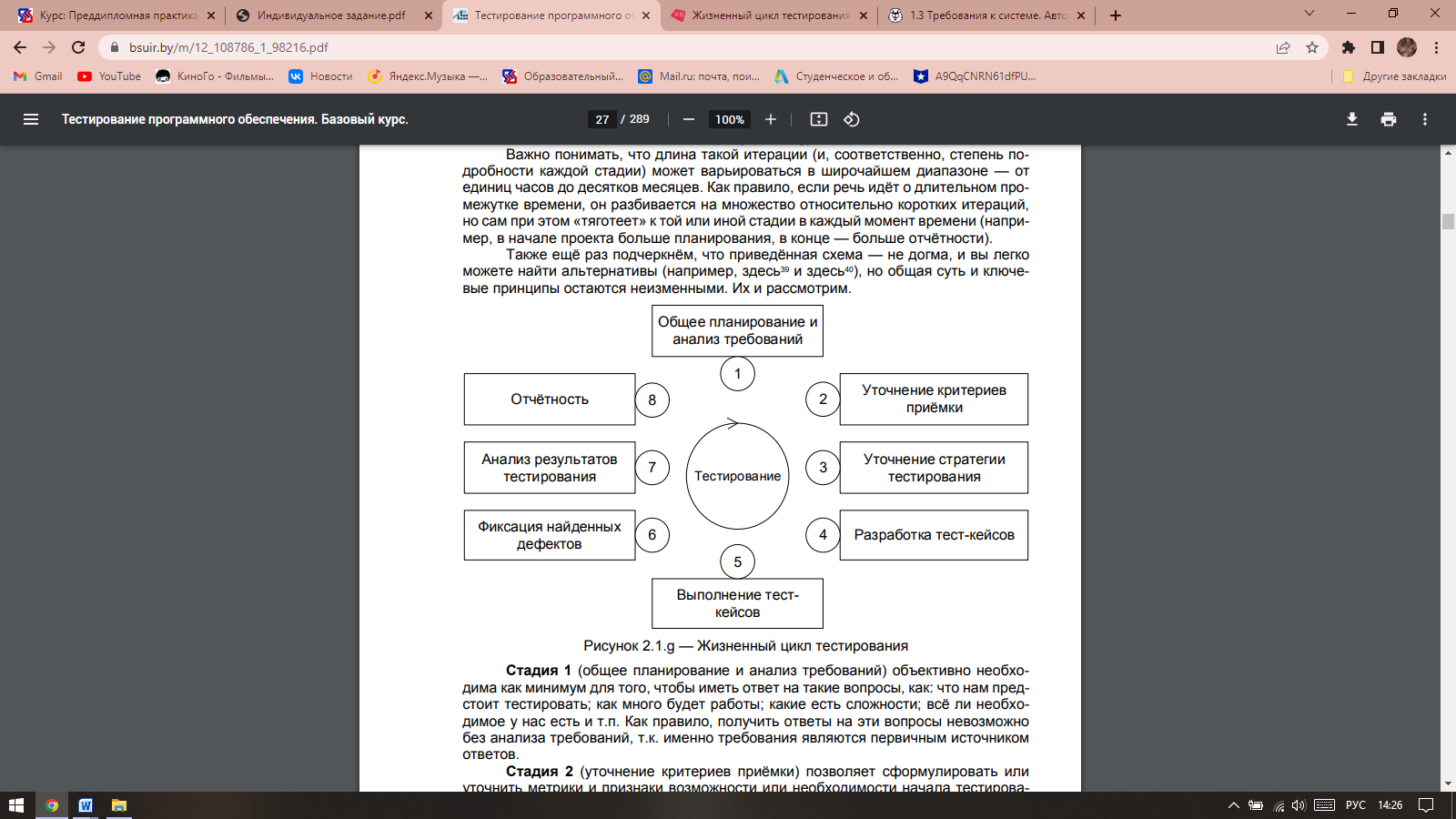


Рисунок 3.1 – Жизненный цикл тестирования

Общее планирование и анализ требований объективно необходима как минимум для того, чтобы иметь ответ на такие вопросы, как: что нам предстоит тестировать; как много будет работы; какие есть сложности; всё ли необходимое есть.

Уточнение критериев приёмки, позволяет сформулировать или уточнить метрики и признаки возможности или необходимости начала тестирования, приостановки и возобновления тестирования, завершения или прекращения тестирования.

Уточнение стратегии тестирования, представляет собой ещё одно обращение к планированию, но уже на локальном уровне: рассматриваются и уточняются те части стратегии тестирования, которые актуальны для текущей итерации.

Разработка тест-кейсов посвящена разработке, пересмотру, уточнению, доработке, переработке и прочим действиям с тест-кейсами, наборами тест-кейсов, тестовыми сценариями и иными артефактами, которые будут использоваться при непосредственном выполнении тестирования.

Выполнение тест-кейсов и фиксация найденных дефектов тесно связаны между собой и фактически выполняются параллельно: дефекты фиксируются сразу по факту их обнаружения в процессе выполнения тест-кейсов. Однако зачастую после выполнения всех тест-кейсов и написания всех отчётов о найденных дефектах проводится явно выделенная стадия уточнения, на которой все отчёты о дефектах рассматриваются повторно с целью формирования единого понимания проблемы.

Анализ результатов тестирования и отчётность также тесно связаны между собой и выполняются практически параллельно. Формулируемые на стадии анализа результатов выводы напрямую зависят от плана тестирования, критериев приёмки и уточнённой стратегии.

## 4 ТЕСТИРОВАНИЕ ДОКУМЕНТАЦИИ И ТРЕБОВАНИЙ

Требования – описание того, что должно быть реализовано.  
Требования описывают то, что необходимо реализовать, без детализации технической стороны решения.

Требования являются отправной точкой для определения того, что проектная команда будет проектировать, реализовывать и тестировать. Элементарная логика говорит нам, что если в требованиях что-то «не то», то и реализовано будет «не то», т.е. колоссальная работа множества людей будет выполнена впустую. Вне зависимости от того, какая модель разработки ПО используется на проекте, чем позже будет обнаружена проблема, тем сложнее и дороже будет её решение. А в самом начале («водопада», «спуска по букве v», «итерации», «витка спирали») идёт планирование и работа с требованиями.

Если проблема в требованиях будет выяснена на этой стадии, её решение может свестись к исправлению пары слов в тексте, в то время как недоработка, вызванная пропущенной проблемой в требованиях и обнаруженная на стадии эксплуатации, может даже полностью уничтожить проект.

Основные техники сбора и выявления требований:

– интервью. Самый универсальный путь выявления требований, заключающийся в общении проектного специалиста (как правило, специалиста по бизнес-анализу) и представителя заказчика (или эксперта, пользователя и т.д.). Интервью может проходить в классическом понимании этого слова (беседа в виде «вопрос-ответ»), в виде переписки и т.п. Главным здесь является то, что ключевыми фигурами выступают двое – интервьюируемый и интервьюер (хотя это и не исключает наличия «аудитории слушателей», например, в виде лиц, поставленных в копию переписки);

– работа с фокусными группами. Может выступать как вариант «расширенного интервью», где источником информации является не одно лицо, а группа лиц (как правило, представляющих собой целевую аудиторию, и/или обладающих важной для проекта информацией, и/или уполномоченных принимать важные для проекта решения);

– анкетирование. Этот вариант выявления требований вызывает много споров, т.к. при неверной реализации может привести к нулевому результату при объёмных затратах. В то же время при правильной организации анкетирование позволяет автоматически собрать и обработать огромное количество ответов от огромного количества респондентов. Ключевым фактором успеха является правильное составление анкеты, правильный выбор аудитории и правильное преподнесение анкеты;

– семинары и мозговой штурм. Семинары позволяют группе людей очень быстро обменяться информацией (и наглядно продемонстрировать те или иные идеи), а также хорошо сочетаются с интервью, анкетированием, прототипированием и моделированием – в том числе для обсуждения результатов и формирования выводов и решений. Мозговой штурм может проводиться и как часть семинара, и как отдельный вид деятельности. Он позволяет за минимальное время сгенерировать большое количество идей, которые в дальнейшем можно не спеша рассмотреть с точки зрения их использования для развития проекта;

– наблюдение. Может выражаться как в буквальном наблюдении за некими процессами, так и во включении проектного специалиста в эти процессы в качестве участника. С одной стороны, наблюдение позволяет увидеть то, о чём (по совершенно различным соображениям) могут умолчать интервьюируемые, анкетируемые и представители фокусных групп, но с другой – отнимает очень много времени и чаще всего позволяет увидеть лишь часть процессов;

– прототипирование. Состоит в демонстрации и обсуждении промежуточных версий продукта (например, дизайн страниц сайта может быть сначала представлен в виде картинок, и лишь затем свёрстан). Это один из лучших путей поиска единого понимания и уточнения требований, однако он может привести к серьёзным дополнительным затратам при отсутствии специальных инструментов (позволяющих быстро создавать прототипы) и слишком раннем применении (когда требования ещё не стабильны, и высока вероятность создания прототипа, имеющего мало общего с тем, что хотел заказчик);

– анализ документов. Хорошо работает тогда, когда эксперты в предметной области (временно) недоступны, а также в предметных областях, имеющих общепринятую устоявшуюся регламентирующую документацию;

– моделирование процессов и взаимодействий. Может применяться как к «бизнес-процессам и взаимодействиям» (например, «договор на закупку формируется отделом закупок, визируется бухгалтерией и юридическим отделом…»), так и к «техническим процессам и взаимодействиям» (например, «платёжное поручение генерируется модулем «Бухгалтерия», шифруется модулем «Безопасность» и передаётся на сохранение в модуль «Хранилище»»). Данная техника требует высокой квалификации специалиста по бизнес-анализу, т.к. сопряжена с обработкой большого объёма сложной (и часто плохо структурированной) информации. Самостоятельное описание. Является не столько техникой выявления требований, сколько техникой их фиксации и формализации. Очень сложно пытаться самому «придумать требования за заказчика», но в спокойной обстановке можно самостоятельно обработать собранную информацию и аккуратно оформить её для дальнейшего обсуждения и уточнения.

Атрибуты требований:

* корректность (точное описание разрабатываемого функционала);
* проверяемость (формулировка требований таким образом, чтобы можно было выставить однозначный вердикт, выполнено все в соответствии с требованиями или нет);
* полнота (в требовании должна содержаться вся необходимая для реализации функциональности информация);
* недвусмысленность (требование должно содержать однозначные формулировки);
* непротиворечивость (требование не должно содержать внутренних противоречий и противоречий другим требованиям и документам);
* приоритетность (у каждого требования должен быть приоритет (количественная оценка степени значимости требования), этот атрибут позволит грамотно управлять ресурсами на проекте);

– атомарность (требование нельзя разбить на отдельные части без потери деталей);

– модифицируемость (в каждое требование можно внести изменение);

– прослеживаемость (каждое требование должно иметь уникальный идентификатор);

– завершенность (требование должно содержать всю информацию, необходимую для разработчиков).

Тестирование документации – это начальная стадия процесса тестирования, которая выступает как система раннего оповещения об ошибках. Процесс тестирования так или иначе начинается с документации и требований. Тестирование документации предполагает начало тестирования еще до разработки продукта. Тестировщик может указать на логические ошибки в постановке задачи, несоответствия в требованиях, а также составить чек-лист, список проверок по предоставленному требованию.

В процесс тестирования документации важно вовлекать различных специалистов: тестировщики, проджект-менеджеры, бизнес-аналитики, разработчики.

Если ошибка в требованиях будет найдена на этапе тестирования требований, её решением может быть лишь исправление нескольких слов в тексте, в то время как найденный дефект в уже реализованном программном продукте, может привести до закрытия проекта.

В общем случае документацию можно разделить на два больших вида в зависимости от времени и места её использования.

Продуктная документация – используется проектной командой во время разработки и поддержки продукта. Она включает:

– план проекта и в том числе тестовый план;

– требования к программному продукту и функциональные спецификации;

– архитектуру и дизайн;

– тест-кейсы и наборы тест-кейсов;

– технические спецификации, такие как схемы баз данных, описания алгоритмов, интерфейсов и т.д.

Проектная документация включает в себя как продуктную документацию, так и некоторые дополнительные виды документации и используется не только на стадии разработки, но и на более ранних и поздних стадиях (например, на стадии внедрения и эксплуатации). Она включает:

– пользовательскую и сопроводительную документацию, такую как встроенная помощь, руководство по установке и использованию, лицензионные соглашения;

– маркетинговую документацию, которую представители разработчика или заказчика используют как на начальных этапах (для уточнения сути и концепции проекта), так и на финальных этапах развития проекта (для продвижения продукта на рынке).

Как известно, хорошая документация должна обладать следующими свойствами:

– требования должны быть полными, правильно и в полной мере описывать функцию, которую необходимо реализовать;

– однозначность (одинаковое восприятие требований всеми членами команды, никаких расхождений в трактовке быть не должно);

– непротиворечивость (не должно быть противоречивых требований, конфликтующих между собой).

## 5 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Тестирование программного обеспечения – это процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта.

Цель тестирования – проверка соответствия программного обеспечения предъявляемым требованиям, обеспечение уверенности в качестве, поиск очевидных ошибок в программном обеспечении, которые должны быть выявлены до того, как их обнаружат пользователи программы.

Тестирование проводится для:

– проверки соответствия требованиям;

– обнаружение проблем на более ранних этапах разработки и предотвращение повышения стоимости продукта;

– обнаружение вариантов использования, которые не были предусмотрены при разработке, а также взгляд на продукт со стороны пользователя;

– повышение лояльности к компании и продукту, т.к. любой обнаруженный дефект негативно влияет на доверие пользователей.

Тестирование охватывает ряд видов деятельности, такие как:

– постановка задачи для теста;

– проектирование;

– написание тестов;

– тестирование тестов;

– выполнение тестов;

– изучение результатов тестирования.

Основной задачей тестирования является обеспечение и контроль качества продукта.

Обеспечение качества и контроль качества – эти термины похожи на взаимозаменяемые, но между ними присутствует разница.

Обеспечение качества – более широкое понятие, которое может включать в себя контроль качества как составляющую, охватывающих все технологические этапы разработки. Суть обеспечения качества состоит в организации процесса разработки программного обеспечения, управление этим процессом и оценка этого процесса, с целью повышения качества всех его составляющих и конечного результата.

Контроль качества, в свою очередь, обладает более традиционным набором средств (от традиционных подходов к тестированию до самых современных технологий). Обычно, когда говорят «тестирование программного обеспечения», подразумевают именно контроль качества.

К задачам обеспечения качества относятся:

– проверка технических характеристик и требований;

– оценка рисков;

– планирование задач для улучшения качества продукции;

– подготовка документации, тестового окружения и данных;

– тестирование;

– анализ результатов тестирования, составление отчетов и других документов.

К задачам контроля качества относятся:

* проверка готовности продукта к релизу;
* проверка соответствия требований и качества данного проекта.

Верификация и валидация – два понятия тесно связаны с процессами тестирования и обеспечения качества. К сожалению, их часто путают, хотя отличия между ними достаточно существенны.

Верификация (verification) – это процесс оценки системы или её компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа. Таким образом выполняются ли наши цели, сроки, задачи по разработке проекта, определенные в начале текущей фазы.

Валидация (validation) – это определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, его требованиям к системе.

Жизненный цикл программного обеспечения – период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.

Жизненный цикл программного обеспечения делится на десять фаз:

– выработка требований;

– разработка спецификаций;

– общее проектирование;

– проектирование архитектуры;

– детальное проектирование;

– реализация и отладка;

– интеграция;

– сертификация;

– внедрение;

– cопровождение.

Модель разработки программного обеспечения – структура, систематизирующая различные виды проектной деятельности, их взаимодействие и последовательность в процессе разработки программного обеспечения.

Моделей разработки программного обеспечения много, но в общем случае классическими можно считать водопадную, v-образную, итерационную инкрементальную, спиральную и гибкую.

Тестирование на разных уровнях производится на протяжении всего жизненного цикла разработки и сопровождения программного обеспечения. Уровень тестирования определяет то, над чем производятся тесты: над отдельным модулем, группой модулей или системой, в целом. Проведение тестирования на всех уровнях системы – это залог успешной реализации и сдачи проекта.

По уровню детализации приложения выделяют три уровня тестирования:

– модульное;

– интеграционное;

– системное.

Модульным тестированием – называется проверка минимально функциональной автономной единицы программы. Чаще всего это проверка функций и процедур внутри кода программы. Такое тестирование чаще всего автоматизируют.

Целью модульного тестирования является проверка поведения каждой части программного обеспечения, независимо от других частей.

Преимущества модульного тестирования:

– модульное тестирование упрощает изменение и поддержку кода;

– модульное тестирование ускоряет разработку;

– тесты более надежны и в долгосрочной перспективе выполняются быстрее;

– менее затратное по времени и другим ресурсам;

– такой подход улучшает дизайн кода и позволяет проводить его рефакторинг;

– модульные тесты при интеграции также дают равенство сборки.

Недостатки модульного тестирования:

– модульное тестирование не выявляет всех ошибок в программе;

– не может оценить все пути выполнения, даже в тривиальных программах;

– в основном фокусируется на единицах измерения и не может выявлять ошибки интеграции на более широком уровне.

Интеграционное тестирование – это проверка взаимодействия между различными компонентами системы.

Целью данного вида тестирования является нахождение проблем взаимодействия модулей (компонент, подсистем). При наличии резерва времени на данное стадии тестирование ведется итерационно, с постепенным подключением последующих подсистем. Тестирование выполняется через интерфейс модулей с использованием метода «черного ящика».

Существует несколько стратегий проведения интеграционного тестирования, которые основываются на знаниях об архитектуре системы.

Монолитное тестирование предполагает, что отдельные компоненты системы серьезного тестирования не проходили. Система проверяется вся в целом после разработки всех модулей. Этот подход не следует путать с системным тестированием. Несмотря на то что при монолитном тестировании проверяется работа всей системы в целом, основная задача этого тестирования – определить проблемы взаимодействия отдельных модулей системы. Основные недостатки монолитного тестирования заключаются в сложности выявления источников ошибок.

Системное тестирование – это тестирование полностью собранной системы в том виде, в котором она будет поставляться заказчику. Тестирование проводится в условиях, приближенных к реальным. В рамках этого тестирование проводятся как функциональные, так и не функциональные тесты всей системы. Так же проверяется производительность и надежность системы. Основной задачей системного тестирования является проверка как функциональных, так и не функциональных требований к системе в целом.

Можно выделить два подхода к системному тестированию:

– на базе требований – для каждого требования пишутся тестовые случаи, проверяющие выполнение данного требования;

– на базе случаев использования – на основе представления о способах использования продукта создаются случаи использования системы.

На основании этой информации можно сделать вывод, что это обязательная процедура для обеспечения максимальной эффективности и безопасности работы продукта. Уровни тестирования позволяют определить то, над чем производятся исследования. Например, это могут быть отдельные модули, части из нескольких схожих модулей, система в целом. Правильно выполненные работы на всех уровнях системы являются залогом успешной реализации и сдачи подготовленного специалистами проекта.

Существует огромное количество видов тестирований. Хороший специалист сможет практически сходу определиться, с какими видами он будет работать над поставленной перед ним задачей. Что значительно облегчит работу, но если же не ориентироваться в этом достаточно хорошо, придется затратить большее количество времени и ресурсов.

## 6 ЧЕК-ЛИСТЫ, ТЕСТ-КЕЙСЫ, НАБОРЫ ТЕСТ-КЕЙСОВ

## 6.1 Чек-лист

Тестировщику приходится работать с огромным количеством информации, выбирать из множества вариантов решения задач и изобретать новые. В процессе этой деятельности объективно невозможно удержать в голове все мысли, а потому продумывание и разработку тест-кейсов рекомендуется выполнять с использованием так называемых «чек-листов».

Чек-лист (checklist) – набор идей.

Последнее слово не зря взято в скобки, т.к. в общем случае чек-лист – это просто набор идей: идей по тестированию, идей по разработке, идей по планированию и управлению – любых идей.

Понятие «чек-листа» не завязано на тестирование как таковое – это совершенно универсальная техника, которая может применяться в любой без исключения области жизни. В русском языке вне контекста информационных технологий чаще используется понятное и привычное слово «список» (например, «список покупок», «список дел» и т.д.), но в тестировании прижилась калькированная с английского версия – «чек-лист».

Чек-лист чаще всего представляет собой обычный и привычный нам список, который может быть:

– списком, в котором последовательность пунктов не имеет значения (например, список значений некоего поля);

– списком, в котором последовательность пунктов важна (например, шаги в краткой инструкции);

– структурированным (многоуровневым) списком (вне зависимости от учёта последовательности пунктов), что позволяет отразить иерархию идей.

Важно понять, что нет и не может быть никаких запретов и ограничений при разработке чек-листов – главное, чтобы они помогали в работе. Иногда чек-листы могут даже выражаться графически, хотя традиционно их составляют в виде многоуровневых списков.

Поскольку в разных проектах встречаются однотипные задачи, хорошо продуманные и аккуратно оформленные чек-листы могут использоваться повторно, чем достигается экономия сил и времени.

Для того чтобы чек-лист был действительно полезным инструментом, он должен обладать рядом важных свойств:

– логичность. Чек-лист пишется не «просто так», а на основе целей и для того, чтобы помочь в достижении этих целей. К сожалению, одной из самых частых и опасных ошибок при составлении чек-листа является превращение его в свалку мыслей, которые никак не связаны друг с другом;

– последовательность и структурированность. Со структурированностью всё достаточно просто – она достигается за счёт оформления чек-листа в виде многоуровневого списка. Что до последовательности, то даже в том случае, когда пункты чек-листа не описывают цепочку действий, человеку всё равно удобнее воспринимать информацию в виде неких небольших групп идей, переход между которыми является понятным и очевидным (например, сначала можно прописать идеи простых позитивных тест-кейсов, потом идеи простых негативных тест-кейсов, потом постепенно повышать сложность тест-кейсов, но не стоит писать эти идеи вперемешку);

– полнота и не избыточность. Чек-лист должен представлять собой аккуратную «сухую выжимку» идей, в которых нет дублирования (часто появляется из-за разных формулировок одной и той же идеи), и в то же время ничто важное не упущено.

## 6.2 Тест-кейс

Тест-кейс (test-case) – набор тестовых входных данных, условий выполнения и ожидаемых результатов, разработанных с конкретной целью, такой как проверка некоторого пути выполнения программы или проверка соответствия некоторому требованию.

Под тест-кейсом также может пониматься соответствующий документ, представляющий формальную запись тест-кейса.

Высокоуровневый тест-кейс (high level test case) – тест-кейс без конкретных входных данных и ожидаемых результатов.

Как правило, ограничивается общими идеями и операциями, схож по своей сути с подробно описанным пунктом чек-листа. Достаточно часто встречается в интеграционном тестировании и системном тестировании, а также на уровне дымового тестирования. Может служить отправной точкой для проведения исследовательского тестирования или для создания низкоуровневых тест-кейсов.

Низкоуровневый тест-кейс (low level test case) – тест-кейс с конкретными входными данными и ожидаемыми результатами.

Спецификация тест-кейса (test case specification) – документ, описывающий набор тест-кейсов (включая их цели, входные данные, условия и шаги выполнения, ожидаемые результаты) для тестируемого элемента (test item, test object).

Спецификация теста (test specification) – документ, состоящий из спецификации тест-дизайна (test design specification), спецификации тест-кейса (test case specification) и/или спецификации тест-процедуры (test procedure specification).

Тест-сценарий (test scenario, test procedure specification, test script) – документ, описывающий последовательность действий по выполнению теста (также известен как «тест-скрипт»).

Жизненный цикл тест-кейса:

– создан;

– запланирован;

– не выполнен;

– выполняется;

– пропущен;

– провален;

– пройден успешно;

– заблокирован;

– закрыт;

– требует доработки.

Как уже было сказано выше, термин «тест-кейс» может относиться к формальной записи тест-кейса в виде технического документа. Эта запись имеет общепринятую структуру, компоненты которой называются атрибутами (полями) тест-кейса.

В зависимости от инструмента управления тест-кейсами внешний вид их записи может немного отличаться, могут быть добавлены или убраны отдельные поля, но концепция остаётся неизменной.

Атрибуты тест-кейса:

– идентификатор. Представляет собой уникальное значение, позволяющее однозначно отличить один тест-кейс от другого и используемое во всевозможных ссылках;

– приоритет. Показывает важность тест-кейса. Он может быть выражен буквами (A, B, C, D, E), цифрами (1, 2, 3, 4, 5), словами («крайне высокий», «высокий», «средний», «низкий», «крайне низкий») или иным удобным способом;

– связанное с тест-кейсом требование. Показывает то основное требование, проверке выполнения которого посвящён тест-кейс (основное – потому, что один тест-кейс может затрагивать несколько требований);

– модуль и подмодуль приложения. Указывают на части приложения, к которым относится тест-кейс, и позволяют лучше понять его цель;

– заглавие (суть) тест-кейса. Призвано упростить и ускорить понимание основной идеи (цели) тест-кейса без обращения к его остальным атрибутам;

– исходные данные, необходимые для выполнения тест-кейса. Позволяют описать всё то, что должно быть подготовлено до начала выполнения тест-кейса;

– шаги тест-кейса описывают последовательность действий, которые необходимо реализовать в процессе выполнения тест-кейса;

– ожидаемые результаты. По каждому шагу тест-кейса описывают реакцию приложения на действия, описанные в поле «шаги тест-кейса». Номер шага соответствует номеру результата.

## 6.3 Набор тест-кейсов

Набор тест-кейсов – совокупность тест-кейсов, выбранных с некоторой общей целью или по некоторому общему признаку. Иногда в такой совокупности результаты завершения одного тест-кейса становятся входным состоянием приложения для следующего тест-кейса.

Крайне неудобно (более того, это ошибка) каждый раз писать в каждом тест-кейсе одни и те же приготовления и повторять одни и те же начальные шаги. Намного удобнее объединить несколько тест-кейсов в набор или последовательность.

В общем случае наборы тест-кейсов можно разделить на свободные (порядок выполнения тест-кейсов не важен) и последовательные (порядок выполнения тест-кейсов важен).

Преимущества свободных наборов:

– тест-кейсы можно выполнять в любом удобном порядке, а также создавать «наборы внутри наборов»;

– если какой-то тест-кейс завершился ошибкой, это не повлияет на возможность выполнения других тест-кейсов.

Преимущества последовательных наборов:

– каждый следующий в наборе тест-кейс в качестве входного состояния приложения получает результат работы предыдущего тест-кейса, что позволяет сильно сократить количество шагов в отдельных тест-кейсах;

– длинные последовательности действий куда лучше имитируют работу реальных пользователей, чем отдельные «точечные» воздействия на приложение.

Единственная задача наборов – повысить эффективность тестирования за счёт ускорения и упрощения выполнения тест-кейсов, увеличения глубины исследования некоей области приложения или функциональности, следования типичным пользовательским сценариям или удобной последовательности выполнения тест-кейсов и т.д.

Набор тест-кейсов всегда создаётся с какой-то целью, на основе какой-то логики, и по этим же принципам в набор включаются тесты, обладающие подходящими свойствами. Если же говорить о наиболее типичных подходах к составлению наборов тест-кейсов, то можно обозначить следующее:

– на основе чек-листов. Каждый пункт чек-листа может превратиться в несколько тест-кейсов – и вот мы получаем готовый набор);

– на основе разбиения приложения на модули и подмодули. Для каждого модуля (или его отдельных подмодулей) можно составить свой набор тест-кейсов;

– по принципу проверки самых важных, менее важных и всех остальных функций приложения;

– по принципу группировки тест-кейсов для проверки некоего уровня требований или типа требований, группы требований или отдельного требования;

– по принципу частоты обнаружения тест-кейсами дефектов в приложении;

– по архитектурному принципу. Наборы для проверки пользовательского интерфейса и всего уровня представления, для проверки уровня бизнес-логики, для проверки уровня данных;

– по видам тестирования;

– по области внутренней работы приложения.

Без хороших инструментальных средств управления тест-кейсами работать с наборами тест-кейсов крайне тяжело, т.к. приходится самостоятельно следить за приготовлениями, «недостающими шагами», изолированностью или обобщённостью, свободностью или последовательностью.

## 7 **ОТЧЕТЫ О ДЕФЕКТАХ**

Дефектом упрощённо можно считать любое расхождение ожидаемого (свойства, результата, поведения, которое мы ожидали увидеть) и фактического (свойства, результата, поведения, которое мы на самом деле увидели). При обнаружении дефекта создаётся отчёт о дефекте.

Дефект – расхождение ожидаемого и фактического результата.

Ошибка – действие человека, приводящее к некорректным результатам.

Ожидаемый результат – поведение системы, описанное в требованиях.

Фактический результат – поведение системы, наблюдаемое в процессе тестирования.

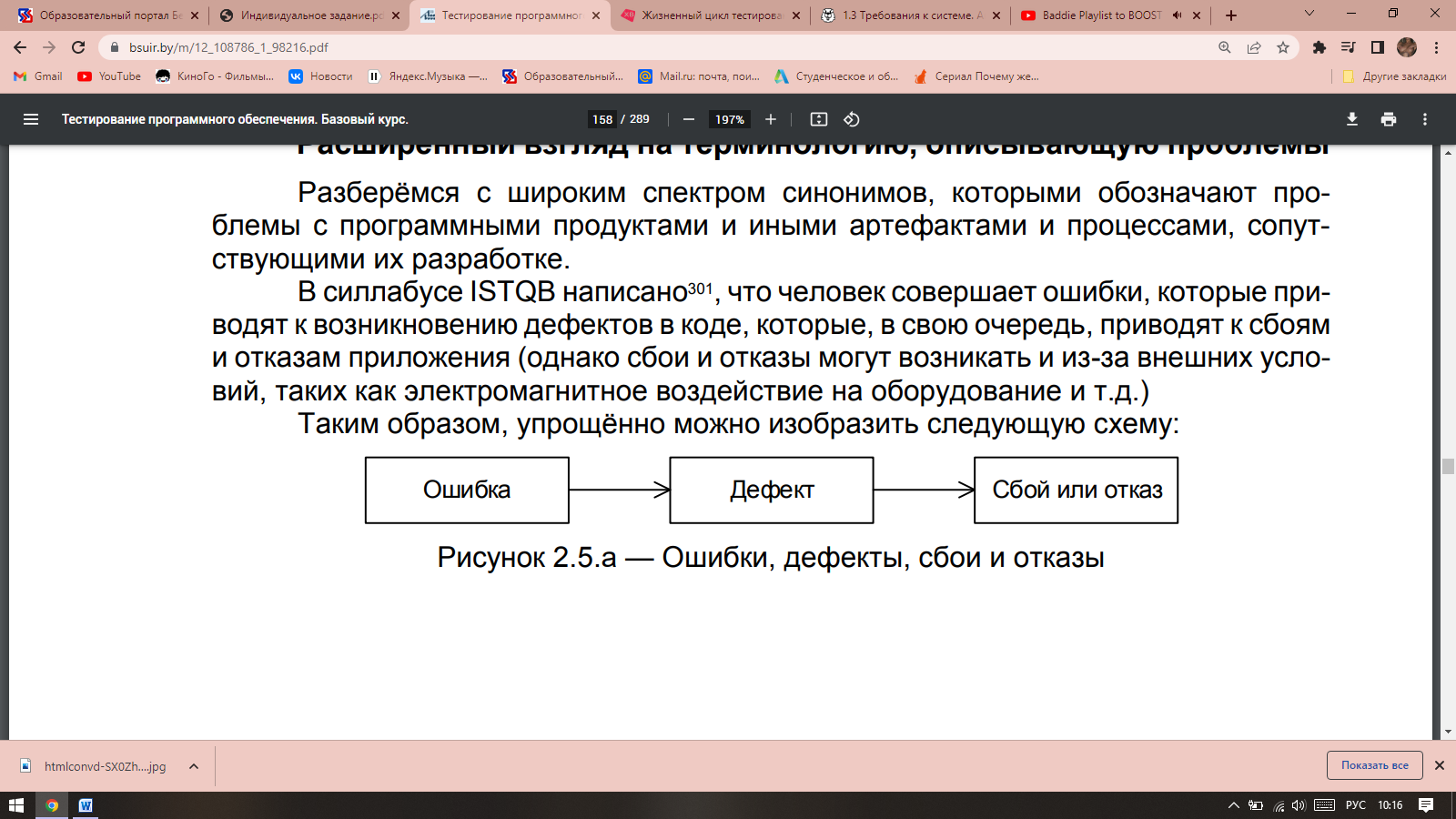


Рисунок 7.1 – Ошибки, дефекты, сбои и отказы

Отчёт о дефекте – документ, описывающий и приоритизирующий обнаруженный дефект, а также содействующий его устранению.

Как следует из самого определения, отчёт о дефекте пишется со следующими основными целями:

– предоставить информацию о проблеме. Уведомить проектную команду и иных заинтересованных лиц о наличии проблемы, описать суть проблемы;

– приоритизировать проблему. Определить степень опасности проблемы для проекта и желаемые сроки её устранения;

– содействовать устранению проблемы. Качественный отчёт о дефекте не только предоставляет все необходимые подробности для понимания сути случившегося, но также может содержать анализ причин возникновения проблемы и рекомендации по исправлению ситуации.

Отчёт о дефекте (и сам дефект вместе с ним) проходит определённые стадии жизненного цикла, которые схематично можно показать так (рис. 7.2).

Набор стадий жизненного цикла, их наименование и принцип перехода от стадии к стадии может различаться в разных инструментальных средствах управления жизненным циклом отчётов о дефектах. Более того – многие такие средства позволяют гибко настраивать эти параметры. На рис. 7.2 показан лишь общий принцип.

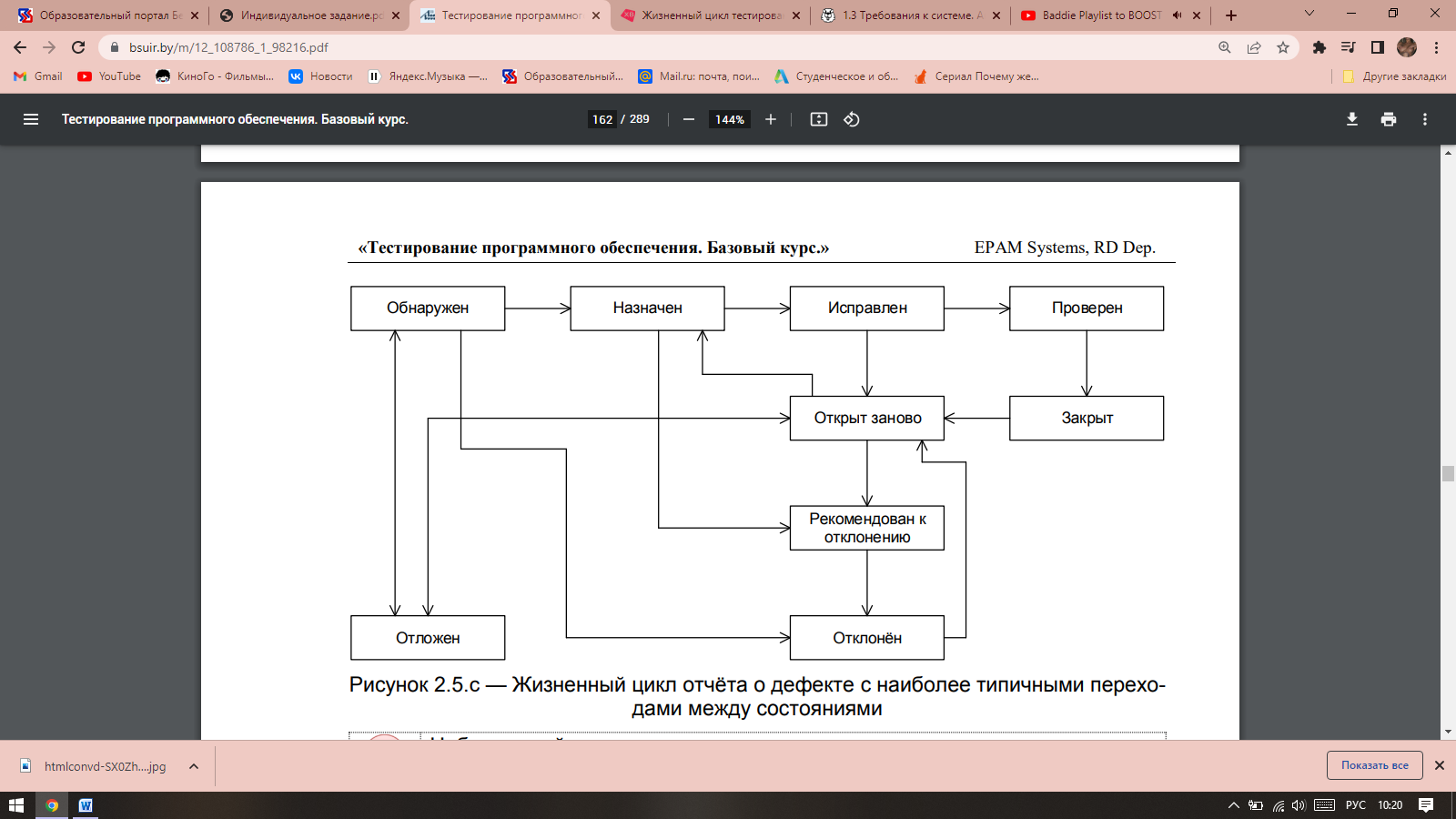


Рисунок 7.2 – Жизненный цикл отчёта о дефекте с наиболее типичными переходами между состояниями

Жизненный цикл отчёта о дефекте:

– обнаружен (submitted) – начальное состояние отчёта (иногда называется «Новый» (new)), в котором он находится сразу после создания. Некоторые средства также позволяют сначала создавать черновик (draft) и лишь потом публиковать отчёт;

– назначен (assigned) – в это состояние отчёт переходит с момента, когда кто-то из проектной команды назначается ответственным за исправление дефекта. Назначение ответственного производится или решением лидера команды разработки, или коллегиально, или по добровольному принципу, или иным принятым в команде способом или выполняется автоматически на основе определённых правил;

– исправлен (fixed) – в это состояние отчёт переводит ответственный за исправление дефекта член команды после выполнения соответствующих действий по исправлению;

– проверен (verified) – в это состояние отчёт переводит тестировщик, удостоверившийся, что дефект на самом деле был устранён. Как правило, такую проверку выполняет тестировщик, изначально написавший отчёт о дефекте. По поводу того, должен ли проверять факт устранения дефекта именно тот тестировщик, который его обнаружил, или обязательно другой, есть много «священных войн». Сторонники второго варианта утверждают, что свежий взгляд человека, ранее не знакомого с данным дефектом, позволяет ему в процессе верификации с большой вероятностью обнаружить новые дефекты;

– закрыт (closed) – состояние отчёта, означающее, что по данному дефекту не планируется никаких дальнейших действий. Здесь есть некоторые расхождения в жизненном цикле, принятом в разных инструментальных средствах управления отчётами о дефектах;

– открыт заново (reopened) – в это состояние (как правило, из состояния «Исправлен») отчёт переводит тестировщик, удостоверившийся, что дефект по-прежнему воспроизводится на билде, в котором он уже должен быть исправлен;

– рекомендован к отклонению (to be declined) – в это состояние отчёт о дефекте может быть переведён из множества других состояний с целью вынести на рассмотрение вопрос об отклонении отчёта по той или иной причине. Если рекомендация является обоснованной, отчёт переводится в состояние «Отклонён» (см. следующий пункт);

– отклонён (declined) – в это состояние отчёт переводится в случаях, подробно описанных в пункте «Закрыт», если средство управления отчётами о дефектах предполагает использование этого состояния вместо состояния «Закрыт» для тех или иных резолюций по отчёту;

– отложен (deferred) – в это состояние отчёт переводится в случае, если исправление дефекта в ближайшее время является нерациональным или не представляется возможным, однако есть основания полагать, что в обозримом будущем ситуация исправится (выйдет новая версия библиотеки, вернётся из отпуска специалист по некоей технологии, изменятся требования заказчика и т.д.).

В зависимости от инструментального средства управления отчётами о дефектах внешний вид их записи может немного отличаться, могут быть добавлены или убраны отдельные поля, но концепция остаётся неизменной.

Атрибуты отчёта о дефекте:

– идентификатор (identifier) представляет собой уникальное значение, позволяющее однозначно отличить один отчёт о дефекте от другого и используемое во всевозможных ссылках;

– краткое описание (summary) должно в предельно лаконичной форме давать исчерпывающий ответ на вопросы «Что произошло?» «Где это произошло»? «При каких условиях это произошло?»;

– подробное описание (description) представляет в развёрнутом виде необходимую информацию о дефекте, а также (обязательно!) описание фактического результата, ожидаемого результата и ссылку на требование (если это возможно);

– шаги по воспроизведению (steps to reproduce, STR) описывают действия, которые необходимо выполнить для воспроизведения дефекта. Это поле похоже на шаги тесткейса, за исключением одного важного отличия: здесь действия прописываются максимально подробно, с указанием конкретных вводимых значений и самых мелких деталей, т.к. отсутствие этой информации в сложных случаях может привести к невозможности воспроизведения дефекта;

– воспроизводимость (reproducibility) показывает, при каждом ли прохождении по шагам воспроизведения дефекта удаётся вызвать его проявление. Это поле принимает всего два значения: всегда (always) или иногда (sometimes);

– важность (severity) показывает степень ущерба, который наносится проекту существованием дефекта. В общем случае выделяют следующие градации важности:

1. критическая (critical) – существование дефекта приводит к масштабным последствиям катастрофического характера, например, потеря данных, раскрытие конфиденциальной информации, нарушение ключевой функциональности приложения;
2. высокая (major) – существование дефекта приносит ощутимые неудобства многим пользователям в рамках их типичной деятельности, например, недоступность вставки из буфера обмена, неработоспособность общепринятых клавиатурных комбинаций, необходимость перезапуска приложения при выполнении типичных сценариев работы;
3. вредняя (medium) – существование дефекта слабо влияет на типичные сценарии работы пользователей, и/или существует обходной путь достижения цели, например, диалоговое окно не закрывается автоматически после нажатия кнопок «OK»/«Cancel», при распечатке нескольких документов подряд не сохраняется значение поля «Двусторонняя печать», перепутаны направления сортировок по некоему полю таблицы;
4. низкая (minor) – существование дефекта редко обнаруживается незначительным процентом пользователей и (почти) не влияет на их работу, например, опечатка в глубоко вложенном пункте меню настроек, некое окно сразу при отображении расположено неудобно (нужно перетянуть его в удобное место), неточно отображается время до завершения операции копирования файлов;

– срочность (priority) показывает, как быстро дефект должен быть устранён. В общем случае выделяют следующие градации срочности:

1. наивысшая (ASAP, as soon as possible) срочность указывает на необходимость устранить дефект настолько быстро, насколько это возможно. В зависимости от контекста «настолько быстро, насколько возможно» может варьироваться от «в ближайшем билде» до единиц минут;
2. высокая (high) срочность означает, что дефект следует исправить вне очереди, т.к. его существование или уже объективно мешает работе, или начнёт создавать такие помехи в самом ближайшем будущем;
3. обычная (normal) срочность означает, что дефект следует исправить в порядке общей очерёдности. Такое значение срочности получает большинство дефектов;
4. низкая (low) срочность означает, что в обозримом будущем исправление данного дефекта не окажет существенного влияния на повышение качества продукта;

– симптом (symptom) – позволяет классифицировать дефекты по их типичному проявлению. Не существует никакого общепринятого списка симптомов. Более того, далеко не в каждом инструментальном средстве управления отчётами о дефектах есть такое поле, а там, где оно есть, его можно настроить;

– возможность обойти (workaround) – показывает, существует ли альтернативная последовательность действий, выполнение которой позволило бы пользователю достичь поставленной цели (например, клавиатурная комбинация Ctrl+P не работает, но распечатать документ можно, выбрав соответствующие пункты в меню). В некоторых инструментальных средствах управления отчётами о дефектах это поле может просто принимать значения «Да» и «Нет», в некоторых при выборе «Да» появляется возможность описать обходной путь. Традиционно считается, что дефектам без возможности обхода стоит повысить срочность исправления;

– комментарий (comments, additional info) – может содержать любые полезные для понимания и исправления дефекта данные. Иными словами, сюда можно писать всё то, что нельзя писать в остальные поля;

– приложения (attachments) – представляет собой не столько поле, сколько список прикреплённых к отчёту о дефекте приложений (копий экрана, вызывающих сбой файлов и т.д.).

## 8 АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ

Автоматизированное тестирование – это процессы, которые запускают программы и скрипты для тестирования отдельных модулей, используя повторяющиеся действия. Фактически, это значит, что программа запускает определенные скрипты, чтобы проверить все составляющие проекта и оценить его. Для того, чтобы создать программу тестирования требуются определенные ресурсы.

Автоматизированное тестирование использует программные средства для выполнения тестов и проверки корректности результатов выполнения, что упрощает тестирование и сокращает его длительность. Главное преимущество автоматизированного тестирования состоит в возможности повторного прогона тестов без участия человека.

Традиционный и наиболее популярный среди разработчиков способ состоит в организации автоматизации тестирования на уровне кода. Данный подход будет подробно рассмотрен при описании модульного тестирования. Автоматизированное тестирование на уровне кода часто критикуют за невозможность тестирования пользовательского интерфейса программы.

Второй способ автоматизации тестирования состоит в имитации действий пользователя с использованием специальных инструментальных средств (GUI-тестирование). Данный вид тестирования относится к тестированию методом «черного ящика».

Существуют четыре поколения инструментов и техник, предназначенных для организации GUI-тестирования:

* утилиты записи и воспроизведения (capture/playback tools) записывают действия во время ручного тестирования. В дальнейшем они позволяют воспроизвести ранее записанные действия без участия человека, значительно увеличивая продуктивность и устраняя повторение однообразных действий. Основным недостатком инструментальных средств данного поколения является то, что любое изменение расположения визуальных элементов программы приводит к необходимости повторной записи ручных тестов;
* сценарий (scripting) – форма автоматизации тестирования с использованием специализированных скриптовых языков. Язык должен поддерживать эмуляцию действий пользователя и получение результатов действий. Разработкой тестов занимаются программисты, работающие отдельно от тестировщиков, непосредственно запускающие те6сты. Изменения в тестируемом продукте требуют внесения исправлений и в соответствующих скриптах;
* data-driven testing – методология автоматизации тестирования, основанная на использовании в скриптах параметров выполнения тестов. Параметры, задающие логику работы тестов (например, входные значения и ожидаемые результаты), находятся в некотором внешнем хранилище. Подобный подход позволяет организовать выполнение сценариев с различными наборами входных параметров и повысить гибкость тестирования.

В автоматизированном тестировании должен присутствовать тестировщик, который создаст программу и затем будет ее запускать. Программное обеспечение для автоматизации тестирования также может вводить тестовые данные в тестовую среду, сравнивать ожидаемые и фактические результаты и создавать подробные отчеты о тестах. Как правило, автоматизация тестирования требует значительных вложений денег и ресурсов.

Последовательные циклы разработки, особенно в крупных компаниях потребуют многократного выполнения одного и того же набора тестов. Используя инструмент автоматизации тестирования, можно записать этот набор тестов и при необходимости воспроизвести его. После автоматизации набора тестов вмешательство человека не требуется. Цель автоматизации – уменьшить количество тестовых примеров, которые нужно запускать вручную, а не полностью исключить ручное тестирование.

Достоинства автоматизированного тестирования:

– нагрузка на приложение. Когда используется автоматизированное тестирование, становится возможным моделирование большой нагрузки, которая приближена к реальной ситуации;

– временной фактор. Ручное тестирование – это долгий и ресурсоемкий процесс, в то время как код для сценария пишется один раз;

– повторяемость. Код автоматизированных тестов может быть использован неоднократно, особенно при внедрении новой функциональности.

Недостатки автоматизированного тестирования:

– отсутствие обратной связи. Автоматизированное тестирование не способно предоставить обратную связь относительно качества продукта – оно лишь выполняет запрограммированные сценарии;

– отсутствие тестирования глазами пользователя;

– отсутствие возможности тестирования цвета, дизайна и эргономики;

– надежность. Автоматизированные тесты могут упасть по многим причинам, например, при большой загруженности тестовой машины или при проблемах с сетью;

– стоимость. Для небольших проектов инструменты автоматизированного тестирования могут оказаться достаточно затратными, поэтому более рационально их использовать для долгосрочных проектов.

## 9 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

В индивидуальном задании от организации было дано тестирование мобильного приложения «Белтелеком», тестирование было организовано из нескольких этапов:

– ручное тестирование методом черного ящика:

1. расширенное;
2. удобства использования;
3. пользовательского интерфейса.

Главные выводы: итогом проведения тестирования мобильного приложения, стал оформленный конечный результат процесса тестирования с описанными дефектами, использованной документацией, представленные в приложениях А, Б, В, Г. Вся тестовая документация была занесена в github репозиторий (приложение Д).

## ЗАКЛЮЧЕНИЕ

В результате прохождения технологической практики на РУП «Белтелеком» была изучена структура предприятия и её производственная структура, инфокоммуникационная структура и технологическое программное обеспечение. Было получено и выполнено индивидуальное задание.

Были использованы все знания и умения, приобретенные в ходе обучения по специальности на практике. Более того, для решения некоторых задач, возникающих в ходе проведения тестирования, были отработаны умения по поиску новой информации и её анализе. Доступные и самые распространенные методы тестирования были использованы и получен соответствующий опыт по работе с ними.

В процессе выполнения индивидуального задания были создана тестовая документация:

– тест-кейсы;

– тест-план.

Были получены следующие практические навыки:

– составление тестовой документации для тестирования мобильного приложения;

– проведение ручного тестирования новых функциональностей продукта;

– воспроизведение и заведение дефектов, обнаруженных на этапе эксплуатации.

Результатом прохождения практики стало успешное тестирование разработанного мобильного приложения для пользователей.

В завершении хочется отметить, что практика прошла без трудностей благодаря постоянной коммуникации с руководителями и все поставленные задачи выполнены успешно.

## ПРИЛОЖЕНИЕ А

## 

## Тест-план для мобильного приложения «Белтелеком»

1. Цель: закрепить навыки работы со спецификацией, разработать тест кейсы для тестирования приложения.
2. Области подвергаемые тестированию:

* оказание услуг;
* формирование счета;
* пополнение счета;
* осуществление расчётов.

1. Тестовая стратегия и подходы:

Будут применяться следующие три вида тестирования: удобства использования, пользовательского интерфейса и совместимости.

Планируется два этапа проведения процесса тестирования. Первый этап заключается в составлении тест-плана. Второй этап будет посвящен прогону тестов с выявлением и описанием дефектов. Таким образом, достигается максимальная детализация глубины тестирования, что, в свою очередь, позволяет более точно определить затрачиваемые ресурсы.

1. Критерии начала и окончания тестирования:

Тестирование может быть начато, если выполнены следующие условия:

* готова и утверждена необходимая документация;
* тестируемый функционал окончен и готов для передачи в тестирование.

Тестирование окончено, если выполнены следующие условия:

* все найденные дефекты задокументированы.

1. Ресурсы:

* аппаратные ресурсы**:**

1. устройство – IPhone 11;
2. разрешение экрана – 1920x1080;

* временные ресурсы – 8 недель с 15.05.2023 по 04.06.2023;
* программные ресурсы:

1. операционные системы – IOS 16;

* человеческие ресурсы:

1. тестировщик ПО – Ясюкевич Полина Сергеевна;

* финансовые ресурсы – не требовались в проведении тестирования.

**Продолжение приложения А**

1. Расписание:

Таблица А.1 – Расписание сроков тестирования

|  |  |  |
| --- | --- | --- |
| Время | Задачи | Решение/Описание |
| 8 часов | Изучение спецификации и внесение изменений с командой разработки | Изучить спецификацию и требуемое задание, обсудить их с командой разработки и начать тестирование, наладить процесс |
| 20 часов | Составление тест-плана и тест-кейсов, чек-листа | Встреча с командой, определение времени и средств, при которых возможна реализация продукта, написание документации |
| 12 часов | Протестировать совместимость и пользовательского интерфейса | Тестирование совместимости и пользовательского интерфейса мобильного приложения |
| 3 часа | Составить отчёт по выполненной работе | Описание выполненной работы |
| 5 часа | Протестировать удобство использования | Тестирование удобства использования приложением |
| 3 часа | Составить отчёт по выполненной работе | Описание выполненной работы |
| 4 часа | Составить финальный отчёт | Описание проделанной работы за все время тестирования |
| 1 час | Провести проверку отчёта | Проверка правильности выполнения отчета |
| 2 часа | Провести совещание | Проводится совещание на предприятии для финального обсуждения об выполнении отчета |

**Продолжение приложения А**

1. Роли и ответственность:

Таблица А.2 – Роли и ответственность в тестировании

|  |  |  |
| --- | --- | --- |
| **ФИО** | **Роль** | **Ответственность** |
| Ясюкевич Полина | Тестировщик ПО | Составление документации и проведение тестирования. |

1. Риски и ограничения:

− ввиду ограниченности ресурсов, невозможно гарантировать полный просмотр базы данных;

− человеческий фактор.

1. Документация:

После окончания тестирования планируется наличие таких документов:

− тест-план;

− тест-кейсы.

Главные выводы: итогом проведения тестирования стал оформленный конечный результат процесса тестирования с описанными дефектами, использованной документацией и распределение времени.

## ПРИЛОЖЕНИЕ Б

## Тест-кейсы для тестирования пользовательского интерфейса

Таблица Б.1 – Тест-кейсы для тестирования пользовательского интерфейса

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Ссылка на требование** | **Шаги** | **Ожидаемый результат** | **Фактический результат** | **Статус** | **Комментарии** |
| 1 |  | В поле ввода ввести неверные данные | Текстовые поля не должны превышать максимальное количество символов и не разрешать заполнение только специальными символами | При вводе огромного количества символов в поле, оно выдает ошибку ввода (текстовое сообщение) | Выполнено успешно |  |
| 2 |  | Найти кнопку для перехода и произвести переход | Все кнопки навигации на странице работают и перенаправляют пользователей на нужную страницу или экран | Всё перенаправление на другие страницы работает корректно | Выполнено успешно |  |
| 3 |  | Ввести данные и нажать на кнопку подтверждения | Пользовательский интерфейс имеет работающую кнопку подтверждения каждый раз, когда пользователь хочет сохранить или удалить элемент | При вводе информации есть кнопка подтверждения | Выполнено успешно |  |
| 4 |  | Просмотреть все страницы мобильного приложения и оценить общий внешний вид страниц | Общий вид страницы: гармонично и приятно выглядит программный продукт, при масштабировании ничего не нарушается. Пользователи могут открывать страницы с разных устройств с разными разрешениями, на любом из них текст был читаемым, все элементы отображаются правильно, не наезжают друг на друга и не исчезают | Программный продукт выглядит гармонично и на различных устройствах элементы отображаются корректно | Выполнено успешно |  |
| **Продолжение приложения Б**  **Продолжение приложения Б** | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 5 |  | Просмотреть все страницы мобильного приложения и просмотреть текстовое наполнение | Текст: он не содержит ошибок, опечаток и других возможных погрешностей, так же должен быть выровнен по одинаковым параметрам, чтобы общая картина смотрелась целостно и гармонично | В программном продукте текст не содержит ошибок и корректно отображается в разделах | Выполнено успешно |  |
| 6 |  | Просмотреть страницу с формой, выполнить действие с формой | Формы: при работе с формами есть множество элементов, на которые стоит обратить внимание. Во-первых, это само окно формы, его расположение, внешний вид и так далее. Во-вторых, элементы на форме: радиокнопки, чек-боксы, выпадающие списки, текстовые поля, и так далее. Все элементы должны соответствовать требованиям и спецификациям. | Приложение содержит формы, которые работают корректно | Выполнено успешно |  |
| 7 |  | Просмотреть все страницы мобильного приложения и оценить единый внешний вид | Единый внешний вид: для целостности интерфейса весь дизайн унифицирован. Цвета, шрифты, поля, иконки, изображения и другие элементы на страницах программного продукта являются стандартными для него. | Внешний вид страниц мобильного приложения имеет целостную картину (весь дизайн в едином стиле) | Выполнено успешно |  |

## ПРИЛОЖЕНИЕ В

## Тест-кейсы для тестирования удобства использования

Таблица В.1 – Тест-кейсы для тестирования удобства использования

|  |  |  |
| --- | --- | --- |
| **Описание теста** | **Результат** | **Рекомендации** |
| Цветовая гамма страницы | No | Рекомендуется добавить более ярких цветов на страницу |
| Размер шрифта | Ok |  |
| Цвет текста | Ok |  |
| Расположение текста | Ok |  |
| Адаптация мобильного приложения к изменению размера окна | Ok |  |
| Наполнение контентом | Ok |  |
| Скорость загрузки | Ok |  |
| Работа ссылок | Ok |  |

## ПРИЛОЖЕНИЕ Г

## Тест-кейсы для расширенного тестирования

Таблица В.1 – Тест-кейсы для расширенного тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Тестируемый модуль** | **Шаги** | **Ожидаемый результат** | **Фактический результат** | **Статус** |
| ST\_001 | Страница «Аккаунт» | Войти в приложение, Зайти на страницу «Аккаунт» | На странице аккаунта есть окно в котором содержится информация о счёте, тарифе и днях до оплаты | На странице есть окно с информацией, которая есть в требованиях | Выполнено успешно |
| ST\_002 | Страница «Аккаунт» | Войти в приложение, Зайти на страницу «Аккаунт», Нажать на новость | На странице аккаунта появляется всплывающая реклама/акция, при нажатии на которую должен происходить переход на страницу «Новости» | При нажатии на рекламу/акцию происходит переход | Выполнено успешно |
| ST\_003 | Страница «Аккаунт» | Войти в приложение, Зайти на страницу «Аккаунт» | В правом верхнем углу находится голубой крест, при нажатии на который должен происходить переход на страницу с добавлением услуги | При нажатии на голубой крест происходит переход на новую страницу с добавлением услуги | Выполнено успешно |
| ST\_004 | Страница с добавлением услуги | Войти в приложение, Зайти на страницу «Аккаунт», Нажать на голубой крест в правом верхнем углу, Заполнить поле ввода логина | Ввод логина в поле формы | Логин вводится в поле | Выполнено успешно |
| **Продолжение приложения Г** | | | | | |
| ST\_005 | Страница с добавлением услуги | Войти в приложение, Зайти на страницу «Аккаунт», Нажать на голубой крест в правом верхнем углу, Заполнить поле ввода пароля | Ввод пароля в поле формы | Пароль не вводится в поле |  |
| ST\_006 | Страница с добавлением услуги | Войти в приложение, Зайти на страницу «Аккаунт», Нажать на голубой крест в правом верхнем углу, Заполнить поле ввода пароля, Нажать на показ пароля | При вводе пароля можно просмотреть его путем нажатия на символ глаза, который должен показываться постоянно. При нажатии на этот символ пароль просматривается | Так как пароль не вводится и символ глаза не присутствует, то невозможно его просмотреть с данным символом. При обстоятельстве, когда символ показывается, но пароль не вводится, после нажатия на символ глаза в строке ввода пароля появляется одна точка, которая не просматривается, а символ глаза пропадает |  |
| ST\_007 | Страница с добавлением услуги | Войти в приложение, Зайти на страницу «Аккаунт», Нажать на голубой крест в правом верхнем углу, Нажать на «Добавить услуги по номеру телефона», Ввести номер домашнего телефона | Стоит ограничение (7 символов) на ввод цифр в поле для ввода домашнего номера | При вводе максимального количества цифр, дальнейший ввод невозможен | Выполнено успешно |
| ST\_008 | Страница с добавлением услуги | Войти в приложение, Зайти на страницу «Аккаунт», Нажать на голубой крест в правом верхнем углу, Нажать на «Добавить услуги по номеру телефона», Ввести фамилию | Ввод фамилии в поле формы | В поле ввода вводится фамилия | Выполнено успешно |
| **Продолжение приложения Г** | | | | | |
| ST\_009 | Страница с добавлением услуги | Войти в приложение, Зайти на страницу «Аккаунт», Нажать на голубой крест в правом верхнем углу, Нажать на «Добавить услуги по номеру телефона», Ввести неверный номер домашнего телефона и фамилию, Нажать на стрелку для перехода на следующую страницу | После ввода неверного номера домашнего телефона и фамилии, и перехода на новую страницу с помощью стрелки, должно всплывать окно в котором описывается, что данные введены неверные | При неверном заполнении полей выводится сообщение о ошибке «Неверный номер телефона или фамилия. Проверьте введенные данные» | Выполнено успешно |
| ST\_010 | Страница с добавлением услуги | Войти в приложение, Зайти на страницу «Аккаунт», Нажать на голубой крест в правом верхнем углу, Нажать на «Добавить услуги по номеру телефона», Ввести неверный номер домашнего телефона и фамилию, Нажать «Готово» в правом верхнем углу | После ввода неверного номера домашнего телефона и фамилии, и нажатии на «Готово» в правом верхнем углу, должно всплывать окно в котором описывается, что данные введены неверные | При неверном заполнении полей выводится сообщение о ошибке «Неверный номер телефона или фамилия. Проверьте введенные данные» | Выполнено успешно |
| ST\_011 | Страница «История платежей» | Войти в приложение, Зайти на страницу «История платежей» | После входа на страницу с историей платежей должны появиться контракты, выбор вида оплаты и платежи | На странице с историей платежей есть вся информация заявленная в требованиях | Выполнено успешно |
| ST\_012 | Страница «История платежей» | Войти в приложение, Зайти на страницу «История платежей», Нажать на «Все контракты», Выбрать контракт | При нажатии на «Все контракты» должен предоставляться доступ на действующие контракты. При нажатии на контракт происходит переход обратно на страницу «История платежей», но с информацией о данном контракте | При нажатии на «Все контракты» происходит переход на страницу с контрактами, и после выбора контракта выдается информация о выбранном контракте | Выполнено успешно |
| **Продолжение приложения Г** | | | | | |
| ST\_013 | Страница «История платежей» | Войти в приложение, Зайти на страницу «История платежей», Нажать на «Платежи из Мой Белтелеком», Выбрать «Все платежи» | При нажатии на «Платежи из Мой Белтелеком» должен происходить переход обратно на страницу «История платежей», но с информацией о платежах | При нажатии на «Платежи из Мой Белтелеком» происходит переход на страницу обратно на страницу «История платежей», но с информацией о платежах | Выполнено успешно |
| ST\_014 | Страница «Новости» | Войти в приложение, Зайти на страницу «Новости» | На странице с новостями находятся новости и фильтр | На странице с новостями есть вся информация заявленная в требованиях | Выполнено успешно |
| ST\_015 | Страница «Новости» | Войти в приложение, Зайти на страницу «Новости», Нажать на символ фильтра в правом верхнем углу, Выбрать разные категории фильтра | При нажатии на символ фильтра происходит переход на страницу под названием «Фильтры», где осуществляется выбор категории фильтра | После нажатия на символ фильтра происходит переход на новую страницу с выбором категории фильтра | Выполнено успешно |
| ST\_016 | Страница «Новости» | Войти в приложение, Зайти на страницу «Новости», Нажать на символ фильтра в правом верхнем углу, Выбрать разные категории фильтра и применить их | После выбора категории фильтра, фильтрация новостей осуществляется по выбранным категориям | Фильтрация новостей осуществляется по выбранным категориям | Выполнено успешно |
| ST\_017 | Страница «Прочее» | Войти в приложение, Зайти на страницу «Прочее» | После входа на страницу с прочим должны появиться множество пунктов | На странице с прочим есть вся информация заявленная в требованиях | Выполнено успешно |
| ST\_018 | Страница «Прочее» | Войти в приложение, Зайти на страницу «Прочее», Нажать на «Подключить услугу» | После нажатия «Подключить услугу» должен происходить переход на страницу «Подключить услугу» с формами «Добавить услугу Белтелеком в аккаунт» и «Подключиться к Белтелеком» | После нажатия «Подключить услугу» происходит переход на страницу «Подключить услугу» с формами «Добавить услугу Белтелеком в аккаунт» и «Подключиться к Белтелеком» | Выполнено успешно |
| **Продолжение приложения Г** | | | | | |
| ST\_019 | Страница «Прочее» | Войти в приложение, Зайти на страницу «Прочее», Нажать на «Подключить услугу», Нажать на «Добавить» | После нажатия на добавление услуги в аккаунт происходит переход на новую страницу с добавлением услуги | Происходит переход на новую страницу после нажатия на добавление услуги | Выполнено успешно |
| ST\_020 | Страница «Прочее» | Войти в приложение, Зайти на страницу «Прочее», Нажать на «Подключить услугу», Нажать на «Добавить», Заполнить поле ввода логина | Ввод логина в поле формы | Логин вводится в поле | Выполнено успешно |
| ST\_021 | Страница «Прочее» | Войти в приложение, Зайти на страницу «Прочее», Нажать на «Подключить услугу», Нажать на «Добавить», Заполнить поле ввода пароля | Ввод пароля в поле формы | Пароль не вводится в поле |  |
| ST\_022 | Страница «Прочее» | Войти в приложение, Зайти на страницу «Прочее», Нажать на «Подключить услугу», Нажать на «Добавить» для добавление услуги в аккаунт, Заполнить поле ввода пароля, Нажать на показ пароля | При вводе пароля можно просмотреть его путем нажатия на символ глаза, который должен показываться постоянно. При нажатии на этот символ пароль просматривается | Так как пароль не вводится и символ глаза не присутствует, то невозможно его просмотреть с данным символом. При обстоятельстве, когда символ показывается, но пароль не вводится, после нажатия на символ глаза в строке ввода пароля появляется одна точка, которая не просматривается, а символ глаза пропадает |  |
| ST\_023 |  | Войти в приложение,  Зайти на страницу «Прочее»,  Нажать на «Подключить услугу»,  Нажать на «Добавить»,  Нажать на «Добавить услуги по номеру телефона», Ввести номер домашнего телефона | Стоит ограничение (7 символов) на ввод цифр в поле для ввода домашнего номера | При вводе максимального количества цифр, дальнейший ввод невозможен | Выполнено успешно |

## ПРИЛОЖЕНИЕ Д

## Добавление тестовой документации в github репозиторий

…

Рисунок Д.1 –Отражение тестовой документации в репозитории