

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



Основная профессиональная образовательная программа
Направление подготовки 09.03.01 Информатика и вычислительная техника
Направленность (профиль) «Технологии разработки программного обеспечения»
форма обучения – очная

Курсовая работа

«Управление проектом по разработке многопользовательской
интеллектуальной онлайн-викторины»

Обучающегося 3 курса
Нюхалова Дениса Глебовича

Научный руководитель:
Кандидат физико-математических наук, доцент
Жуков Николай Николаевич

Санкт-Петербург
2019

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ	2
ВВЕДЕНИЕ.....	3
1.1 Модели и методологии разработки	4
1.2 «Анализ предметной области. Выбор методологии для управления проектом»	13
«Обзор процесса проектирования и разработки на практике».....	16
2.1 «Этапы проектирования и разработки»	16
2.2 «Выбор подходящих средств проектирования»	23
ЗАКЛЮЧЕНИЕ	24
СПИСОК ЛИТЕРАТУРЫ.....	25
ПРИЛОЖЕНИЕ 1	27
ПРИЛОЖЕНИЕ 2	29

ВВЕДЕНИЕ

Сегодня, чаще всего, разработка программного обеспечения – это огромный труд команды из нескольких человек. Очевидно, что процесс создания чего-то нового разными людьми, требует разделения обязанностей и четкого понимания каждым членом команды своих целей, возможностей, времени и т.д. на каждом этапе разработки. На самом деле, даже человеку, разрабатывающему программу в одиночку будет очень сложно обойтись без предварительного проектирования, распределения сил и времени. Именно для наибольшей эффективности и упрощения процесса создания ПО были созданы модели и методологии разработки.

В данной курсовой работе, автор ставит перед собой цель, на примере собственного программного проекта, изучить **предметную область** проекта: различные модели и методологии разработки программного обеспечения. Планируется проанализировать различные парадигмы разработки: рассмотреть принципы и понятия, лежащие в их основе, изучить актуальность и применимость тех или иных моделей и методологий по отношению к различным программным продуктам.

Предметом данной работы является исследование этапов проектирования авторского программного продукта – многопользовательской интеллектуальной онлайн-викторины, с названием «ЧГК онлайн».

«Введение в предметную область, обзор моделей и методологий разработки»

1.1 Модели и методологии разработки

Начать стоит с того, что стоит за определениями «модель» и «методология» разработки ПО. Для этого, рассмотрим этапы, через которые проходит любой программный продукт в процессе создания:

1. Подготовка.

На данном этапе проводится анализ аналогичных программ, собирается информация об их сложности, рентабельности, аспектах реализации. На ее основе делаются выводы об актуальности самой задумки и целесообразности ее реализации.

2. Проектирование.

На данном этапе обсуждаются и в последствии определяются элементы и компоненты ПО, безопасность, архитектура, интерфейсы, типы данных, продумывается и рисуются дизайны и макеты.

3. Создание.

На данном этапе начинается непосредственно разработка: написание кода, составление документации, тестирование, размещение.

4. Поддержка.

На данном этапе проводится отладка продукта, выпуск обновлений, оценка производительности.

То, что происходит с программным обеспечением на каждом из этапов жизненного цикла и какие из этих этапов, оно вообще проходит, можно назвать **моделью** разработки продукта. **Методология**, в свою очередь, — это применение той или иной модели на практике, она включает в себя набор методов по управлению разработкой: это правила, техники и принципы, которые делают её более эффективной.

Одними из самых известных моделей разработки являются:

- code and fix;
- waterfall model;
- V-model;
- incremental model;
- iterative model;
- spiral model;
- chaos model;
- prototype model.

Разберем некоторые из этих моделей поподробнее.

Начнем с одной из самых известных и самых старых моделей разработки, придуманной еще в 1970 году Уинстоном Ройсом – waterfall model.

Ее суть заключается в том, что разработка идет поэтапно, последовательно проходя одну фазу за другой. Подразумевается, что переход от одной фазы разработки к другой происходит только после полного завершения предыдущей фазы. Переходов назад либо вперед или их перекрытия не происходит. Эти этапы включают в себя:

1. Определение требований;
2. Проектирование;
3. Конструирование (также «реализация» либо «кодирование»);
4. Воплощение;
5. Тестирование и отладка (также «верификация»);
6. Инсталляция;
7. Поддержка;

Результатом первого этапа является список требований к ПО. После этого начинается фаза проектирования, обязательной частью которой является разработка документов, регламентирующих для программистов способ и план реализации составленных ранее требований. Следующими этапами являются реализация и воплощение. Команда разработчиков делегирует обязанности и приступает к созданию отдельных компонентов приложения. Затем происходит

интеграция отдельных компонентов, разрабатываемых различными командами программистов. После завершения разработки, производится тестирование и отладка продукта. На этой стадии устраняются все недочёты, появившиеся на предыдущих этапах разработки. После этого программный продукт внедряется и обеспечивается его поддержка — внесение новой функциональности и устранение ошибок.

Структуру waterfall модели можно представить при помощи схемы, изображенной на рисунке ниже

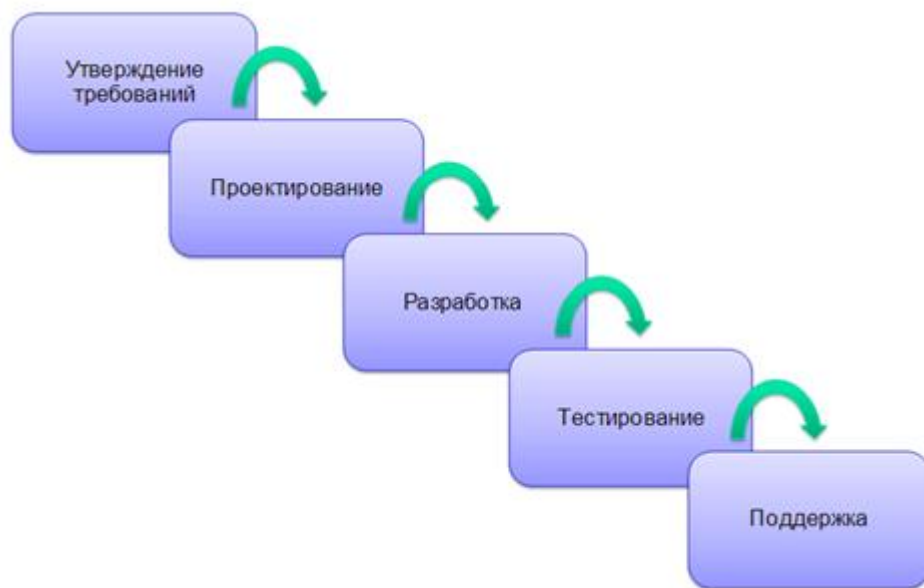


Рис. 1, схема waterfall модели

Такая модель имеет ряд преимуществ и недостатков. К преимуществам можно отнести:

- Эффективность такого подхода в плане времени. Статистически, в водопадной модели 20-40% времени уходит на утверждение требований и проектирование, 30-40% времени - на кодирование, а остальное - на тестирование и внедрение. Время, которое тратится на первых двух этапах, существенно снижает затраты на более поздних этапах.

- Фазы понятные и легко объяснимые. В каскадной модели можно легко и быстро разобраться, а впоследствии начать использовать для разработки.
- В этой модели делается большой упор на документации (документы требований и проектные документы), а также на исходный код.

Среди недостатков можно выделить:

- Изменение требований может сильно нарушить принципы waterfall модели. Это приводит к перепроектированию, переработке и повторному тестированию, а также к увеличению затрат.
- Есть риск неправильно оценить будущие трудности при разработке нового программного продукта или функции, и в этом случае, требуется пересмотр проекта, что также ведет к увеличению временных и финансовых затрат.

Таким образом водопадная модель будет эффективна для проектов, в которых требования и границы прозрачны и точно известны в начале жизненного цикла проекта.

Методологиями, воплощающими водопад можно назвать одноименную waterfall, которая не отличается от исходной модели, а также «модель сашими» Питера ДеГрейса (водопад с перекрывающимися фазами), водопад с подпроектами и водопад с уменьшением риска. Существуют и другие комбинации моделей разработки программного обеспечения, такие как «инкрементная водопадная модель».

Полной противоположностью модели waterfall можно назвать agile. Из названия становится понятно, что сама модель подразумевает гибкость в отношении разработки ПО. Впервые эта модель была описана в 2001 году в «Манифесте гибкой разработки программного обеспечения». Он включает в себя 4 ценности и 12 принципов. Ценности и принципы приведены ниже.

Ценности

1. Люди и взаимодействие важнее процессов и инструментов.
2. Работающий продукт важнее исчерпывающей документации.
3. Сотрудничество с клиентом важнее согласования условий контракта.
4. Готовность к изменениям важнее следования первоначальному плану.

Принципы:

1. Наивысшим приоритетом является удовлетворение потребностей клиента, благодаря регулярной и ранней поставке ценного программного обеспечения.
2. Изменение требований приветствуется, даже на поздних стадиях разработки.
3. Работающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев.
4. На протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе.
5. Над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, создайте условия, обеспечьте поддержку и полностью доверьтесь им.
6. Непосредственное общение является наиболее практичным и эффективным способом обмена информацией как с самой командой, так и внутри команды.
7. Работающий продукт — основной показатель прогресса.
8. Инвесторы, разработчики и пользователи должны иметь возможность поддерживать постоянный ритм бесконечно.
9. Постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта.

10. Простота — искусство минимизации лишней работы — крайне необходима.
11. Самые лучшие требования, архитектурные и технические решения рождаются у самоорганизующихся команд.
12. Команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы

Исходя из принципов и ценностей Agile, можно выделить следующую закономерность, присутствующую во многих методологиях, основанных на Agile: В Agile, разработка имеет повторяющийся характер. С определенной периодичностью команда разработчиков представляет заказчику свои наработки, а он в свою очередь оценивает их и определяет, удовлетворяет ли его результат или нет.

Этот подход изображен на схеме ниже:

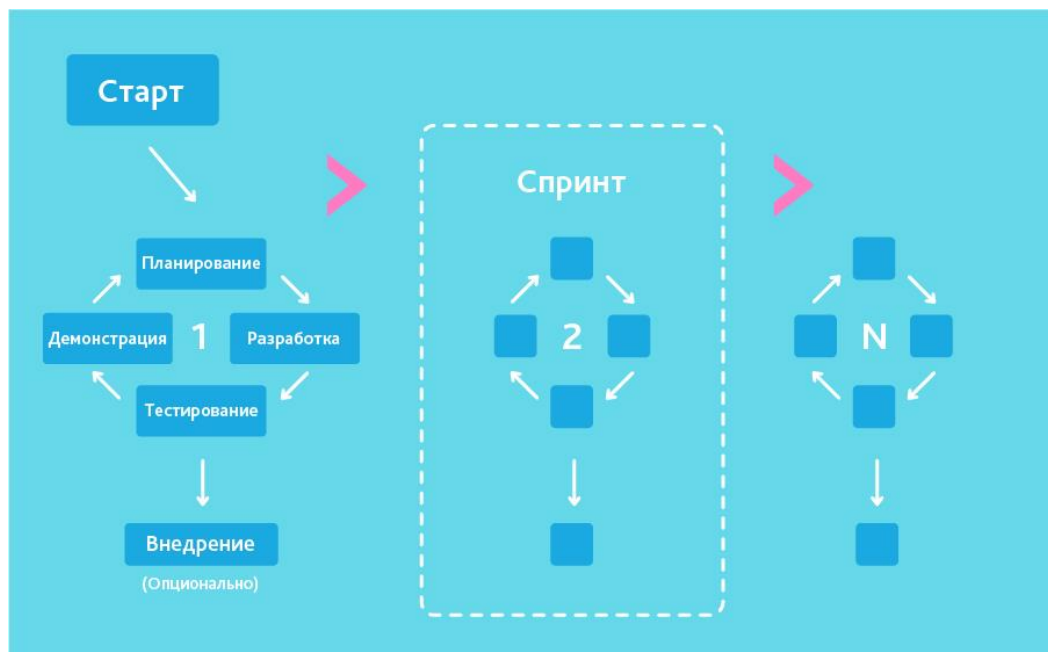


Рис. 2, один из подходов Agile

К преимуществам Agile подхода можно отнести:

- переделывание и повторяющаяся разработка, и переработка компонентов приложения подразумевается уже в самих принципах Agile. Заказчик может вносить любые изменения и правки в программный продукт, а следовательно, вероятность того, что он будет удовлетворен конечным продуктом, а в последствии будет и дальше с вами сотрудничать, возрастает;
- не обязательно нужно иметь полное представление о том, что вы хотите получить в конце. Нет необходимости тратить много времени на просчет рисков, бюджета и проектировку всего приложения;
- большое количество методологий и фреймворков, основанных на agile, можно выбрать принцип разработки, подходящий именно вашей команде.

К недостаткам же можно отнести следующие пункты:

- есть риск уйти в «бесконечную» разработку. Из-за постоянных усовершенствований и дополнений готовый продукт может так никогда и не выйти, а силы и бюджет команды могут со временем закончиться;
- Так как модель agile во многом направлена на промежуточные результаты, у разработчиков может возникнуть мотивация решать все поступившие задачи простейшим и быстреешим возможным способом, при этом зачастую не обращая внимания на правильность кода с точки зрения требований нижележащей платформы. Это может привести к снижению качества продукта и накоплению дефектов;

Можно сделать вывод о том, что гибкая модель будет идеальной для IT-компаний, стартапов, проектов в инновационных сферах, где имеется потребность в создании новых, не использующихся ранее функций.

Как уже было упомянуто в преимуществах agile, данная модель имеет большое количество методологий и фреймворков. Важно уточнить, что фреймворк — это более сформированная методология со строгими правилами. Среди методологий и фреймворков можно выделить:

- Scrum. В основе данного фреймворка лежит четкое разделение по ролям. Выделяется 3 основных роли:
 - Product owner – он отвечает за максимизацию ценности для заказчика.
 - Scrum master – его область это максимизация эффективности, посредством, помощи, обучении и мотивации команде.
 - Development team – команда программистов, разрабатывающая продукт.

Разработка подразделяется на спринты — которые длятся от недели до четырех недель и заканчиваются выпуском части продукта.

«Скрам отлично работает, когда сотрудники самомотивированы, ответственны. А еще, когда не до конца понятно, что делать, и вы постоянно ищете, как улучшить процессы разработки ПО» -

выдержка из интервью с Владимиром Морозовым, преподавателем Android разработки, скрам-мастером.

- Kanban. Данная методология строится вокруг Kanban-доски, виртуальной или реальной. Эта доска помогает в визуализации и оптимизации рабочего процесса, с ее помощью удобно отслеживать и визуализировать выполненную и предстоящую работу, организовать совместную работу, выстраивать общение команды и стандартизировать процесс разработки.

Kanban-доска состоит из трех частей: «Запланировано», «В работе» и «Сделано», на которых располагаются задачи, поставленные перед командой разработчиков. Доску не обязательно должна иметь именно

такое разделение, ее можно настроить в соответствии с процессом, принятым в той или иной команде, в зависимости от ее размеров, структуры и целей.

Методика Kanban основана на полной прозрачности работы и обмене информацией по ресурсам в режиме реального времени. Таким образом, Kanban-доска становится единственным достоверным источником информации о работе команды.

- eXtreme Programming. В eXtreme Programming или экстремальном программировании, ключевыми являются 3 основных практики:
 1. Парное программирование. Суть такого программирования понятна из названия. Над одним программным компонентом работают сразу два разработчика: первый пишет код, а второй отлаживает его и исправляет ошибки, затем они меняются местами. В процессе разработки появляются две разных реализации, затем выбирается лучшая из них.
 2. Много внимания уделяется тестированию. Приоритетным является TDD (test-driven development) подход — разработка через тестирование. Это значит, что сначала пишутся тесты, а только после этого разрабатывается компонент программы.
 3. У каждого члена команды есть полный доступ ко всему написанному коду. Его можно свободно изменять, но в случае, если изменения привели к ошибкам, тот член команды, который внес изменения, должен все исправить.

Экстремальное программирование предполагает также максимально быструю работу. Чем короче релизы, тем лучше качество продукта.

Интеграция новых частей в основной код происходит так быстро, как это возможно. Как только тесты показали, что функция работает корректно, она интегрируется в систему.

Конечно, помимо agile и waterfall существует еще много моделей разработки. Решение рассмотреть именно их показалось автору оправданным из за того, что они являются простыми и понятными, а в сути своей отражают два разных подхода к управлению программными проектами: детерминированный и адаптивный.

1.2 «Анализ предметной области. Выбор методологии для управления проектом»

Автор курсовой работы поставил перед собой задачу реализовать многопользовательскую онлайн викторину «ЧГК онлайн». Данный проект был вдохновлен успехом знаменитой советской и российской интеллектуальной телевизионной игры «Что? Где? Когда?». В рамках подготовки к выбору методологии был проведен поиск идейно схожих проектов в интернете.

Самыми популярными являются следующие программные продукты:

- Игра, размещенная на платформе «Яндекс.Игры» с названием «Что? Где? Когда?»

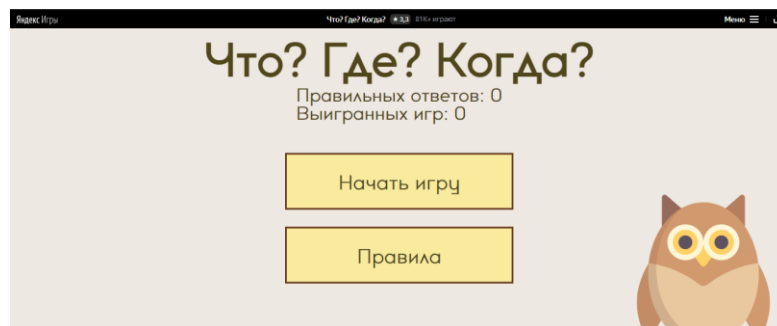


Рис. 3, «Что? Где? Когда?»

- Мобильное приложение от «Mail.ru Group» под названием «Викторина Что? Где? Когда? Своя игра. Квиз онлайн» для платформ iOS и Android.

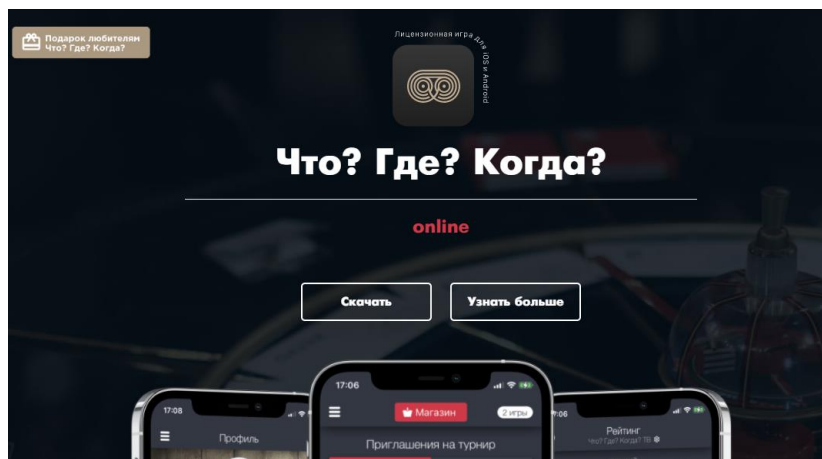


Рис. 4, лицензионная игра «Что? Где? Когда?»

Из главных особенностей данных викторин, можно выделить их отличающуюся от телевизионной структуру. Структура данных квиз-игр, следующая: игроку задается вопрос один из присутствовавших в телеигре вопросов, за 1 минуту он должен ответить на него и, в случае если ответ был правильный ему начисляются баллы.

Уникальность авторского приложения, в отличие от приложений-конкурентов будет заключаться в максимальном приближении онлайн формата игры к формату телевизионному. В понимании автора, игра вызывает интерес и имеет популярность, во многом, благодаря командной игре участников. Исходя из анализа и пожеланий автора, можно сформировать список требований к проектируемому программе:

1. Суть игры максимально близка к телевизионному формату: есть ведущий, шесть игроков и 13 вопросов, разыгрываемые случайным образом. Игра ведется до 6 очков.
2. Игра должна быть мультиплеерной, у игроков должна иметься возможность обсуждать вопросы при помощи микрофона или чата.
3. Должна быть разработана база вопросов для игры.

4. В качестве уникальной функциональной особенности игры, у игроков будет возможность создавать собственные «пакеты» с вопросами, которые можно разыгрывать в ходе игры.

Планирующийся продукт имеет ряд уникальных для игр-конкурентов особенностей. Из проведенного ранее обзора моделей разработки, самым подходящим видится agile подход, ведь, исходя из требований можно сделать вывод, о наличии потребности в создании новых, не использующихся ранее функций.

Однако, стоит учесть и ряд других аспектов, ранее не обозначенных в процессе описания требований и анализе конкурентов. Разработка приложения будет происходить силами одного человека, а не команды разработчиков. Agile методологии и фреймворки, в массе своей нацелены на обширный штат программистов. Также, автор не видит необходимости в изменении требований, обозначенных выше.

Поэтому, наиболее эффективной моделью проектирования видится waterfall model. Такая модель видится автору понятной, эффективной по потраченным силам и времени.

«Обзор процесса проектирования и разработки на практике»

2.1 «Этапы проектирования и разработки»

Была выбрана модель waterfall. Как известно, она состоит из 7 этапов. Первым является утверждение требований.

Требования к реализации были обозначены в предыдущей главе. Переформулируем их для более удобного создания технического задания:

1. Структура и аспекты игры, следующие: максимальное количество участников игры – 7. Ведущий, проверяющий ответы и шесть игроков, разыгрывающие вопросы. Минимальным количеством игроков будет 4 человека: 1 ведущий и 3 игрока. Случайным образом разыгрывается 13 вопросов. Игра ведется до 6 очков, в случае правильного ответа «очко» присуждается команде игроков, в случае неправильного - ведущему.
2. У игроков должна иметься возможность обсуждать вопросы при помощи микрофона или чата.
3. Должен присутствовать функционал, позволяющий составлять собственные вопросы. Также должна быть возможность выбирать вопросы, которые были разыграны в телевизионной версии игры.

Сформировав требования к продукту, можно приступить к следующему этапу разработки – проектированию. Первым делом, на данном этапе были задокументированы требования к ПО. Для этого было составлено техническое задание. Его содержание представлено на рисунке ниже. В данном техническом задании, объявленные выше требования были отражены следующим образом:

3. Требования к программе

1. Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

1. Задавать игрокам интеллектуальные вопросы
2. Давать возможность игрокам обсуждать вопросы в группе из 3-6 человек
 - a. Возможность обсуждать вопросы при помощи текстового чата
 - b. Возможность обсуждать вопросы при помощи голосового чата
3. Разработать базу данных «классических» вопросов из телеверсии шоу «Что? Где? Когда?»
4. Сохранять прогресс игры на время игровой сессии
5. Разработать функциональную возможность по созданию собственных пакетов вопросов.
6. Давать возможность аутентификации в свой игровой аккаунт при помощи логина и пароля или аккаунта Google

Рис. 6, техническое задание, требования к функциональным характеристикам

Также, на этапе проектирования имеет смысл определиться со средствами и способом реализации приложения. Выбор был сделан в пользу web-приложения в силу следующих причин:

1. Web-приложение проще в плане реализации многопользовательской составляющей игры. В web разработке реализация таких функций, как сессии или передача сообщений по TCP соединению, видится для автора более простой и понятной.
2. В web-приложении, в отличие от, например, мобильного, имеется больше свободы и возможностей для конфигурации интерфейса. Автор не видит необходимости в реализации адаптивного интерфейса под мобильный форм-фактор, ведь в таком случае программа может лишиться или сделать неудобными такие функциональные особенности приложения, как, например, чат.

Выбор web реализации, был отражен в техническом задании, в разделе «Условия эксплуатации»:

4. Условия эксплуатации

1. Требования к составу и параметрам технических средств

В состав технических средств должен входить ПК, с системным программным обеспечением Windows 7/8/10, Linux или MacOS, с установленным на нем браузером из списка ниже:

- Internet Explorer 8 и выше
- Opera 10.0 и выше
- Google Chrome 18.0 и выше
- Mozilla Firefox 13.0 и выше

2. Требования к информационной и программной совместимости

Разработка интернет-сервиса производится средствами языка программирования JavaScript и его фреймворков, языка разметки html и формального языка стилей CSS.

Рис.6, техническое задание, условия эксплуатации

Затем, для верного представления функций и возможностей программы на этапе разработки, возникла необходимость в создании диаграммы проектируемой системы. Выбор пал на use-case диаграмму, так как для разработки web-приложения будет использоваться такой язык программирования, как JavaScript. В JavaScript разработка приложений ведется преимущественно в процедурной, функциональной и императивной парадигмах. И хоть, не исключено использование автором в процессе разработки объектно-ориентированных конструкций, для отображения поведения которых используется UML диаграмма классов, было принято решение остановиться на use-case диаграмме, отобразив с ее помощью отношения программных компонентов и пользователей в проектируемой системе. Разработанную диаграмму можно увидеть на рисунке ниже.

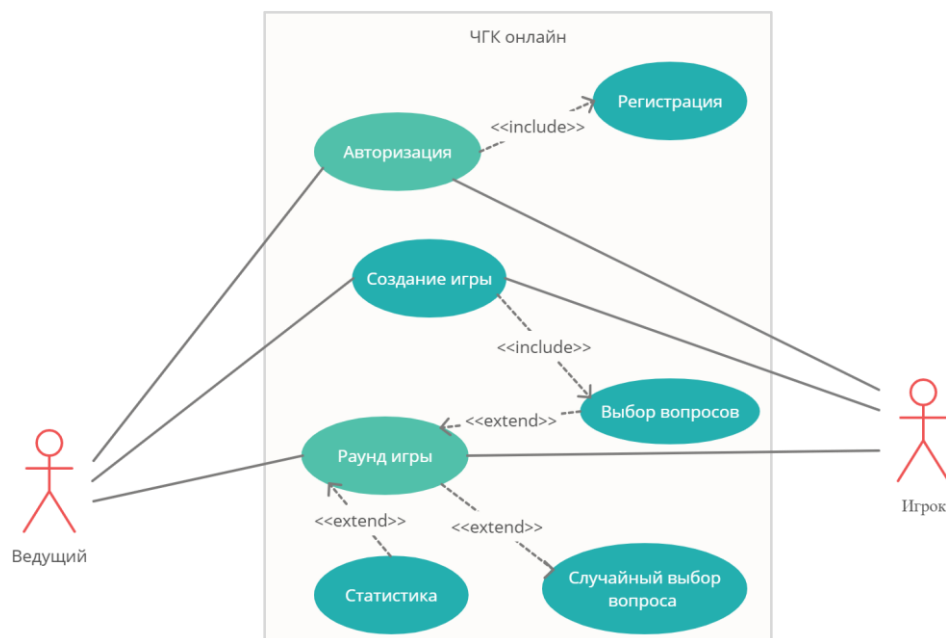


Рис. 6, use-case диаграмма

Следующим шагом для успешного завершения этапа проектирования, являлось продумывание UI/UX дизайна приложения и составление макета. Часть макета представлена на рисунке ниже. Полный макет представлен в приложении 1.

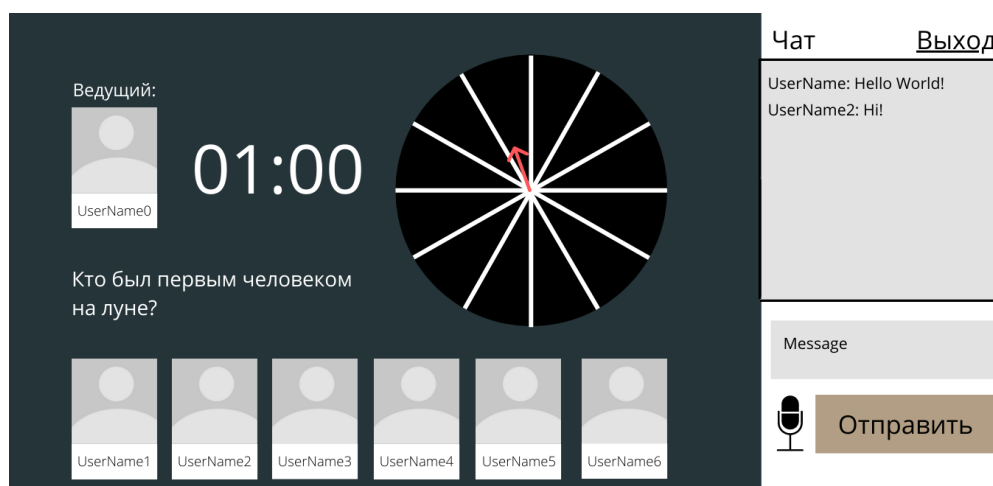


Рис.7 Часть макета

На приложенной части макета изображен момент игрового процесса. Для взаимодействия пользователя с функциональными частями приложения были использованы спроектированы элементы (некоторые из них отображены только в приложении):

- «колесо», при помощи которого будут случайным образом выбираться вопросы;
- чат в правой части интерфейса;
- кнопка включения/отключения микрофона;
- таймер, показывающий оставшееся до ответа времени;
- кнопка, для ведущего, показывающая все вопросы;
- окно авторизации;
- окно регистрации, с возможностью зарегистрироваться через google.

Для оптимизации разработки на следующем этапе waterfall модели была составлена диаграмма Ганта. Диаграмма Ганта — это горизонтальная столбчатая диаграмма с временной шкалой, которая используется для иллюстрации плана работ по проекту с привязкой ко времени. Диаграмма Ганта для разработки «ЧГК онлайн» представлена в приложении 2.

В этой диаграмме также указано время на разработку UX/UI дизайна, но непосредственно разработка будет включать в себя 3 «подэтапа»:

1. Верстка шаблонов страниц.

На данном этапе создается разметка web-страниц.

2. Прописывание поведения web-страниц и взаимодействия сайта с сервером.

Данный «подэтап» начинается, когда готова, хотя бы, половина шаблонов страниц. Это сделано для оптимизации работы по времени. Эффективнее, несложным по структуре шаблонам, по типу страниц регистрации, сразу начинать прописывать поведение.

3. Развертывание на сервере.

Размещение программы на сервере начинается, когда большинство программных компонентов, не относящихся к взаимодействию

приложения с сервером уже готовы. Серверные части ПО, в свою очередь, дорабатываются во время размещения.

Пройдя по всем этим «подэтапам», можно смело заявить, что этап waterfall «разработка» завершен. Следующим на очереди стоит этап тестирования.

В рамках первого этапа тестирования, силами разработчика(ов) будет проведена внутренняя проверка графического интерфейса. Не имеет смысла создавать систему интегрированных или unit тестов, так как в приложении будет мало относительно сложных логических структур. При тестировании GUI «ЧГК онлайн» стоит обратить внимание на следующее:

- расположение, размер, цвет, ширину и длину элементов;
- возможность ввода букв или цифр в окне регистрации и чата;
- реализацию функционала приложения с помощью графических элементов;
- корректное размещение сообщений об ошибках регистрации, уведомлений;
- читабельность, цвет, размер и расположение шрифта, а также орфографию текста;
- переходит ли курсор из текстового в «поинтер» при наведении на активные элементы, выделяются ли выбранные элементы;
- проверить расположение и отображение всех элементов при различных разрешениях экрана, а также при изменении размера окна браузера.

Следующей частью этапа тестирования является разработка программы юзабилити тестирования. В ней стоит рассмотреть следующие аспекты тестирования:

- обозначить методы тестирования, например юзабилити-тестирование в формате «think-aloud» или опросник субъективной удовлетворенности;
- описать условия проведения тестирования;
- подробно прописать процедуру исследования, обозначить пожелания к рекрутингу респондентов, аппаратуру для тестирования, инструкцию для респондентов и т.д.

По разработанной программе можно организовать процесс тестирования. Организовать его можно как своими силами, так и вынеся на аутсорсинг.

Заключительным этапом waterfall является поддержка. В проекте «ЧГК онлайн» поддержку можно осуществить, создав специальную почту для сбора обратной связи и сообщений об ошибках. Целесообразно также организовать регулярную проверку этой почты или организовать пересылку писем на почту автора приложения.

Также можно разместить исходный код приложения на платформе GitHub. Это также позволит эффективно собирать обратную связь, а также изучать предложения по доработке приложения другими пользователями-разработчиками.

В конце этой части курсовой работы, хочется отметить эффективность waterfall модели для данного проекта. Уже на реализации этапа проектирования, образуется набор таких документов, как техническое задание, use-case диаграмма, макет и диаграмма Ганта. С их помощью процесс разработки становится простым и понятным, а также удобным для модерации и управления.

Так как на первых этапах проектирования были четко определены, а в последствии закреплены в техническом задании, требования к функциям программного продукта, необходимости в приделывании проекта из-за изменения требований отпала.

Таким образом, получилось, насколько это было возможно, ощутить все преимущества модели водопад, при этом не столкнувшись с ее недостатками.

2.2 «Выбор подходящих средств проектирования»

В ходе управления проектом по модели водопад, на этапе проектирования были разработаны: техническое задание, use-case диаграмма, диаграмма Ганта и UI/UX макет. В данной главе будет дан краткий обзор на средства проектирования, при помощи которой создавалась документация по управлению программным продуктом «ЧГК онлайн».

Техническое задание было сделано при помощи приложения из офисного пакета Microsoft, «Microsoft Office Word». Данная издательская система позволила быстро создать техническое задание с учетом всех ГОСТов и стандартов. Она была выбрана из-за широкой распространённости и обширного и простого в освоении инструмента для создания технических текстов.

Для создания use-case диаграммы использовался онлайн-сервис «creately». Это приложение позволяет создавать диаграммы различных типов и сложности, встраивать изображения, а также использовать готовые шаблоны. Данный сервис был выбран из-за простоты в освоении и качественных шаблонов.

При создании диаграммы Ганта использовался сервис «toggl». Из его особенностей можно выделить возможность приглашать других пользователей, оповещения по e-mail, а также возможность использовать сервис как календарь. Он позволил автору эффективно и быстро распланировать разработку проекта.

Макет был создан при помощи средства «Canva». Возможно, его функционала может быть недостаточно для проектирования сложных интерфейсов. Однако, для реализации выбранного автором стиля инструментария «Canva» было достаточно.

ЗАКЛЮЧЕНИЕ

Целью данной работы было рассмотреть предметную область «модели и методологии ПО». В ходе работы был проведен краткий анализ моделей Waterfall и Agile, а также трёх методологий. Были сделаны выводы об эффективности той или иной модели в различных программных проектах.

Данный анализ позволил автору определиться с моделью разработки, используемой в предмете курсовой работы – «исследование этапов проектирования авторского программного продукта». Были успешно рассмотрены все этапы управления и разработки проекта, а также проведен анализ сервисов проектирования.

СПИСОК ЛИТЕРАТУРЫ

1. Модели и методологии разработки ПО // Geekbrains URL: <https://geekbrains.ru/posts/methodologies> (дата обращения: 15.12.2020).
2. Каскадная модель // Wikipedia URL: https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%81%D0%BA%D0%B0%D0%B4%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C (дата обращения: 15.12.2020).
3. Agile-манифест разработки программного обеспечения // Agile-манифест URL: <https://agilemanifesto.org/iso/ru/manifesto.html> (дата обращения: 15.12.2020).
4. Agile или Waterfall — какой вариант соответствует вашему бизнесу? // Worksection URL: <https://worksection.com/blog/waterfall-vs-agile.html> (дата обращения: 15.12.2020).
5. Agile и Scrum на практике: вопросы скрам-мастеру // Geekbrains URL: https://geekbrains.ru/posts/agile_scrum_gu (дата обращения: 15.12.2020).
6. Модели и методологии разработки стартапа // StartupJedi URL: <https://medium.com/%D1%81%D1%82%D0%B0%D1%80%D1%82%D0%B0%D0%BF-%D0%B4%D0%B6%D0%B5%D0%B4%D0%B0%D0%B9/%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8-%D0%B8-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D0%B8-%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8-%D1%81%D1%82%D0%B0%D1%80%D1%82%D0%B0%D0%BF%D0%B0-591fdd2101a2> (дата обращения: 15.12.2020).
7. ТЕСТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА // QAEvolution URL: <https://qaevolution.ru/testirovanie-po/vidy-testirovaniya-po/testirovanie-polzovatelskogo-interfejsa/> (дата обращения: 15.12.2020).
8. Ещё раз про семь основных методологий разработки // Habr URL: <https://habr.com/ru/company/edison/blog/269789/> (дата обращения: 15.12.2020).
9. Toggl URL: <https://toggl.com/> (дата обращения: 15.12.2020).
10. Canva URL: <https://www.canva.com/> (дата обращения: 15.12.2020).
11. Creately URL: <https://app.creately.com/> (дата обращения: 15.12.2020).

12. Лещева, И. А. Основы управления проектами : учебное пособие / И. А. Лещева, Э. В. Страхович. — Санкт-Петербург : СПбГУ, 2011. — 96 с. — ISBN 978-5-9924-0059-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/47511> (дата обращения: 22.12.2020). — Режим доступа: для авториз. пользователей.
13. Управление инновационными и предпринимательскими проектами : учебное пособие / Н. В. Власова, А. М. Губернаторов, И. А. Львов [и др.]. — Москва : Научный консультант, 2018. — 122 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/111834> (дата обращения: 22.12.2020). — Режим доступа: для авториз. пользователей.
14. Основы управления ИТ-проектами : учебное пособие / составители Е. Р. Кирколуп [и др.]. — Барнаул : АлтГПУ, 2017. — 176 с. — ISBN 978-5-88210-861-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/112162> (дата обращения: 22.12.2020). — Режим доступа: для авториз. пользователей.
15. Некрасова, Н.А. Управление проектами как инструмент подготовки выпускной квалификационной работы магистранта в сфере-ИКТ / Н.А. Некрасова, А.Т. Аяпбергенова // Вестник Кыргызско-Российского славянского университета. — 2014. — № 1. — С. 128-132. — ISSN 1694-500X. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/journal/issue/291967> (дата обращения: 22.12.2020). — Режим доступа: для авториз. пользователей.

ПРИЛОЖЕНИЕ 1

Добро пожаловать в
ЧГК онлайн

Логин:

Пароль:

Войти

Регистрация


Регистрация

Логин:


Пароль:

Подтверждение
пароля:

Зарегистрироваться
через google



Зарегистрироваться



UserName


Кол-во выигранных игр: 5

Изменить фото

Создать пакет вопросов


Создать игру

Ведущий:




UserName0


01:00




Кто был первым человеком на луне?




UserName1




UserName2




UserName3



UserName4



UserName5




UserName6

Чат

Выход


UserName: Hello World!
 UserName2: Hi!

Message



Отправить

Ведущий:




UserName0

01:00


Сейчас играет:
 Кто был первым человеком на луне?

Пауза


Показать все вопросы




UserName1




UserName2




UserName3



UserName4



UserName5




UserName6

Чат

Выход

UserName: Hello World!
 UserName2: Hi!

Message



Отправить

ПРИЛОЖЕНИЕ 2

