Разработка программного обеспечения – это огромный труд команды из нескольких человек. Этот процесс 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.

*Swap slide*

Предметная область данного курсового 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».

*Swap slide*

Начать стоит с определения понятий модель и методология. 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.

*Swap slide*

Одной из самых первых и известных моделей является каскадная или waterfall. В ней 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 вперёд или их перекрытия не происходит. Эти этапы включают в себя:

1. Определение требований. На нем формируется список требований к ПО

2. Проектирование. Ее обязательной частью является разработка документов, регламентирующих для программистов способ и план реализации составленных ранее требований

3. Реализация и воплощение. Команда разработчиков делегирует обязанности и приступает к созданию отдельных компонентов приложения. Затем происходит интеграция отдельных компонентов, разрабатываемых различными командами программистов

4. Тестирование и отладка. На этой стадии устраняются все недочёты, появившиеся на предыдущих этапах разработки

6. Поддержка. Внесение новой функциональности и устранение ошибок.

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.

*Swap slide*

Следующей рассматриваемой моделью является Agile или гибкая методология. Ее можно назвать полной противоположностью модели waterfall. Из названия становится понятно, что сама модель подразумевает гибкость в отношении разработки ПО.

*Swap slide*

Впервые эта модель была описана в 2001 году в «Манифесте гибкой разработки программного обеспечения». Он включает в себя 4 ценности и 12 принципов.

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 определят, удовлетворяет ли его результат или нет.

*Swap slide*

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 быстрейшим возможным способом, при этом зачастую не обращая внимания на правильность кода с точки зрения требований нижележащей платформы.

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.

*Swap slide*

Agile имеет множество методологий, одними из самых популярных являются Scrum, Kanban, eXtreme Programming. Если говорить о каждом вкратце:

• 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.

• Scrum. В основе данного фреймворка лежит четкое разделение по ролям. Основных ролей 3 : Product owner, Scrum master и Development team. Разработка подразделяется на спринты — которые длятся от недели до четырех недель и заканчиваются выпуском части продукта.

• Экстремальное программирование. Данная методология подразумевает использование 3 основных практик: Парное программирования, test-driven development, доступ к коду у кждого члена команды. Оно предполагает также максимально быструю работу. Чем короче релизы, тем лучше качество продукта. Интеграция новых частей в основной код происходит так быстро, как это возможно

Помимо agile и waterfall существует еще много моделей разработки. Решение рассмотреть именно их показалось мне 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.

*Swap slide*

Автор курсовой работы поставил перед собой задачу исследовать управление проектом по созданию многопользовательской 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.

Были проанализированы попытки реализации данной игры изучены их положительные и отрицательные особенности. В отличии от приложений-конкурентов, уникальность авторского приложения, будет заключаться в максимальном приближении онлайн формата игры к формату телевизионному. Данная игра вызывает интерес и имеет популярность, во многом, благодаря командной игре участников.

*Swap slide*

Из проведенного анализа и авторских идей, был сформирован список требований к проектируемомой 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.

*Swap slide*

Планирующийся продукт имеет ряд уникальных для игр-конкурентов особенностей. Из проведенного ранее обзора моделей разработки, самой подходящей видится agile, ведь, исходя из требований можно сделать вывод, о наличии потребности в создании новых, не использующихся ранее функций, которые лучше реализовывать, перерабатывая компоненты приложения несколько раз.

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, в массе своей нацелены на обширный штат программистов. Также, автор не видит необходимости в изменении требований, обозначенных выше, а также четко определяет способы реализации уникальных функциональных особенностей.

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

*Swap slide*

Вернемся к структуре каскадной модели. На первой ступени мы сформировали требования к программе. Давайте зарегламентируем их в техническом задании.

*Swap slide*

В тз была задана более полная формулировка требований.

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 отличии от, например, мобильного, имеется больше свободы и возможностей для конфигурации интерфейса. Автор не видит необходимости в рализации 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»:

*Swap slide*

Определившись с формой приложения 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 системы. Разработанную диаграмму можно увидеть на слайде

*Swap slide*

Следующим шагом для успешного завершения этапа проектирования, являлось продумывание UI/UX дизайна приложения и составление макета.

Полный макет представлен в приложении 1 к курсовой работе а 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;

*Swap slide*

• таймер, показывающий оставшееся до ответа времени;

• кнопка, для ведущего, показывающая все вопросы;

*Swap slide*

• окно авторизации;

• окно регистрации, с возможностью зарегистрироваться через google.

*Swap slide*

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 указанно 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

*Swap slide*

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.

*Swap slide*

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.

*Swap slide*

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.**

*Swap slide*

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 создание с учетом всех ГОСТов и стандартов. Она была выбрана из-за широкой распространённости и обширного и простого в освоении инструментала 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.

*Swap slide*

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.