**Ретроспектива**

Студента гр. 588-1

Маковский Н.Д.

«\_\_» 2021

1. **Выполнена ли задача в полном объеме согласно ТЗ? Если задача выполнена не в полном объеме, то почему? Если в ходе разработки ТЗ частично или полностью пришлось изменить, то почему?**

Задача выполнена в полном объеме согласно ТЗ. В ходе разработки не пришлось изменять ТЗ.

1. **Выполнена ли задача в срок? Сколько потребовалось человеко-часов на выполнение всех лабораторных, включая написание пояснительной записки? Какие задачи заняли больше всего времени и создали больше всего трудностей? Были ли это форс-мажорные трудности или этих трудностей можно было избежать? Если программа разработана не в срок, то почему? Если какие-либо этапы разработки были завершены не в срок, то почему? Сколько времени заняло написание документации (составление отчетов, рисование диаграмм) от общего времени разработки?**

Задача выполнена в срок. На выполнение всех лабораторных работ, включая написание пояснительной записки, в сумме потребовалась примерно 6 дней, т.е. 144 часа.

Больше всего времени заняло выполнение 2-й лабораторной работы, так как в ней было необходимо разработать классы бизнес-логики приложения, «привыкнуть» к синтаксису языка C# (ранее практически не было опыта разработки программ на этом языком программирования) и к использованию xml-комментариев. Также довольно много времени заняло выполнение 3-й лабораторной работы: поскольку в ней создавался адаптивный пользовательский интерфейс, потребовалось дополнительно изучить основы верстки и разработки логики пользовательского интерфейса в WinForms по списку источников к этой лабораторной работе.

Написание всей документации в целом заняло приблизительно   
1/7 часть от общего времени разработки. На составление отчетов по лабораторным работам уходило в среднем по 2-3 часа, включая рисование диаграмм в Enterprise Architect. Написание пояснительной записки к проекту оказалось самым продолжительным по времени и потребовало около 5 часов.

1. **Было ли ТЗ написано в достаточном для реализации объеме? Были ли найдены в ТЗ ошибки или противоречия? Достаточно ли понятна работа приложения из предоставленных макетов интерфейса?**

ТЗ было написано в достаточном для реализации объеме, никаких ошибок или противоречий не было найдено. Назначение приложения, пользовательский интерфейс, требования к бизнес-логике подробно изложены в ТЗ, предоставленные макеты интерфейса дают довольно полное представление о работе приложения.

1. **Были ли найдены ошибки в примечаниях от руководителя?**

Ошибок не было найдено.

1. **Возникали ли какие-либо затруднения при работе со средой разработки, системой версионного контроля, редактором диаграмм? Удобна ли в использовании система версионного контроля? Удалось ли следовать индивидуальной модели ветвления при разработке?**

При работе со средой разработки (Visual Studio 2019 Professional) не было никаких затруднений, ранее уже приходилось работать в данной IDE. Первоначально возникали проблемы при работе в системе версионного контроля Git, например, один раз при создании дополнительных веток разработки возникла «путаница» с коммитами, также не сразу выработалась привычка как можно чаще делать коммиты. Это можно объяснить недостатком опыта работы с подобными системами, поэтому по мере разработки проекта данные проблемы возникали все реже, кроме того, зачастую трудности довольно быстро разрешались путем обращения к сторонним ресурсам (различные книги и интернет-ресурсы по работе с Git) и изучения необходимых для работы команд.

Определенные затруднения появлялись при работе с редактором диаграмм Enterprise Architect. Изначально интерфейс редактора был не совсем понятен, но к моменту составления диаграммы классов для   
5-й лабораторной работы в этой программе стало намного легче ориентироваться. Также не удалось воспользоваться возможностями расширения JetBrains Resharper: оно очень сильно замедляло работу Visual Studio, вплоть до «торможения» по несколько секунд при переключении вкладок, которое не удавалось устранить какими-либо другими способами, кроме отключения расширения.

Система версионного контроля Git очень удобна и достаточно понятна в использовании, имеет много преимуществ: хранение всех этапов разработки приложения, возможность быстрого восстановления версии при возникновении ошибок/потере данных на локальной машине, возможность продолжать разработку проекта с любой точки сохранения. Следовать индивидуальной модели ветвления при разработке также удалось.

1. **Возникли ли какие-либо затруднения при проведении приемочного тестирования? Сколько раз пришлось проводить приемочное тестирование и исправлять замечания, прежде чем заказчик принял проект? Если более одного раза, то почему? Что можно исправить или изменить, чтобы в будущем сдача проекта проходила быстрее?**

При проведении приемочного тестирования никаких трудностей не возникло, оно было успешно пройдено с первого раза. При самостоятельном проведении приемочного тестирования (до непосредственной сдачи проекта) было обнаружено несколько ошибок в работе приложения, но благодаря такой «самопроверке» их удалось своевременно устранить. Т.е. приемочное тестирование не только не вызвало затруднений, но и помогло заблаговременно исправить мелкие недочеты.

1. **Общее заключение. Чему удалось научиться в ходе выполнения лабораторных работ? Можно ли считать проект завершенным успешно?**

В ходе выполнения лабораторных работ приобретены навыки подключения к проекту сторонних библиотек через менеджер подключения пакетов NuGet, работы с библиотекой сериализации Newtonsoft.JSON.NET, написания юнит-тестов при помощи библиотеки NUnit. Получены знания о том, как организован процесс тестирования в разработке ПО, как происходит процесс сборки установочного пакета приложения (с применением программы создания сценариев для сборки установщика InnoSetup и его последующей автоматизации), какие существуют виды технической документации, когда и как составляется техническая документация. Также удалось научиться искать более простые методы решения определенных задач, приобрести больше опыта в области программирования на языке C#.

В целом проект можно считать успешно завершенным и готовым к использованию, так как приложение реализовано в полном объеме согласно ТЗ, а также получены новые навыки в разработке приложений.