**План управления конфигурациями**

|  |  |  |  |
| --- | --- | --- | --- |
| Номер версии | Дата изменений | Изменения | Автор изменений |
| Версия 1.0. | 20.12.15 | Документ был создан | Сорокин Станислав |
| Версия 2.0 | 15.02.16 | В пункте 3.4 изменились сроки выдачи задания и срок отправки заданий на проверку | Сорокин Станислав |

1. Общие положения
   1. Назначение документа

Данный документ предназначен для того, чтобы определить количество разработчиков, распределение их ответственностей в проекте, показать, как построен процесс разработки от анализа предметной области до тестирования приложения.

* 1. Модель жизненного цикла

Для разработки проекта была выбрана спиральная модель жизненного цикла, поэтому процесс разработки, согласно календарному плану, разбивается на отдельные витки, результатом каждого из которых служит полный набор документов, описывающих проект, а также соответствующую этим документам реализацию проекта.

* 1. Список определений

Документация – набор документов, включающий в себя спецификацию требований и сопутствующие ей документы.

Закрепленная роль – роль, которая выполняется участником проекта от начала до конца разработки.

Роль – активность, которую участник проекта выполняет в конкретный промежуток времени.

Участник – человек, входящий в команду разработчиков проекта.

1. Участники проекта
   1. Количество участников и распределение обязанностей

В проекте на данный момент участвуют 10 разработчиков.

Распределение ролей участников проекта не является жестким, то есть в ходе процесса разработки каждый участник может попробовать себя в нескольких ролях. Единственными закрепленными ролями являются роли менеджера и главного программиста.

В качестве непостоянных можно выделить следующие роли:

* составитель требований,
* проектировщик структуры web-интерфейса,
* проектировщик макетов страниц
* проектировщик UML и других схем и диаграмм,
* проектировщик базы данных,
* архитектор системы,
* front-end разработчик,
* back-end разработчик,
* тестировщик,
* контент-менеджер,
* составитель разделов и задач,
* разработчик тестов и чекеров для задач.

Участник может выполнять в один и тот же момент времени сразу несколько ролей.

* 1. Case-средства

В данном разделе приводится таблица с используемыми в разработке case-средствами и их назначением в рамках данного проекта.

|  |  |  |
| --- | --- | --- |
| № п/п | Case-средство | Назначение |
| 1 | Microsoft Word 2013 | Составление и редактирование документации |
| 2 | Microsoft Power Point 2013 | Создание презентации для отчетного мероприятия |
| 3 | NinjaMock | Прототипирование интерфейсов |
| 4 | Silverrun | Проектирование концептуальных моделей базы данных |
| 5 | ERwin | Проектирование логической и физической моделей базы данных |
| 6 | Microsoft Visio 2013 | Создание схем и диаграмм |
| 7 | UMLStar | Проектирование UML-диаграмм |
| 8 | PyCharm | IDE для реализации |
| 9 | Google Chrome | Тестирование работы системы, заполнение базы данных |
| 10 | Microsoft Excel 2013 | Создание отчетов о проделанной работе, хранение оценок участников |
| 11 | Skype | Коммуникация между участниками |
| 12 | Git | Система контроля версий |
| 13 | Trello | Система управления проектом |
| 14 | Django | Фреймворк для реализации |

1. Процесс разработки
   1. Организация работы внутри одного витка

Для реализации полноценного завершенного витка необходимо пройти следующие этапы:

1. Анализ и модификация требований. Включает в себя анализ предметной области, создание первичных требований для витка, проектирование сопутствующих диаграмм и схем, поясняющих те или иные процессы, а также внесение необходимых исправлений в эти документы по ходу разработки данного витка.
2. Проектирование Базы Данных. Включает в себя последовательное проектирование концептуальной, логической и физической модели базы данных.
3. Проектирование веб-интерфейса пользователя. Включает в себя создание документа, в котором указаны все web-страницы, которые присутствуют в данном витке, подробное описание содержания этих страниц, а также, в качестве приложения, макеты страниц.
4. Наполнение Базы Данных.
5. Разработка. Включает в себя реализацию всех функций системы, указанных в документации.
6. Тестирование. Включает в себя проверку работы приложения согласно документации.
7. Верификация. Включает в себя подтверждение выполнения того или иного задания. Этот процесс присутствует на всех этапах разработки.
   1. Система управления проектами

Для удобного и эффективного взаимодействия между участниками проекта, а также для выдачи, выполнения и верификации заданий используется система управления проектами Trello. Данная система содержит специальную доску, на которой размещаются все актуальные задания. Задания одной тематики объединяются в колонки. Каждое задание представляет собой карточку, содержащую следующую информацию:

* название задания,
* исполнители задания,
* лейбл состояния задания,
* срок исполнения задания,
* краткое описание задания,
* прикрепленные необходимые материалы.

Лейблы представляют из себя цветную пиктограмму, причем каждый цвет соответствует определенному состоянию задания:

* синий - менеджер представил задание, однако исполнители не приступили к его выполнению;
* оранжевый - задание взято на выполнение или исправление замечаний;
* желтый - задание отправлено на верификацию менеджеру и главному программисту;
* красный - к выполнению есть замечания;
* зеленый - задание выполнено.

Помимо этого, в каждой карточке можно обмениваться сообщениями по поводу выполнения задания.

* 1. Контроль управления версиями

Для контроля управления версиями документации существует репозиторий <https://github.com/SorokinStanislav/UPP>. В нем содержатся все документы, созданные в процессе разработки. В корне репозитория содержатся общие документы:

* оценки участникам за прошедшие недели,
* список всех рецензентов по прошедшим неделям,
* директория с литературой по теме курса,
* readme с правилами пользования git.

Также в репозитории содержатся директории, соответствующие неделям разработки проекта. В каждой директории содержатся документы, созданные или модифицированные в рамках данной недели.

Для контроля управления версиями реализации приложения существует репозиторий <https://github.com/k7-223UPP/upp>, структура которого соответствует паттерну используемого фреймворка Django.

* 1. Организация работы в течение недели

Задания участникам выдаются менеджером, как правило, в четверг или пятницу каждой недели. Для каждого задания обозначается крайний срок его выполнения. После выполнения задания исполнитель ожидает верификации менеджера, а в некоторых случаях еще и главного программиста. Если замечаний к выполнению задания нет, то оно считается выполненным. В противном случае задание отправляется исполнителям на доработку, причем крайний срок выполнения может быть сдвинут на некоторое, определенное менеджером, время. Процесс изменения состояний задания показан на Рисунке 1. Не позднее пяти утра среды каждой недели результаты работы отправляются преподавателю и другой команде разработчиков.

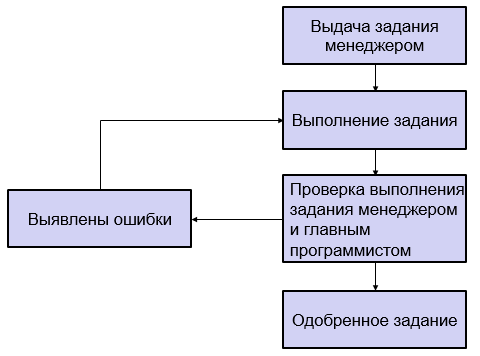


Рис. . Процесс изменения состояний задания

* 1. Анализ результатов завершенного витка

Виток считается завершенным, если составлена спецификация требований с сопутствующими ей документами, а также реализованы все функции, покрывающие требования, указанные в документации.

После этого происходит анализ актуальности разработки, а затем планирование нового витка. В ходе планирования анализируются возможные риски при реализации нового витка.

* 1. Внутрикомандное взаимодействие

Для взаимодействия между участниками проекта используются следующие сервисы:

* Trello,
* Skype,
* Vkontakte,
* командные собрания.