**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 |
| Версия 2.1 | 17.02.16 | Переформулирован пункт 1.1, определение документации и закрепленной роли в пункте 1.3, этап 4) в пункте 3.1, второй абзац пункта 3.2, первый абзац пункта 3.5. Переделана схема в пункте 3.4. Добавлено о назначении участников на роли в пункте 2.1, case-средство 15 в пункте 2.2, сервис TeamViever в пункте 3.6. | Асеев Михаил, Малофеева Анна, Сорокин Станислав |
| Версия 3.0 | 23.02.16 | Создан новый документ | Лазарев Сергей, Петров Анатолий |
| Версия 4.0 | 02.03.16 | Исправлена цель документа в п. 1.1, добавлены термины и определения в п. 1.3, в п. 2.1 роль «менеджер» была переименована в «менеджер по исправлениям», добавлен способ идентификации документов в п. 3.1 добавлены п. 3.3.5 и п. 3.3.6 | Лазарев Сергей |

# Введение

## Цель

Данный документ определяет действия, которые будут выполняться в процессе изучения курса «Тестирование ПО» для достижения целей процесса управления конфигурацией ПО.

## Область применения

Требования и положения данного плана распространяются на все работы, выполняемые в рамках изучения курса «Тестирование ПО», и обязательны для всех участников учебной подгруппы.

## Термины, определения

Таблица 1. Термины и определения

| Термин | Определение |
| --- | --- |
| Базовая версия | Верифицированная версия ЕК, которая в дальнейшем используется участниками проекта при выполнении заданий. |
| Документация | Набор документов, используемых при проектировании и реализации проекта |
| Единица конфигурации | Совокупность данных ЖЦ ПО, которая в целях УК ПО рассматривается как единое целое. |
| Задание | Текстовое описание запроса на изменение, выдаваемое менеджером в СУП Trello. |
| Закрепленная роль | Роль, которая выполняется участником проекта на протяжении всего процесса разработки. |
| Конфигурация | Составное данное ЖЦ ПО, которое может включать в себя любые артефакты, агрегации и конфигурации |
| Роль | Активность, которую участник проекта выполняет в конкретный промежуток времени. |
| Участник | Человек, входящий в команду разработчиков проекта. |
| Чекер | Программа, осуществляющая проверку передаваемых ей данных. |

Таблица 2. Сокращения

| Термин | Определение |
| --- | --- |
| ЕК | Единица конфигурации |
| ЖЦ | Жизненный цикл |
| ПО | Программное обеспечение |
| СУП | Система управления процессами |
| УК | Управление конфигурацией |

# Среда УК ПО

## Организация и ответственности

Все участники процессов ЖЦ ПО вовлечены в процесс УК ПО.

В рамках процесса УК ПО каждый участник проекта может иметь одну или несколько ролей. Роли в процессе УК ПО назначаются на основании ролей участника проекта в других процессах ЖЦ ПО:

* Автор. Обладает полномочиями создания новых и изменение существующих ЕК.
* Менеджер по изменениям. Имеет полномочия на проведение анализа сообщений о проблеме и создание заданий, решающих поставленную проблему.
* Посетитель. Имеет полномочия просматривать данные ЖЦ ПО и вносить свои предложения.

Любой участник проекта имеет возможность в любой момент сообщить о проблеме менеджеру или главному программисту. В случае если участник проекта сообщил о проблеме главному программисту, то последний в свою очередь должен обсудить возникшую проблему с менеджером.

Каждый процесс ЖЦ ПО взаимодействует с процессом УК ПО при любом обращении к данным ЖЦ ПО.

## Взаимодействие УК ПО с другими процессами.

Все процессы разработки ПО взаимодействуют посредством процесса УК ПО, таким образом, все данные, создающиеся или модифицирующиеся в ходе выполнения процессов разработки ПО, являются входными данными для процесса УК ПО, все процессы разработки ПО получают входные данные из процесса УК ПО.

# Мероприятия процесса УК ПО

## Идентификация конфигурации

Цель мероприятия по идентификации конфигурации – однозначно определить каждую единицу конфигурации, включая ее последовательные версии, для того чтобы обеспечить основу для управления единицами конфигурации и возможность ссылаться на них.

Идентификация конфигурации осуществляется посредством использования системы контроля версий Github.

Для контроля идентификации 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 идентификации реализации приложения существует репозиторий <https://github.com/k7-223UPP/upp>, структура которого соответствует архитектуре приложения Django [ссылка].

Также для контроля идентификаций документов каждое изменение заносится в специальную таблицу в начале редактируемого документа, которая содержит в себе следующие поля:

1. «Номер версии» – номер версии верифицированного документа;
2. «Дата верификации» – дата одобрения документа менеджером по изменениям;
3. «Изменения» – наименование изменения и его местонахождение в документе, который был подвергнут редакции;
4. «Автор изменений» – участник проекта, осуществивший редакцию или создание документа.

В случае, когда документ создается в первый раз или пересоздается полностью, в поле «Изменения» записывается следующая фраза «Создан новый документ».

## Базовые версии

Цель введения базовых версий – заложить основу для дальнейших мероприятий ЖЦ ПО, дать возможность ссылаться на эти версии и контролировать их.

Изменение базовой версии ЕК происходит в рамках процесса управления изменениями (п. 3.3. данного плана).

## Регистрация проблем и управление изменениями

В данном разделе рассмотрена общая схема управления изменениями.

### Регистрация проблем.

Любой участник проекта, обнаруживший несоответствие в данных ЖЦ ПО, недостатки результатов процессов, некорректную работу ПО и т.п., обязан сообщить об этом либо менеджеру, либо главному программисту.

### Рассмотрение изменений

После получения сигнала о проблеме данная проблема проходит процедуру рассмотрения, осуществляемая менеджером.

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,
* метка состояния задания,
* срок исполнения задания,
* краткое описание задания,
* прикрепленные необходимые материалы.

Метки представляют из себя цветной знак, причем каждый цвет соответствует определенному состоянию задания:

* синий - менеджер предоставил 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 и документами по поводу выполнения задания.

Для каждой рассматриваемой проблемы определяется следующий набор действий:

1. Если проблема признана актуальной, МИ анализирует, какие изменения нужно произвести, чтобы исправить проблему, создает соответствующее задание (в данном контексте будем называть задание запросом на изменение, или запрос).

В каждом отдельном запросе менеджер указывает данные ЖЦ, подлежащие изменению, и заполняет информацию о предполагаемом изменении. После этого менеджер помечает данный запрос синей меткой.

1. Если проблема признана неактуальной, то менеджер имеет полное право отклонить сообщение о проблеме.

После того, как сообщение о проблеме было помечено синим цветом, соответствующими процессами проводятся мероприятия по внесению изменений с целью устранения проблемы.

### Внесение изменений

Любое изменение единицы конфигурации приводит к изменению ее идентификации, т.е. появлению в системе контроля версий новой версии данной ЕК.

Участник проекта, которому поручено внесение изменений, должен подтвердить получение задание, установив на задании оранжевую метку.

После того, как участник проекта внес все необходимые изменения в рамках данного запроса на изменение, он прикрепляет измененную ЕК к своему заданию и помечает задание желтой меткой. По результатам проверки менеджером принимается решение о дальнейшем изменении ЕК (заданию присваивают красную метку) или об установлении новых базовых версий единиц конфигураций. После того, как все базовые версии установлены, менеджер присваивает заданию зеленую метку.

### Закрытие сообщения о проблеме

Ежедневно менеджер проверяет наличие завершенных заданий. Если у менеджера нет никаких замечаний к заданию, то оно помечается зеленой меткой, в противном случае – красной меткой.

### ЖЦ сообщения о проблеме

Рисунок 1 показывает ЖЦ сообщения о проблеме.

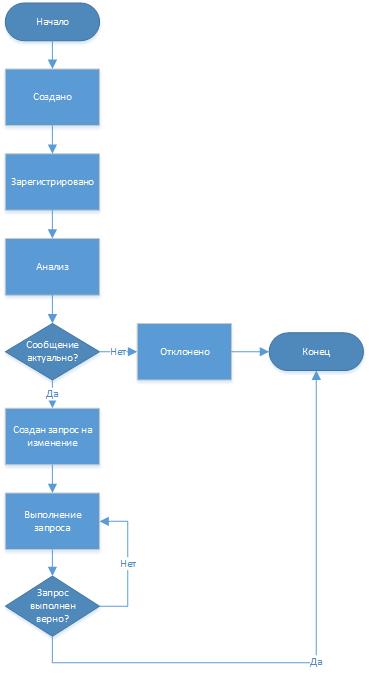


Рисунок 1. ЖЦ сообщения о проблеме

### ЖЦ запроса на изменение

Рисунок 2 показывает ЖЦ запроса на изменение.



Рисунок 2. ЖЦ запроса на изменение

# Критерии перехода

Процесс УК ПО начинается одновременно с процессом планирования создания ПО и заканчивается с выпуском последней версии ПО в рамках учебного курса