

Мероприятия процесса УК ПО

Идентификация конфигурации

Цель мероприятия по идентификации конфигурации – однозначно определить каждую единицу конфигурации (и её последовательные версии) так, чтобы обеспечить основу для управления единицами конфигурации и ссылки на них.

В данном разделе описывается подход к процессу идентификации следующих элементов конфигурации:

- документов;
- сообщений о проблемах;
- запросов на изменения;
- программного кода.

Идентификация документа

Версии документов идентифицируются по их названию. Название имеет составную структуру: имя, отражающее суть документа, нижнее подчеркивание, дата изменения в формате год, месяц, день, час, минута. Итоговый формат названия можно описать в виде следующего шаблона: <Name>_<YYYYMMdd_hhmm>. Пример: План_20220930_1830.

Уникальность идентификации обеспечивается его связью с временем изменения.

Идентификация СП

СП регистрируются в issue репозитория проекта на Github [5].

СП идентифицируются уникальным номером. Первому СП проставляется номер 1, каждому последующему на единицу больше. Данная функциональность обеспечивается сервисом Github[5].

Идентификация запроса на изменения

На базе СП могут быть созданы ЗИ, которые регистрируются в соответствующем репозитории проекта в качестве Pull Request на Github[5].

Идентификация запросов обеспечивается присваиванием уникального номера, аналогично идентификации СП (см. описание инструментария).

Базовая версия и трассируемость

Определения

Базовая версия (Baseline) – Утверждённая зарегистрированная конфигурация одной или более единиц конфигурации, которая в дальнейшем служит основой для последующей разработки и изменяется только в соответствии с процедурами управления изменениями.

Трассируемость (Traceability) – Доказательство связи между определенными единицами, например, между результатами процессов, между результатом и порождающим его процессом, а также между требованием и его реализацией.

Базовые версии

Для каждого элемента конфигурации устанавливается базовая версия, которые должны быть защищены от изменений средствами, описанными в разделе Регистрация проблем и управление изменениями.

Трассируемость

Должна быть обеспечена трассируемость базовой версии/единицы конфигурации к той базовой версии/единице конфигурации, из которой она была получена.

Данное свойство документов обеспечивается стандартом идентификации, а именно временем изменения, зафиксированного в названии.

Трассируемость бизнес-требований, требований к ПО высокого и низкого уровней обеспечивается при помощи способа нумерации требований, описанных в стандарте на разработку требований.

Трассируемость между СП и требованиями к ПО обеспечивается указанием в описании СП номеров требований к ПО.

Трассируемость между ЗИ и СП реализуется средствами Github[5]: возможность просмотра всех ЗИ, относящихся к данному СП.

Трассируемость между ЗИ и изменением в программном коде реализуется средствами Github[5]: возможностью просмотра всех изменений, относящихся к данному ЗИ.

Трассируемость между предыдущей и последующей версиями единицы конфигурации устанавливается с помощью запросов на изменения и истории изменений, сохраняющейся в системе СКУ Github для каждой единицы конфигурации.

Регистрация проблем и управление изменениями

В данном разделе рассмотрена общая схема управления изменениями, реализованная в СКУ Github[5].

Регистрация проблем

Любой участник проекта, обнаруживший несоответствие требований стандарту или описанию проекта, несоответствие программного кода стандартам или требованиям и ненормальную работу ПО и т.п., обязан зафиксировать это несоответствие в виде сообщения о проблеме. При этом он должен внести следующую информацию:

- Заголовок сообщения о проблеме – короткий текст, описывающий суть проблемы;
- Описание проблемы – подробное описание несоответствия с указанием ссылок на соответствующие стандарты или требования, если это возможно;
- Список единиц конфигурации и их версий, в которых проблема была обнаружена.

Если участник проекта не обладает достаточными знаниями для точной и полной идентификации единиц конфигурации и их версий, в которых несоответствие было обнаружено, то список таких ЕК может быть заполнен не полностью.

Рассмотрение изменений

СП должно быть рассмотрено группой управления изменениями, возглавляемой менеджером изменений. В процессе рассмотрения СП принимается решение, нужно ли данную проблему решать. Если проблема признана актуальной, то описание проблемы может быть дополнено МИ.

Далее производится анализ, какие изменение необходимо провести, чтобы решить проблему, и МИ создает нужное количество ЗИ с привязкой к СП. В каждом запросе на изменении МИ указывает данные ЖЦ, подлежащие изменению, и детально заполняет информацию о предполагаемом изменении, так чтобы автор смог понять необходимые изменения без анализа описания сообщения о проблеме. ЗИ передается в работу автору.

СП может потребовать дальнейшего анализа. Такое сообщение ГУИ откладывается на дальнейшее рассмотрение.

Также СП может быть отклонено, если проблема неактуальна или уже решена с объяснением причины отклонения сообщения о проблеме.

Внесение изменений

Любое изменение единицы конфигурации приводит к появлению в системе новой версии данной единицы конфигурации.

Автор рассматривает порученный ему запрос на изменение и приступает к изменению единиц конфигурации. В качестве исходных версий для изменений служат только указанные базовые версии.

После внесения изменений автор должен создать запись о формальной инспекции. По результатам ФИ устанавливаются новые базовые версии ЕК (см План разработки[2] и План верификации[3]). После ФИ проводится тестирование в соответствии с процессом интеграции ПО, описанном в плане разработки ПО.

Могут быть случаи, когда невозможно завершить запрос на изменение. Тогда автор создает новое СП с объяснением причины.

Заккрытие сообщений о проблеме

СКУ Github позволяет автоматически закрывать СП, когда все ЗИ были завершены. Таким образом вмешательство ГУИ не нужно.

Жизненный цикл сообщения о проблеме

Рисунок 1 показывает жизненный цикл сообщения о проблеме.

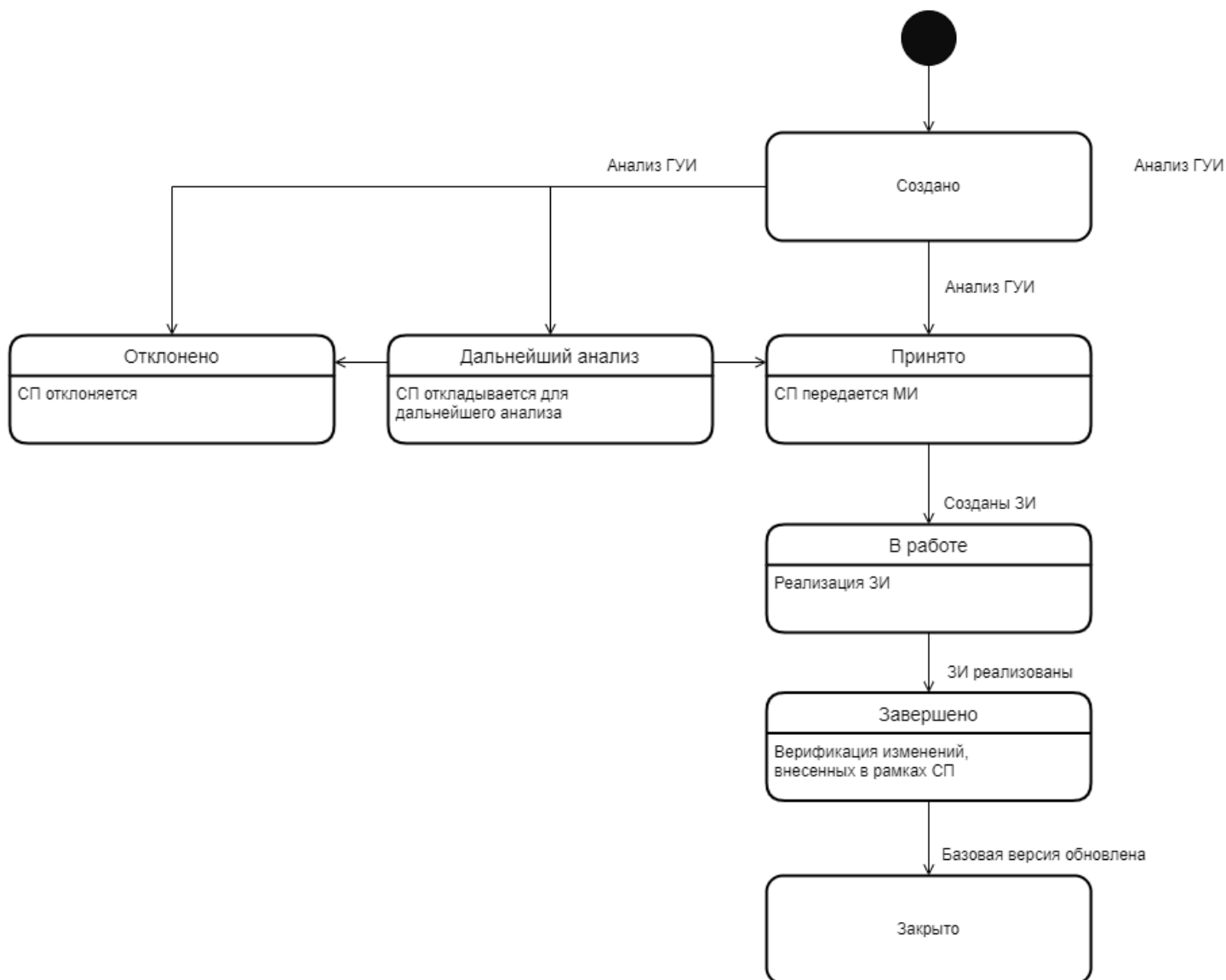


Рисунок 1. Жизненный цикл сообщения о проблеме.

Жизненный цикл запроса на изменение

Рисунок 2 показывает жизненный цикл запроса на изменение.

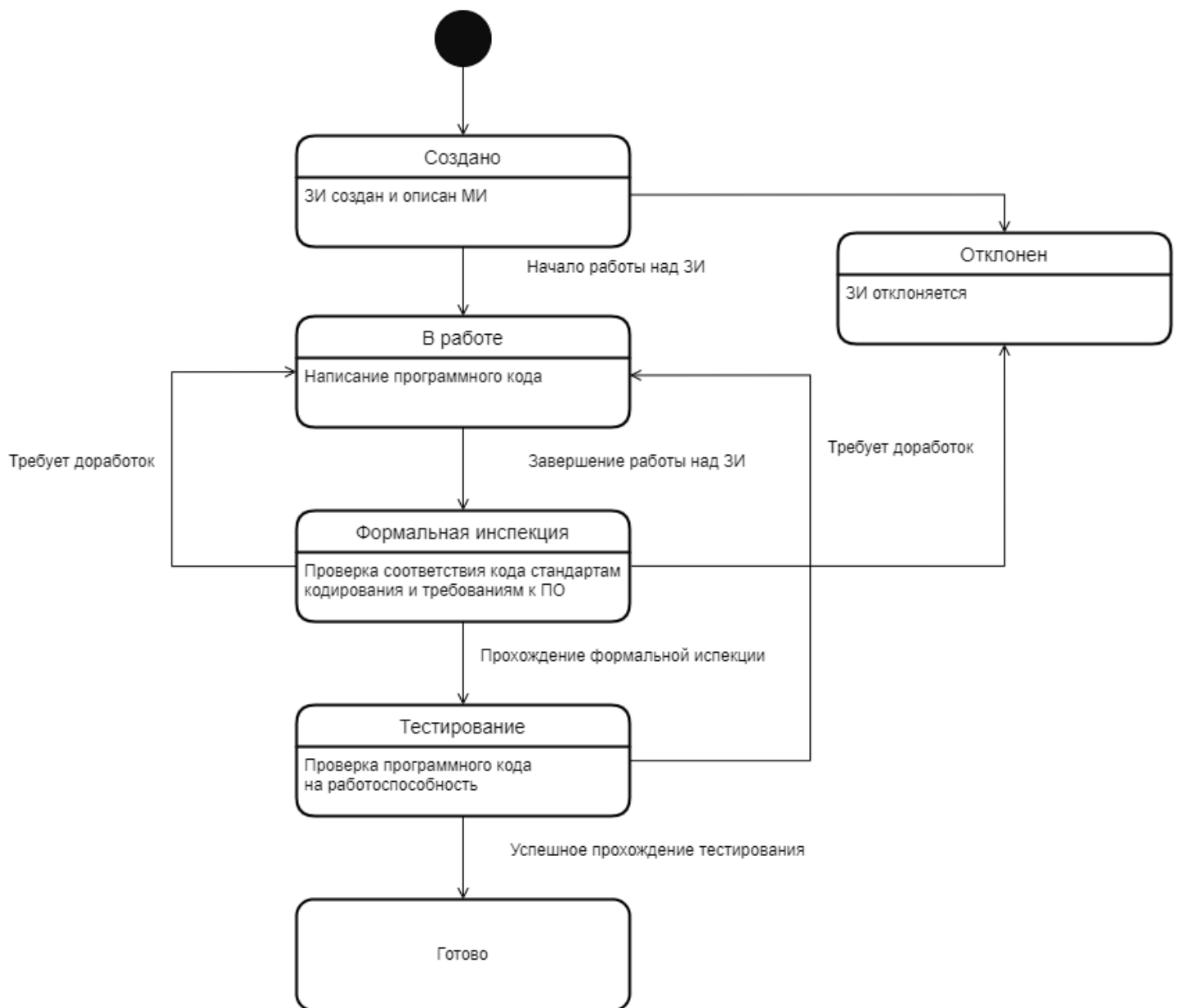


Рисунок 2. Жизненный цикл запроса на изменение

Регистрация проблем и управление изменениями после сертификации

Регистрация проблем и управление изменениями после сертификации в рамках данной работы не рассматривается.

Контроль среды ЖЦ ПО

Контроль планов и стандартов

Управление планами и стандартами проекта осуществляется в соответствии с данным планом. Сведения о действующих версиях планов и стандартов проекта включаются в каталог

комплектации ПО путем задания в нем актуальной базовой версии каталога планов и стандартов.

Контроль инструментальных средств

Данный документ, план верификации ПО, инструкция по тестовому окружению и стандарт на кодирование ПО содержат перечень инструментальных средств для разработки, верификации и интеграции ПО. Управление данными средствами осуществляется в соответствии с УК ПО путем изменения вышеперечисленных документов. Управление этими средствами осуществляется инженером, назначенным руководителем проекта и имеющим роль Автора.