

# 1. Введение

## 1.1. Цель

Данный документ определяет мероприятия, которые будут выполняться в рамках практикума ТПКС для достижения целей процесса управления конфигурацией ПО, определенных в документе КТ-178С [1].

## 1.2. Область применения

Требования и положения настоящего Плана распространяются на все работы, выполняемые в рамках практикума ТПКС, и обязательны для всех участников практикума.

Положения, описанные в данном документе, могут быть неприменимы при выполнении реального проекта и должны быть пересмотрены с учетом требований реального проекта в каждом конкретном случае.

## 1.3. Ссылки

- [1] КТ-178В. Квалификационные требования. Часть 178В. Требования к ПО бортовой аппаратуры и систем при сертификации авиационной техники.
- [2] ТПКС. План разработки ПО.
- [3] ТПКС. План верификации ПО.
- [4] ТПКС. Стандарт на кодирование ПО.
- [5]

## 1.4. Термины определения и соглашения

Таблица 1. Термины, определения и соглашения

Термин	Определение, толкование
Сообщение о проблеме	Документ процесса УК ПО, содержащий описание несоответствия в данных ЖЦ ПО или в описании процессов ЖЦ ПО.
Запрос на изменение	Документ процесса УК ПО, созданный для внесения изменений в данные ЖЦ ПО с целью устранения несоответствия, описанного в СП.
Артефакт	Атомарное данное ЖЦ ПО. Артефактом может быть, например, требование или файл исходного кода
Агрегация	Составное данное ЖЦ ПО, которое включает в себя

Термин	Определение, толкование
	однотипные артефакты и/или агрегации.
Конфигурация	Составное данное ЖЦ ПО, которое может включать в себя любые артефакты, агрегации и конфигурации
Категория запроса на изменение	Атрибут запроса на изменение, идентифицирующий процесс ЖЦ ПО, данные которого будут меняться в рамках этого ЗИ
Формальная инспекция	Способ верификации документов, основанный на экспертной оценке их правильности, выполняемой одним или несколькими инспекторами, как правило, с использованием проверочных перечней.
Единица конфигурации	Совокупность данных ЖЦ ПО, которая в целях УК ПО рассматривается как единое целое.
Тип единицы конфигурации	Тип документов ЖЦ ПО, соответствует разделам 11.1 – 11.20 КТ-178В
ТПКС	Название практикума, моделирующего процесс разработки реальной системы.

Таблица 2. Сокращения

Термин	Определение, толкование
ГК	Гарантия качества
ГУИ	Группа управления изменениями
ЕК	Единица конфигурации
ЖЦ	Жизненный цикл
ЗИ	Запрос на изменение
МИ	Менеджер изменений
ПО	Программное обеспечение
СП	Сообщение о проблеме
УК	Управление конфигурацией
ФИ	Формальная инспекция

## 2. Среда УК ПО

### 2.1. Инструменты УК ПО

В качестве системы конфигурационного управления в практикуме ТПКС используется система контроля версиями Git. Она служит инструментальным средством поддержки процесса УК ПО, обеспечивая хранение данных ЖЦ ПО, идентификацию конфигурации, трассировку, регистрацию проблем, управление изменениями, учет состояния конфигурации и контроль за состоянием среды ЖЦ ПО.

Таблица 3 содержит перечень инструментальных средств, которые будут использоваться для поддержки процесса УК ПО

*Таблица 3. Сокращения*

Инструментальное средство	Предназначение
Git	Система контроля версиями
Github	Веб-сервис для хостинга проекта, использующего git

### 2.2. Взаимодействие УК ПО с другими процессами ЖЦ ПО

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

## 3. Мероприятия процесса УК ПО

### 3.1. Идентификация конфигурации

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

В рамках практикума ТПКС идентификация версии ПО в самом программном продукте рассматриваться не будет.

Идентификация единиц конфигурации осуществляется автоматически системой Git. При помещении в систему каждая ЕК получает идентификатор, уникальный в рамках системы (создание ЕК встроенными средствами системы является частным случаем помещения их в систему). Этот идентификатор сохраняется за ней в течение всего времени ее жизни и не может быть присвоен никакой другой ЕК.

В системе Git единицей конфигурации является *commit* (коммит). Коммит хранит состояние проекта и позволяет с помощью системы Git откатываться к этому состоянию в любой момент времени. По отношению друг к другу коммиты идут один за другим в простейшем случае, однако система также позволяет использовать *ветвление*. В Git ветки называются *branch* и также имеют свои уникальные идентификаторы. Каждая ветка содержит свои коммиты и позволяет переключаться между собой и другими ветками. Ветки можно сливать (в Git эта операция называется *merge*), чтобы применить изменения из одной ветки в другую.

В рамках практикума ТПКС определяются три конфигурации, используемые для организации процессов ЖЦ разрабатываемого ПО: каталог планов и стандартов, каталог среды ЖЦ ПО и каталог комплектации ПО.

Каталог планов и стандартов идентифицирует актуальные планы и стандарты (т.е. содержит ссылки на них).

Каталог среды жизненного цикла (раздел 11.15 КТ-178С [1]) идентифицирует используемые инструменты разработки и верификации, описания их использования и, возможно, другие данные, необходимые для использования инструментов (например, конфигурационные файлы, шаблоны и т.п.).

Каталог комплектации ПО определяет конфигурацию разрабатываемого программного продукта (раздел 11.16 КТ-178С [1]). Каталог комплектации ПО создается в начале работ по практикуму и заполняется по мере появления и обновления данных ЖЦ. В ходе разработки ПО каталог комплектации ПО содержит ссылки на актуальные данные ЖЦ. Указание актуальных планов, стандартов и данных

среды ЖЦ ПО осуществляется путем включения в каталог комплектации ПО соответствующих версий каталога планов и стандартов и каталога среды ЖЦ ПО.

В формате записей в системе представляются сообщения о проблемах, запросы на изменение, результаты верификации (записи ФИ) и протоколы ГК ПО (записи ГК ПО). Поскольку проведение аудита ГК ПО в рамках практикума не предусматривается, записи ГК ПО в рамках практикума ТПКС не создаются.

ПРИЛОЖЕНИЕ А содержит перечень данных ЖЦ ПО.

## 3.2. Базовые версии и трассируемость

Цель введения базовых версий – заложить основу для дальнейших мероприятий жизненного цикла ПО, дать возможность ссылаться на эти версии, контролировать их и обеспечить трассируемость между единицами конфигурации.

Базовая версия агрегации может состоять только из базовых версий агрегаций и артефактов, входящих в нее.

Конфигурация может состоять из данных, отнесенных к обеим категориям контроля. Если в конфигурацию входят данные, отнесенные к категории контроля 1, то в базовую версию конфигурации могут входить только базовые версии таких данных.

### 3.2.1. Каталог планов и стандартов

Установка базовой версии каталога планов и стандартов производится по результатам ФИ, проводимой для проверки согласованности планов и стандартов. Наличие базовых версий отдельных планов или стандартов само по себе не является указанием для разработчиков о том, что именно эти версии являются актуальными и должны использоваться разработчиками ПО (они могут оказаться несогласованными между собой).

### 3.2.2. Каталог среды жизненного цикла ПО

Установка базовой версии каталога среды ЖЦ ПО производится по результатам ФИ, проводимой для проверки исполнения ЗИ, на основании которого выполнено изменение каталога среды ЖЦ ПО. Указание актуальной версии среды ЖЦ ПО производится путем включения в каталог комплектации ПО соответствующей версии каталога среды ЖЦ ПО.

### 3.2.3. Каталог комплектации ПО

Каталог определяет актуальную конфигурацию разрабатываемого ПО и включает ссылки на все данные ЖЦ, используемые при его разработке. Планы, стандарты и инструменты разработки включаются в каталог комплектации ПО путем указания ссылок на соответствующие (актуальные) версии каталога планов и стандартов и каталога среды ЖЦ ПО.

Установка базовой версии каталога комплектации ПО производится по результатам ФИ, проводимой для проверки исполнения ЗИ, на основании которого выполнено изменение каталога комплектации ПО. ФИ, проводимая при установлении базовой версии каталога комплектации ПО, отвечает целям рассмотрения соответствия ПО требованиям (раздел 8.3 КТ-178С [1]).

### 3.2.4. Трассируемость

Трассируемость единиц конфигурации осуществляется встроенными механизмами системы Github.

Трассируемость требований к ПО высокого уровня на системные требования осуществляется с помощью присвоенного порядкового места соответствующим единицам конфигурации.

Трассируемость описания проекта ПО и требований к ПО низкого уровня на требования к ПО высокого уровня, исходного кода ПО на описание проекта ПО и требования к ПО, тестовых примеров и процедур на проект ПО и требования к ПО осуществляется с помощью присвоенного порядкового места соответствующим единицам конфигурации. Порядок ЕК определяется тем, создан ли он первым или после какого ЕК (по порядку).

Система Git также позволяет проводить детализированную трассируемость, то есть трассируемость отдельных частей ЕК. В данной системе отдельные части ЕК сами являются ЕК, поэтому система ведет себя с ними так же, как и со стандартными ЕК.

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

Трассируемость между базовыми версиями единицы конфигурации устанавливается в рамках процесса управления изменениями, описанного в разделе 3.3 данного документа.

Трассируемость между причинами изменений и изменениями устанавливается с помощью связи сообщений о проблемах и запросов на изменения.

Трассируемость единиц конфигурации и процессов, которые их определяют, осуществляется путем выбора соответствующих типов, правильного размещения единиц конфигурации в соответствии с ПРИЛОЖЕНИЕ А и использованием запросов на изменения соответствующих категорий (см. раздел 3.3.2).

### 3.3. Регистрация проблем и управление изменениями

В данном разделе рассмотрена общая схема управления изменениями, реализованная в Github. Практикум ТПКС не полностью охватывает ЖЦ ПО с момента начала разработки и заканчивая сертификацией, поэтому при работе в рамках практикума будут использоваться лишь некоторые положения раздела.

#### 3.3.1. 3.3.1 Регистрация проблем

Любой участник проекта, обнаруживший несоответствие в данных ЖЦ ПО, несоответствие выполняемых процессов процессам, описанным в планах, недостатки результатов процесса, 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;
- Список единиц конфигурации и их версий, в которых несоответствие было обнаружено.

В случае, если предлагается улучшение процесса, в списке единиц конфигурации указываются планы и стандарты, описывающие рассматриваемый процесс.

Любое предполагаемое изменение в данных ЖЦ ПО, в том числе и не связанное с устранением проблем (например, добавление новой функциональности), также обрабатывается через сообщение о проблеме. В этом случае в описание проблемы заносится информация о предлагаемом изменении, а список единиц конфигурации, в которых обнаружено несоответствие, может быть не заполнен.

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.

После того, как вся необходимая информация о проблеме или предлагаемом изменении заполнена, сотрудник переводит сообщение о проблеме в состояние "открыто".

В практикуме ТПКС взаимодействие с системным процессом и изменение требований к системе не предусмотрено. В связи с этим сообщения о проблемах, обнаруженных в системных требованиях, в рамках данного практикума не рассматриваются.

### 3.3.2. 3.3.2 Рассмотрение изменений

После создания любое сообщение о проблеме проходит процедуру рассмотрения группой управления изменениями, возглавляемой менеджером изменений. ГУИ формируется с таким расчетом, чтобы в нее входили сотрудники, хорошо разбирающиеся в предметной области и данных, в которых выявлена проблема, и являющиеся участниками процессов ЖЦ ПО, данные которых могут быть затронуты во время устранения проблемы. В порядке исключения МИ может производить единоличные рассмотрения СП, если указанные в них проблемы достаточно просты и МИ обладает достаточными знаниями для корректной оценки проблемы.

Рассмотрения СП группой управления изменениями происходят с регулярностью, определяемой руководителем проекта. Перед собранием ГУИ менеджер изменений выявляет сообщения о проблемах, находящиеся в состоянии «Открыто» (“Open”), проводит рассмотрение таких проблем.

Для каждого рассматриваемого сообщения о проблеме ГУИ определяет необходимые действия, при этом МИ фиксирует принимаемые решения в системе Github:

- Если проблема признана актуальной, МИ может дополнить описание проблемы или внести дополнительные данные с указанием версий в список единиц конфигурации, в которых обнаружена проблема. После этого ГУИ анализирует, какие изменения нужно произвести, чтобы исправить проблему, и МИ создает необходимое количество запросов на изменения.
- Категория запроса на изменение определяет процесс ЖЦ ПО, выходные данные которого будут изменены для устранения проблемы. Таким образом, при создании запросов на изменения МИ определяет процессы ЖЦ, которые нужно выполнить или повторить для устранения проблемы.
- В каждом отдельном запросе на изменение МИ указывает данные ЖЦ, подлежащие изменению, и заполняет информацию о предполагаемом изменении, при этом детализация должна быть достаточной для того, чтобы автор мог корректно реализовать изменения без анализа сообщения о проблеме.



- После завершения создания связанных ЗИ, перевести сообщение о проблеме в состояние «Закрыто» (“Closed”).
- Перевести сообщение о проблеме в состояние «Закрыто», если проблема признана неактуальной или эта проблема уже решена. МИ заполняет комментарий, в котором описывается причина отклонения сообщения о проблеме, в случае, если решение уже существует, приводится ссылка на него.

### 3.3.3. 3.3.3 Внесение изменений

Любое изменение единицы конфигурации приводит к изменению ее идентификации – появлению в системе новой версии данной единицы конфигурации.

Сотрудник, которому поручено внесение изменений, создает отдельную ветку для работы с конкретной ЕК.

После того, как сотрудник внес все необходимые изменения в рамках данного запроса на изменение. По результатам ФИ устанавливаются новые базовые версии единиц конфигурации (см. План разработки ПО [2] и План верификации ПО [3]). После того, как все новые базовые версии установлены, при закрытии записи о формальной инспекции запрос на изменение переводится в состояние «Закрыто» (“Closed”).

В случае если сотрудник, выполняющий изменения по некоторому ЗИ, обнаруживает невозможность его завершения, он создает СП с указанием причин этого. При появлении новых СП, относящихся к уже изменяемому объекту и/или уже выполняемому ЗИ, МИ может назначить повторное рассмотрение исходного СП (т.е. СП, по которому был создан выполняемый ЗИ). В результате этого рассмотрения выполняемый ЗИ может быть изменен или же могут быть созданы дополнительные ЗИ для данного СП. Такая ситуация возможна, если, например, МИ определил, что проблема содержится в исходном коде, но при изменении исходного кода выяснилось, что нужно также поменять архитектуру ПО. В таком случае МИ создает запрос на описания проекта ПО, работа над исходным кодом будет остановлена, а процессы ЖЦ ПО будут повторены начиная от разработки описания проекта ПО.

В случае, если во время работы над изменением принято решение отказаться от внесения изменения, МИ переводит запрос на изменение из состояния «Открыто» в состояние «Закрыто», занося в комментарий причину перевода.

Данные ЖЦ категории контроля 2 (см ПРИЛОЖЕНИЕ А) могут быть изменены без создания запроса на изменение и проведения формальной инспекции.

### 3.3.4. 3.3.5 Жизненный цикл сообщения о проблеме

Рисунок 1 показывает жизненный цикл сообщения о проблеме.

\*пикча\*

Рисунок 1. Жизненный цикл сообщения о проблеме.

### 3.3.5. 3.3.6 Жизненный цикл запроса на изменение

Рисунок 2 показывает жизненный цикл запроса на изменение.

\*пикча\*

Рисунок 2. Жизненный цикл запроса на изменение

### 3.3.6. 3.3.7 Регистрация проблем и управление изменениями после сертификации

Регистрация проблем и управление изменениями после сертификации в рамках практикума ТПКС не рассматривается.

## 3.4. 3.4 Учет состояния конфигурации

Цель мероприятий учета состояния конфигурации заключается в том, чтобы предоставить данные для управления конфигурацией в процессах жизненного цикла ПО, принимая во внимание идентификацию конфигурации, базовые версии, сообщения о проблемах и управление изменениями.

Учет состояния конфигурации обеспечивается возможностями системы Git. Все действия над единицами конфигурации в этой системе протоколируются. Участники проекта в любой момент времени могут получить следующую информацию:

- Дата и время создания очередной версии единицы конфигурации
- Идентификатор пользователя, создавшего очередную версию единицы конфигурации
- Текущее состояние единицы конфигурации (см. разделы 3.2, 3.3.5, 3.3.6)

## 3.5. 3.5 Архивирование, воспроизведение и выпуск

Архивирование (резервное копирование) данных ЖЦ ПО, хранящихся в системе Git, в рамках практикума ТПКС не предусматривается.

Данные ЖЦ ПО практикума ТПКС, хранящиеся под управлением системы Git, являются оригиналами и должны воспроизводиться исключительно из системы Git.

Мероприятия по выпуску ПО предусматривают создание базовой версии каталога комплектации ПО.

### 3.6. Контроль загрузки ПО

Загрузка ПО на целевой вычислитель блока и контроль загрузки осуществляются на компьютере с помощью Git bash.

Порядок загрузки ПО следующий:

- 1) Инженер извлекает ссылку для клонирования из страницы проекта на Github.
- 2) На целевом вычислителе блока нужно запустить Git bash.
- 3) В Git bash нужно ввести команду *git clone* и в конце вставить извлеченную ранее ссылку.

### 3.7 Контроль среды ЖЦ ПО

#### 3.7.1 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.

#### 3.7.2 Контроль инструментальных средств

Каталог среды ЖЦ ПО содержит перечень инструментальных средств, используемых для разработки, производства, верификации и загрузки ПО. Данные инструментальные средства будут помещены под конфигурационное управление (см. ПРИЛОЖЕНИЕ А) и будут иметь категорию контроля 2. Управление этими средствами осуществляется инженером, назначенным руководителем проекта и имеющим роль Автора. Сведения об используемых версиях инструментальных средств включаются в каталог комплектации ПО путем задания в нем актуальной базовой версии каталога среды ЖЦ.

### 3.8 Контроль данных (документов) ЖЦ ПО

Таблица 4 определяет соответствие процедур УК ПО, реализующих цели процесса УК ПО согласно КТ-178С для данных ЖЦ ПО, относящихся к 1-й и 2-й категориям контроля данных.

Таблица 4. Цели и процедуры процесса УК ПО.

Цель процесса	Ссылка на раздел КТ-178С	Ссылка на раздел данного документа, описывающий соответствующие процедуры
Идентификация конфигурации	7.2.1 a-d	3.1
	7.2.1 e	3.6
Базовые версии	7.2.2 a	3.2.1 - 3.2.4
	7.2.2 b	3.2.4
	7.2.2 c	3.2.1 - 3.2.4. Все данные ЖЦ находятся под контролем системы СПР, в том числе и библиотеки.
	7.2.2 d	3.2.1 - 3.2.4, 3.3.6
	7.2.2 e	3.2.5, 3.3.6
Трассируемость	7.2.2 f, g	3.2.5, 3.3.6
Сообщения о проблемах	7.2.3	3.3.1-3.3.5
Управление изменениями – целостность и идентификация	7.2.4 a, b	Обеспечивается применением системы СПР.
Управление изменениями - отслеживание	7.2.4 c, d, e	3.3.2-3.3.4, 3.3.6
Рассмотрение изменений	7.2.5	3.3.2

Учет состояния конфигурации	7.2.6	3.4
Воспроизведение	7.2.7 a	3.5.3
Защита от несанкционированных изменений	7.2.7 b(1)	3.5
Выбор, обновление, воспроизведение носителей	7.2.7 b(2), (3), (4), c	3.5.4
Выпуск	7.2.7 d	3.5.5
Хранение документов	7.2.7 e	Процедуры управления документами устанавливаются настоящим планом УК ПО.

## 4. Критерии перехода

Процесс УК ПО начинается одновременно с процессом планирования создания ПО и заканчивается с выпуском последней версии ПО в рамках практикума ТПКС.

## 5. Документы УК ПО

Документами процесса УК ПО являются:

- Сообщения о проблемах
- Запросы на изменения
- Каталог комплектации ПО
- Каталог среды ЖЦ ПО

Подробно все документы УК ПО описаны в ПРИЛОЖЕНИЕ А.

## 6. Контроль поставщика

В рамках практикума ТПКС мероприятия по контролю поставщика не рассматриваются.



## ПРИЛОЖЕНИЕ А. Данные ЖЦ ПО

Таблица 5 содержит перечень данных ЖЦ ПО. В столбце «Вид» указан вид и идентификатор ЕК соответствующих данных ЖЦ ПО в СПР Е-178. Атомарные ЕК объединяются в агрегацию верхнего уровня, имя которой приведено в столбце «Место в иерархии», при этом структура хранения может диктовать необходимость использования агрегаций промежуточных уровней. В столбце «Вид» используются следующие сокращения: СП – сообщение о проблеме, ЗИ – запрос на изменение, ФИ – формальная инспекция, Агр – агрегация, К – конфигурация.

Таблица 5. Данные ЖЦ ПО уровня критичности В

Данные ЖЦ ПО	Место в иерархии	Процесс - источник данных	Вид	Категория контроля
План управления конфигурацией ПО	/План УК ПО	Планирование	Агр	1
План разработки ПО	/План разработки	Планирование	Агр	1
План верификации ПО	/План верификации	Планирование	Агр	1
Стандарт на разработку требований к ПО	/Стандарт на разработку требований к ПО	Планирование	Агр	1
Стандарт на проектирование ПО	/Стандарт на проектирование ПО	Планирование	Агр	1
Стандарт на кодирование ПО	/Стандарт на кодирование ПО	Планирование	Агр	1
Каталог комплектации ПО	/Каталог комплектации ПО	УК ПО	К	1

Данные ЖЦ ПО	Место в иерархии	Процесс - источник данных	Вид	Категория контроля
Каталог планов и стандартов	/Каталог планов и стандартов	УК ПО	К	1
Каталог среды ЖЦ ПО	/Каталог среды ЖЦ ПО	УК ПО	К	1
Требования к системе	/Требования к системе	Разработка	Агр	2
Требования к ПО	/Системные требования	Разработка	Агр	1
Инструментальные средства ЖЦ ПО	/Тестовое окружение	Планирование	Агр	2
Описание проекта ПО	/Проект ПО	Разработка	Агр	1
Исходный код	/Код	Разработка	Агр	1
Ключи компиляции, редактирования связей и загрузки	/Исполняемый код/Параметры компиляции, сборки и загрузки	Интеграция	Агр	1
Исполняемый код	/Исполняемый код	Интеграция	Агр	1
Протоколы сборки и загрузки	/Исполняемый код/Протоколы сборки и загрузки	Интеграция	Агр	1

Данные ЖЦ ПО	Место в иерархии	Процесс - источник данных	Вид	Категория контроля
Тестовые примеры	/Тесты	Верификация	Агр	1
Результаты испытаний	/Результаты тестов	Верификация	Агр	2
Отчеты об анализе структурного покрытия	/Результаты тестов /Структурное покрытие	Верификация	Агр	2
Сообщения о проблемах	All PRs	УК ПО	СП	2
Запросы на изменения	All CRs	УК ПО	ЗИ	2
Записи о ФИ	All FIs	Верификация	ФИ	2