Стандарт разработки СПО

Оглавление

[1. Назначение стандарта СПО 2](#_Toc438568928)

[2. Используемые роли, термины и определения 2](#_Toc438568929)

[2.1 Роли 2](#_Toc438568930)

[2.2 Термины и определения 2](#_Toc438568931)

[3. Общие принципы и правила разработки 2](#_Toc438568932)

[4. Модель времени сборки (МВС). Описание документа. 3](#_Toc438568933)

[5. Модель Времени Развертывания (МВР). Описание документа. 3](#_Toc438568934)

[6. Модель Программной Архитектуры (МПА). Описание документа. 3](#_Toc438568935)

[7. Скрипт сборки проекта. Описание скрипта. 4](#_Toc438568936)

[8. Политика версионности ПО 4](#_Toc438568937)

[8.1 Первая цифра номера версии (X) 4](#_Toc438568938)

[8.2 Вторая цифра номера версии (Y) 4](#_Toc438568939)

[8.3 Третья цифра номера версии (Z) 4](#_Toc438568940)

[9. Общая схема процесса разработки ПО 6](#_Toc438568941)

[10. Описание процесса разработки ПО 7](#_Toc438568942)

[11. Правила работы с СУВ 7](#_Toc438568943)

[11.1 Определения 7](#_Toc438568944)

[11.2 Структура части репозитория для исходного кода 7](#_Toc438568945)

[11.3 Правила формирования имен 7](#_Toc438568946)

[12. Структура репозитория проекта в СУВ 7](#_Toc438568947)

[Описание структуры 7](#_Toc438568948)

[13. Правила работы с системой баг/таск трекинга 8](#_Toc438568949)

[Схема работы с системой баг/таск трекинга 8](#_Toc438568950)

[Правила создания задач 8](#_Toc438568951)

# Назначение стандарта СПО

Назначение стандарта разработки СПО (далее – стандарт СПО) на различных платформах:

* унификация разработки, тестирования и документирования СПО на разных технологических платформах;
* стандартизация разработки, тестирования и документирования СПО на разных технологических платформах;
* снижение издержек на разработку, тестирование и документирование СПО;
* прогнозирование и управление качеством результата разработки СПО;
* формирование общих правил по управлению версиями СПО;
* прозрачность состояние проекта в любой момент времени.

# Используемые роли, термины и определения

## Роли

## Термины и определения

**СПО.** Специализированное программное обеспечение, разрабатываемое организацией в рамках своей деятельности.

# Общие принципы и правила разработки

1. В проекте обязательно использование системы управления версиями (далее - СУВ) исходного кода. Система управления версиями исходного кода стандартизуется на уровне предприятия, одна для всех проектов.
2. Обязательные регулярные «коммиты» исходного кода и других проектных материалов в СУВ.
3. В проекте обязательно использование системы баг/таск трекинга (далее – Трекер). Система баг/таск трекинга стандартизуется на уровне предприятия, одна для всех проектов.
4. Вся создаваемая в проекте техническая документация создается в формате «MS Word 97/2003».
5. Для каждого проекта на самом начальном этапе разработки разработчиками создаются следующие документы/скрипты:

* модель времени сборки (далее – МВС);
* модель времени развертывания (далее – МВР);
* модель программной архитектуры (далее – МПА);
* сборочный скрипт проекта.

1. Каждый проект должен быть встроен в «конвейер» разработки ПО – в систему «Непрерывной интеграции и поставки ПО» (далее – НИПО). Для встраивания проекта в систему НИПО используются документы: МВС, МВР, МПА, сборочный скрипт.
2. Артефакты последней успешной сборки проекта должны быть общедоступны.

# Модель времени сборки (МВС). Описание документа.

Назначение документа: описание условий и процесса сборки проекта. Данный документ создается на начальном этапе проекта и дополняется/изменяется разработчиками (обычно – старшим разработчиком в команде) параллельно с разработкой ПО – любые изменения ПО, приводящие к изменениям способа/порядка/условий сборки должны быть своевременно отражены в данном документе. Документ должен быть создан в формате «MS Word 97/2003» и содержать:

* описание среды сборки:
* «железо» (при необходимости)
* ОС (версия, разрядность и т.п.)
* описание ПО, необходимого для сборки (список пакетов/модулей)
* при необходимости (наличие спец. требований) – описание процесса сборки вспомогательного ПО, необходимого для сборки текущего проекта
* описание настройки среды сборки проекта
* описание (последовательность действий) процесса сборки проекта до необходимого конечного артефакта (DEB/RPM пакета, исполняемого модуля, другое)

По согласованию со старшим разработчиком допускается автоматизировать процессы дополнения и изменения данного документа с помощью специализированных средств.

# Модель Времени Развертывания (МВР). Описание документа.

Данный документ создается на самом начальном этапе проекта и дополняется/изменяется разработчиками (обычно – старшим разработчиком в команде) параллельно с разработкой ПО – любые изменения ПО, приводящие к изменениям способа/порядка/условий развертывания должны быть своевременно отражены в данном документе. Документ должен быть создан в формате «MS Word 97/2003» и содержать:

* описание среды развертывания (для каждой среды):
* «железо» (при необходимости)
* ОС (версия, разрядность и т.п.)
* описание ПО, необходимого для развертывания (список пакетов/модулей)
* при необходимости (наличие спец. требований) – описание процесса развертывания вспомогательного ПО, необходимого для развертывания текущего проекта
* описание настройки среды развертывания (для каждой среды)
* описание (последовательность действий) процесса развертывания проекта до необходимого состояния

По согласованию со старшим разработчиком допускается автоматизировать процессы дополнения и изменения данного документа с помощью специализированных средств.

# Модель Программной Архитектуры (МПА). Описание документа.

Данный документ создается на самом начальном этапе проекта и дополняется/изменяется разработчиками (обычно – старшим разработчиком в команде) параллельно с разработкой ПО – любые изменения ПО, приводящие к изменениям архитектуры должны быть своевременно отражены в данном документе. Документ должен быть создан в формате «MS Word 97/2003» и содержать:

* визуальную схему программных модулей проекта. Данная схема должна коррелировать с логической (компонентной) архитектурой проекта, требованиями к ПО, ТЗ на изготовление данного ПО. Уровень детализации схемы согласовывается старшим разработчиком с руководителем отдела разработки или другими заинтересованными в проекте лицами (по указанию руководителя отдела). На схеме должны присутствовать:
* наименование каждого модуля
* связи между модулями с указанием направлений
* при необходимости – дополнительные пояснения к связям/модулям
* краткое описание назначения и свойств каждого программного модуля проекта
* при необходимости – дополнительные пояснения к составу/взаимодействию модулей

Исходный макет визуальной схемы разрабатывается с помощью любого визуального редактора и сохраняется рядом с документом МПА.

По согласованию со старшим разработчиком допускается автоматизировать процессы дополнения и изменения данного документа с помощью специализированных средств.

# Скрипт сборки проекта. Описание скрипта.

(в работе: правила составления скрипта сборки)

# Политика версионности ПО

Для определения значимости стадии разработки программного обеспечения и быстрого определения качества произведенных изменений, во всех проектах ведется трехуровневая последовательная нумерация версий в виде X.Y.Z

## Первая цифра номера версии (X)

Первая цифра – X – главный номер версии, используется для указания на значительные изменения в ходе разработки проекта. Примерами таких изменений могут служить:

* Поддержка новой платформы: аппаратного обеспечения, операционной системы, набора библиотек (framework);
* Расширение круга решаемых задач, например, ввод системы поддержки плагинов;
* Кардинальное изменение интерфейса взаимодействия с пользователем;
* Принятие решения о передаче результатов разработки заказчику.

## Вторая цифра номера версии (Y)

Вторая цифра – Y – вспомогательный или младший номер версии, используется для указания на изменение состава функциональных возможностей существующих задач, решаемых программным обеспечением. Примерами таких изменений могут служить:

* Добавление поддержки нового формата плагинов;
* Изменение формата конфигурационных файлов.

## Третья цифра номера версии (Z)

Третья цифра – Z – номер сборки, используется для указания на проводимые работы по исправлению обнаруженных недочетов и ошибок в ходе тестирования и текущей работы.

# Общая схема процесса разработки ПО

kgkljdhfgkjh

VCS

WIKI

Bug Tracker

CI

DE/IDE

STORAGE

Task Tracker

VM

Test Environment

VCS

VCS

VM

VM

VM

VM

VM

VM

# Описание процесса разработки ПО

(в работе: описание процесса разработки)

# Правила работы с СУВ

## Определения

(в работе)

## Структура части репозитория для исходного кода

(в работе)

## Правила формирования имен

(в работе)

(в работе: правила/приемы работы с СУВ)

# Структура репозитория проекта в СУВ

В каждом проекте используется СУВ, в которой проект находится в специализированном месте – репозитории. Репозиторий проекта – по сути это раздел/каталог для данного проекта в СУВ. Этот раздел должен иметь четко определенную структуру каталогов/подразделов, обязательную для всех проектов.

## Описание структуры

* **[implementation]** – каталог верхнего уровня, в котором содержится непосредственно реализация программного проекта и сопутствующие материалы. В данном каталоге содержатся следующие подкаталоги:
* **[database]** – sql скрипты для работы с БД проекта (создание, обслуживание и т.п.)
* **[documents]** – программистская документация (тех.)
* **[scripts]** – скрипты сборки и т.п.
* **[sources]** – непосредственно исходные тексты программ. В данном каталоге содержатся следующие подкаталоги:
* **[trunk]** – транк или ствол проекта. Основная ветка разработки. Из данной ветки происходят сборки системой CI.
* **[tags]** – помеченные (tags) ветки, обычно релизы или версии ПО.
* **[branches]** – ветки для разработки ПО.
* **[utils]** – различные вспомогательные утилиты для данного проекта. Это могут быть как скрипты, так и исполняемые файлы.
* **[management]** – для ПМ
* **[production]** – продакшн ???
* **[requirements]** – требования ТЗ, постановки и т.п.
* **[standards]** – используемые стандарты (RFC и т.д.)
* **[tests]** – данный каталог предназначен для хранения любых данных, относящихся к тестированию проекта. Для упорядочения и группировки данных по типу и назначению в каталоге содержатся следующие подкаталоги:
* **[cases]** – use-cases
* **[data]** – данные для тестирования
* **[plan]** – тест планы
* **[results]** – результаты тестов
* **[tools]** – опциональный каталог для инструментария проекта (виртуальные машины, образы и т.п.). *Данный каталог не заполняется напрямую, а подключается как external link из другого репозитория!*

# Правила работы с системой баг/таск трекинга

(в работе: как именно работать с данной системой)

## Схема работы с системой баг/таск трекинга

(в работе: визуальная схема)

## Правила создания задач

(в работе: набор обязательных правил)

Обязательно указывать для задачи значения «Дата начала» и «Дата завершения». Если дата завершения точно не известна – можно указать ожидаемую – «не позднее ХХ».