

План управления конфигурацией

Идентификация конфигураций

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 сообщения о проблеме

Сообщения о проблемах регистрируются в issue соответствующего репозитория проекта на Github (см. описание инструментария).

Сообщения о проблемах идентифицируются уникальным номером, который работает по инкрементному принципу, его функционирование обеспечивается сервисом Github (см. описание инструментария).

Идентификация запроса на изменения

Запросы на применение изменений регистрируются в соответствующем репозитории проекта на Github (см. описание инструментария).

Идентификация запросов обеспечивается присваиванием уникального номера, который работает по инкрементному принципу, его функционирование обеспечивается сервисом Github (см. описание инструментария).

Идентификация программного кода

Программный код идентифицируется при помощи возможности проставлять теги (tags), реализованной в системе Github (см. описание инструментария). Теги для базовых версий и производных от них

проставляются путем использования github-actions после успешной сборки версии проекта.

Уникальность обеспечивается инкрементной логикой работы тегов.

Базовые версии 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 “управлении 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.

Свойство трассируемости кода обеспечивается путем использования системы тегов версий и средствами платформы Github (см. описание инструментария).

Регистрация и отслеживание проблем и корректирующие действия

Основные понятия

Целью сообщений о проблемах, отслеживания и корректирующих действий является: * зарегистрировать несоответствие процесса планам создания ПО и стандартам, * зарегистрировать недостатки результатов процессов жизненного цикла ПО, * зарегистрировать

несоответствующую требованиям работу программных продуктов и гарантировать разрешение этих проблем.

Состав сообщения о проблеме

Сообщение о проблеме реализуется средствами issues в системе Github (см. описание инструментария) и включает в себя: * идентификацию элемента конфигурации, в котором обнаружена проблема; * идентификацию единицы конфигурации, которую требуется изменить; * полное описание сути проблемы и идентификацию документа, требования которого нарушаются; * предполагаемое решение (по-возможности).

Типы сообщений о проблеме

Сообщение о проблеме может иметь различные типы, которые отражаются в labels в системе Github (см. описание инструментария). Существуют следующие типы сообщений:

- type: feature – запрос на реализацию нового функционала;
- type: refactor – запрос на изменение существующих элементов конфигурации, без изменения их функциональности, но улучшения удобства взаимодействия с ними;
- type: fix – запрос на устранение несоответствий требованиям, которые могли попасть в базовую версию, в следствие ненадлежащей верификации или иных причин.

Состояние сообщения о проблеме

Сообщение о проблеме может иметь различные состояние, которые отражаются в labels в системе Github (см. описание инструментария). Существуют следующие состояния сообщений: * status: confirm needed – сообщение о проблеме, ожидающее рассмотрения; * status: confirmed – сообщение о проблеме было рассмотрено и ожидает назначение ответственных лиц; * status: rejected – сообщение о проблеме было отклонено из-за неверности или невозможности реализации решения; * status: in progress – сообщение о проблеме, над которой ведется работа; * status: review needed – сообщение о проблеме, требующее рассмотрения; * status: completed – сообщение о проблеме, работа над которой завершена.

Назначение ответственного

После рассмотрения сообщения о проблеме и принятия решения о необходимости и возможности произвести соответствующие изменения, требуется при помощи средств Github назначить ответственные лица

– Assignees (см. описание инструментария), задача которых будет состоять в реализации изменений, решающих данную проблему.

Управление изменениями

Цель мероприятий выполняемых в рамках управления изменениями – обеспечить регистрацию, оценку, принятие решений и утверждение изменений в течение жизненного цикла ПО.

Защита от изменений

При появлении новой базовой версии/единицы конфигурации данная версия/единица конфигурации блокируется для изменения и остается доступна только для чтения.

Документы Доступ только на чтение осуществляется средствами Github.

Код Код может быть изменен только с помощью запросов на изменения. Защита от изменения базовой версии достигается средствами Github, которые запрещают прямого изменения базовой версии.

Сообщения о проблемах Сообщения о проблемах являются неизменяемыми, в следствие чего, их идентификация не меняется. Неизменность сообщений об ошибках достигается средствами Github.

Запросы на изменения Запросы на изменения являются неизменяемыми, в следствие чего, их идентификация не меняется. Неизменность сообщений об ошибках достигается средствами Github.

Изменение идентификации

Управление изменениями должно обеспечить гарантию того, что любое изменение в единице конфигурации повлечет изменение ее идентификации.

Документы Для документов изменение идентификации происходит в соответствии со стандартом идентификации - в названии файла изменяется только дата публикации документа.

Код Для документов изменение идентификации происходит в соответствии со стандартом идентификации - при изменении базовой версии проставляется идентификатор базовой версии.

Сообщения о проблемах Сообщения о проблемах являются неизменяемыми (см. сообщение об ошибке), вследствие чего, идентификация не меняется.

Запросы на изменения Запросы на изменения являются неизменяемыми (см. сообщение об ошибке), вследствие чего, идентификация не меняется.

Источник изменения

Любое изменение может начаться только после появления утвержденного сообщения о проблеме; все изменения должны трассироваться до их источников, то есть сохранять ссылку на сообщение о проблеме.

Идентификация изменения

Документы Идентификация изменений документов происходит за счет средства diff системы управления версиями git. Данное средство предоставляют как Github, так и средство командной строки git.

Код Идентификация изменений кода происходит за счет средства diff системы управления версиями git. Данное средство предоставляют как Github, так и средство командной строки git.

Сообщения о проблемах Сообщения о проблемах являются неизменяемыми (см. сообщение об ошибке), вследствие чего, идентификация изменений не требуется.

Запросы на изменения Запросы на изменения являются неизменяемыми (см. сообщение об ошибке), вследствие чего, идентификация изменений не требуется.

Процессы ЖЦ ПО

Процессы ЖЦ ПО должны повторяться, начиная с той точки, в которой изменение начинает влиять на выходные данные этих процессов.

Рассмотрение изменений

Цель мероприятия рассмотрение изменений – гарантировать, что проблемы и изменения оценены, утверждены или отклонены, а утверждённые изменения реализованы, а также обеспечена обратная связь с затронутыми процессами через сообщения о проблемах и управление изменениями, установленными в процессе планирования создания ПО.

Регламент

1. Для рассмотрений изменений создается Pull Request в Github.
2. Название должно содержать краткое описание изменений.
3. В описании должна быть ссылка на сообщение об ошибке, которое релизуется с помощью `#issue-number`.
4. Добавить в `reviewers` человека, который не причастен к этим изменениям.
5. Reviewers обязаны в течение дня просмотреть изменения и дать вердикт к изменениям: либо принять их, либо отклонить.
6. В случае отклонения необходимо провести правки и начать рассмотрение изменений заново.
7. В случае принятия предлагаемых изменений производится подтверждение того, что затронутые в процессе рассмотрения единицы конфигурации верно идентифицированы.