

Программное обеспечение

«Система мониторинга транспортных сетей передачи данных "Topolograph"»

Описание процессов жизненного цикла программного обеспечения

Оглавление

1. Введение	3
1.1. Назначение документа	3
1.2. Модель жизненного цикла	3
1.3. Основные этапы жизненного цикла	3
2. Организация процессов разработки и контроля версий	3
2.1. Модель ветвления (Git Flow)	3
2.2. Процесс разработки	4
3. Процессы тестирования и обеспечения качества	4
3.1. Постоянная модель тестирования	4
3.2. Правила приемки изменений	4
4. Процесс выпуска релизов и развертывания	5
4.1. Планирование релиза	5
4.2. Развертывание в промышленную среду	5
5. Процессы эксплуатации и поддержки	5
5.1. Поддержка	5
5.2. Управление инцидентами и проблемами	6
6. Процесс вывода из эксплуатации	6

1. Введение

1.1. Назначение документа

Настоящий документ описывает полный жизненный цикл программного обеспечения «Topolograph» - от зарождения идеи до вывода из эксплуатации. Документ определяет методологии, процессы, ответственных участников, а также технические и организационные процедуры, обеспечивающие управление качеством, версиями, изменениями и поддержкой ПО.

1.2. Модель жизненного цикла

Для разработки и сопровождения ПО «Topolograph» применяется гибридная модель, сочетающая:

- Итеративно-инкрементальный подход для планирования крупных релизов.
- Элементы Agile/Scrum для оперативной разработки и исправления инцидентов.

1.3. Основные этапы жизненного цикла

1. Анализ и планирование.
2. Проектирование и разработка.
3. Тестирование и верификация.
4. Внедрение и передача в эксплуатацию.
5. Эксплуатация и техническая поддержка.
6. Модернизация и развитие.
7. Вывод из эксплуатации.

2. Организация процессов разработки и контроля версий

2.1. Модель ветвления (Git Flow)

Управление исходным кодом ведется в системе контроля версий Git с использованием адаптированной модели Git Flow.

- Ветки:
 - main (ранее master): Содержит только стабильный, проверенный код, соответствующий коду в промышленной эксплуатации. Каждый коммит в main - это новая версия (тег).
 - dev: Основная ветка для интеграции новой функциональности. Здесь ведется текущая разработка.
 - release/*: Ветки для подготовки релиза. Создаются из dev для финального тестирования, исправления багов. После готовности сливаются в main и dev.

- hotfix/*: Ветки для срочных исправлений в промышленной версии. Создаются от main, после исправления сливаются в main и dev.
- Версионирование: Применяется семантическое версионирование (SemVer) MAJOR.MINOR.PATCH:
 - MAJOR: Принципиальные изменения, ломающие обратную совместимость API.
 - MINOR: Добавление новой функциональности, новых сервисов, с сохранением обратной совместимости.
 - PATCH: Исправления ошибок, не влияющие на API.

2.2. Процесс разработки

1. Создание задачи: на основе совещательной активности в команде формулируется задача, выраженная в реализации конкретной функциональности в зоне ответственности участника. Задача получает сроки выполнения и принимается в работу немедленно.
2. Создание ветки: Разработчик по своему усмотрению создает ветку или создаёт коммиты непосредственно в ветке dev.
3. Разработка: Внесение изменений в проекты микросервисов. Коммиты в ветку dev с понятными сообщениями. Разработчик самостоятельно выполняет сборку и тестирование микросервиса или модуля в зоне своей ответственности. Он принимает решение о готовности микросервиса.
4. Слияние: Разработчик после принятия решения о готовности микросервиса в новой функциональности самостоятельно сливает ветку dev своих проектов в ветку release.
5. Периодически по совещательному решению команды выполняется слияние ветки release с main с инкрементом версии

3. Процессы тестирования и обеспечения качества

3.1. Постоянная модель тестирования

Тестовая зона продукта Topolograph размещена в открытом доступе на домене test.topolograph.com. К тестовой зоне подключены пользователи-волонтеры (организации и физические лица), которые заинтересованы в использовании комплекса Topolograph в его нестабильной версии. В ходе такого использования фиксируются ошибки и неисправности, которые передаются разработчикам и исправляются в процессе нормальной разработки.

3.2. Правила приемки изменений

Изменение считается готовым к включению в релиз, когда:

- Разработчик завершил работу над задачей, выполнил самостоятельное тестирование;
- Отсутствуют текущие запросы на исправление от пользователей-волонтеров;
- Обновлено документация (комментарии в коде, при необходимости — руководства);

4. Процесс выпуска релизов и развертывания

4.1. Планирование релиза

1. Формирование бэклога релиза: На основе дорожной карты (Roadmap) и приоритетных задач из dev.
2. Создание ветки релиза: Ответвление release/vX.Y.Z от dev.
3. Стабилизация: В ветке релиза запрещается добавление новой функциональности. Допускаются только исправления багов, найденных в ходе тестирования релиза.
4. Сборка дистрибутива: Из ветки релиза запуском инструментов toolchain формирует итоговый дистрибутив:
 - Выгрузка Docker-образов финальной версии микросервисов в файлы img.
 - Сборка архива (Topolograph_vX.Y.Z.zip) с Docker-образами, файлом оркестрации и обновленной документацией.
 - Простановка тега (tag) vX.Y.Z в репозитории Git.

4.2. Развертывание в промышленную среду

Процесс регламентирован «Руководством по установке» и проводится в соответствии с договорными отношениями с заказчиками (пользователями) в следующем порядке:

1. Уведомление: Информирование всех заинтересованных сторон о времени и объеме работ.
2. Резервное копирование: Создание полной резервной копии баз данных и файлов конфигурации.
3. Остановка сервисов: Временная остановка микросервисов Topolograph.
4. Применение обновления: Загрузка новых образов и файла оркестрации.
5. Первоначальный запуск и проверка: Запуск системы, выполнение smoke-тестов.
6. Мониторинг: Усиленный мониторинг системы в первые часы и дни после обновления.
7. Откат (Rollback Plan): Четкий план отката к предыдущей версии в случае критических проблем.

5. Процессы эксплуатации и поддержки

5.1. Поддержка

Техподдержка: Прием запросов от пользователей, первичный анализ, решение типовых проблем. Анализ и устранение ошибок в коде, доработка функциональности, консультации по сложным интеграционным сценариям.

5.2. Управление инцидентами и проблемами

- Инцидент: Любое событие, которое привело или может привести к прерыванию или снижению качества сервиса или прекращению потока входных данных (например, утрата соседства IGP).
- Проблема: Неизвестная первопричина одного или нескольких инцидентов.
- Процесс: Регистрация -> Классификация -> Приоритизация -> Диагностика -> Решение -> Закрытие. Корректирующие действия фиксируются.

6. Процесс вывода из эксплуатации

Вывод Topolograph из эксплуатации инициируется в случае:

- Перехода на новую, принципиально иную платформу сетевого мониторинга.
- Прекращения компанией деятельности в рамках поддержки и развития комплекса.
- Прекращения договорных отношений с заказчиками.

Этапы вывода:

1. Принятие решения и планирование: Согласование с заказчиком (пользователем) архивацию данных.
2. Архивация данных: При необходимости архивирование на долгосрочное хранение.
3. Остановка сервисов: Поэтапное отключение веб-сервисов, регламентных заданий.
4. Демонтаж инфраструктуры: Освобождение серверных ресурсов.
5. Фиксация факта вывода: Подписание акта о выводе системы из эксплуатации.