

ООО «Паллада»

**«Модуль организации проведения поисково-спасательных операций
(работ) главного координационного центра поиска и спасания»**

Документация, содержащая описание технических средств хранения исходного текста и объектного кода программного обеспечения, а также технических средств компиляции исходного текста в объектный код ПО

Москва
2023

Оглавление

Сведения о хранении и компиляции.....	#
Хранение исходного кода.....	#
Компиляция исходного кода.....	#

Сведения о хранении и компиляции

Хранение исходного кода

Для работы над проектом используется система контроля версий Mercurial (Hg). Команда разработчиков взаимодействует с консольным

или браузерным инструментом для выгрузки кода на сервер и изменения структуры.

Mercurial — легковесная распределенная система контроля версий, хранящая данные в рабочих копиях и в централизованных репозиториях. Это позволяет организовать автономную работу разработчиков с последующей интеграцией полученных результатов через централизованный репозиторий. Для работы с исходными кодами развернут единый сервер хранения и веб-интерфейс.

Особенности ведения исходных кодов

- 1. Исходные коды всех проектов хранятся в одном месте.** Для каждого крупного проекта заводится отдельный репозиторий, где расположены его исходные коды. Непосредственная разработка ведется локально в личной рабочей копии каждого разработчика, затем отправляется в централизованное хранилище.
- 2. Mainline development.** Весь актуальный код хранится в главной "ветке", которая называется "default". Все изменения делаются только в ответвлениях от основной ветки, которые затем отправляются в центральное хранилище, где происходит процесс интеграции в основную ветвь.
- 3. Зеленый транк (green trunk).** Любые изменения перед интеграцией в основную ветку проходят одобрение уполномоченными сотрудниками, а также выполняются интеграционные тесты. Изменения, для которых тесты не проходят, либо не получено одобрение, не добавляются в основную ветку. Добавление изменений в основную ветку организовано через пулл-реквесты (pull-request).
- 4. Герметичность.** Любой из проектов, хранящихся в едином репозитории, собирается только с использованием исходных кодов из этого репозитория. Исходные коды внешних библиотек и описание процесса сборки хранятся в этом же репозитории.

Уровни использования единого репозитория

В едином репозитории предусмотрено два уровня использования **tier** (англ. tier - уровень):

- **Tier 0** — проекты, полностью интегрированные со всеми рекомендуемыми инструментами и технологиями разработки в едином

репозитории. Сюда относится большинство проектов на C++, Java, Python и Go.

- **Tier 1** — проекты, исходные коды которых хранятся в едином репозитории, но при этом используют собственные инструменты тестирования и сборки. В первую очередь сюда относятся проекты мобильной и фронтенд-разработки.

ПО «Модуль организации проведения поисково-спасательных операций (работ) главного координационного центра поиска и спасания» относится к tier 1.

Компиляция исходного кода

Для компиляции сервиса из исходного кода используются внутренние сервисы для сборки, тестирования и публикации готового ПО.

В основе процесса сборки лежит система Gradle, которая выполняет задачи по сборке ПО, дистрибутивов, развертыванию среды разработки, подготовки тестовых данных, проведению интеграционного тестирования. Выполнение цепочек задач осуществляется при помощи Jenkins, результаты сборки публикуются на серверах Nexus и WebDAV.

Система сборки позволяет собирать и тестировать код на основных языках: C#, Java и TypeScript. Сборка может выполняться локально с локальным кэшированием, локально с удалённым кэшированием, а также на кластере распределённой сборки. Система сборки дополняется рядом плагинов собственной разработки для получения полного цикла сборки для всех используемых в проекте языков и технологий.

Система сборки обладает следующими свойствами:

- **Полностью статическая.** Все зависимости анализируются заранее и изменения фиксируются в графе команд. На основе анализа каждая команда фиксирует её результат на данном состоянии входных данных и зависимостей. Это позволяет кэшировать результаты, а также используется при анализе изменений для исключения команды из исполнения.
- **Универсальная и высокоуровневая.** Описание системы сборки делается на уровне модулей, макросов и зависимостей между модулями.
- **Декларативная.** В описании сборки большая часть конструкций фиксирует свойства модулей и команд и связи между ними. При этом часть

конструкций выполняется последовательно: установка и вычисление локальных переменных, условные конструкции — порядок написан в gradle-файле.

Все инструменты, используемые в сборке зафиксированы в бинарном виде или строятся из исходного кода в рамках сборки. Таким образом сборка происходит без использования внешних источников.

Развертывание и публикация

Сборка проектов позволяет получать дистрибутивы готового ПО в двух вариантах: установочные пакеты, образы контейнеров. Контейнеры публикуются на специальном сервере, к которому могут подключаться конечные потребители этого ПО и получать обновления по мере их публикаций, что позволяет построить непрерывную цепочку доставки ПО пользователям. Установочные пакеты публикуются на WebDAV серверах для самостоятельного скачивания и установки пользователями.