

## СИСТЕМЫ НЕПРЕРЫВНОЙ СБОРКИ И ИНТЕГРАЦИИ ПРОЕКТОВ

**Аннотация:** В статье рассматривается актуальная проблема возможности применения методики непрерывной интеграции в рамках разработки программного обеспечения.

**Ключевые слова:** непрерывная интеграция, программное обеспечение, разработка, тестирование, система контроля версий.

**Annotation:** The article deals with the actual problem of the possibility of using the method of continuous integration in the framework of software development.

**Key words:** continuous integration, software, development, testing, version control system.

Репутация современных компаний во многом зависит от работоспособности используемых информационных систем. Любые ошибки, допущенные при разработке таких систем, могут привести к значительным затратам материальных и трудовых ресурсов. Так, в апреле 2019 года технологический сбой, вызванный ошибками подрядчика и обновлением программного обеспечения, на семь часов нарушил работу всех сервисов Сбербанка. Ниже будет рассмотрена методология, позволяющая избежать подобного рода ошибок.

Как правило, сложное программное обеспечение (далее - ПО) имеет модульную структуру, то есть состоит из программных модулей, которые разрабатываются разными группами разработчиков независимо друг от друга. К

интеграции модулей приступают только тогда, когда они практически полностью написаны. Соответственно, если на этом этапе в коде обнаружатся ошибки или возникнут проблемы с взаимодействием отдельных компонентов, процесс интеграции может затянуться на неопределенное время.

Существующая практика непрерывной интеграции предполагает, что сборка кода начинает проводиться как можно раньше и повторяется как можно чаще (как минимум — при появлении изменений в коде). Это позволяет находить ошибки или противоречия в различных компонентах на ранних этапах создания информационной системы. В результате снижаются риски увеличения сроков разработки системы и обеспечивается высокое качество программного кода.

В настоящее время цикл разработки, в самом общем случае, состоит из следующих шагов: кодирование - тестирование - сборка - развертывание. Использование комбинированного подхода: методологии непрерывной интеграции (continuous integration, CI) при разработке ПО в сочетании с автоматизированным тестированием, позволит упорядочить процесс создания ПО, автоматизировать этапы сборки и тестирования и сделает развитие проекта более прогнозируемым и эффективным. В качестве рабочих процессов в CI можно выделить коммит, тестирование, сборку, регистрацию ПО и развертывание. Коммит – это изменение программного кода, зарегистрированное репозиторием и отправленное для тестирования CI-боту. Под тестированием понимается процесс контроля качества продукта и его целостности. Сборка ПО заключается в интеграции модулей и описана в начале статьи. После процесса сборки продукт регистрируется в системе контроля версий, как новый релиз. Процесс непрерывной интеграции завершается развертыванием ПО в тестовой и/или в рабочей среде.

Перед началом работы с CI, необходимо:

- внедрить систему контроля версий исходного кода (Git, TortoiseSVN);
- обеспечить полноту и качество покрытия разрабатываемого ПО тестами. Для непрерывной интеграции программного обеспечения требуется создать

репрезентативный набор тестов, то есть набор, охватывающий все возможные сценарии работы ПО;

- выбрать систему непрерывной интеграции и доставки, имеющую все функции, необходимые в процессе разработки ПО, запускающуюся от любых изменений кода в репозитории, а также развертывающую сборки там, где необходимо.

Внедрение методологии "Continuous Integration" предполагает несколько преимуществ:

- 1) полная проверка системы на каждом шаге её разработки;
- 2) отсутствие ошибок сборки, совершенных интегратором;
- 3) автоматизация процессов: сборки проектов, тестирования и составления отчетов о результатах тестирования;
- 4) хорошая абстракция и локализация ошибок;
- 5) высокая скорость и возможность нагрузочного тестирования;
- 6) освобождение трудовых ресурсов для реализации других задач;
- 7) максимальное исключение человеческого фактора.

Главной составляющей изучения роли методологии "Continuous Integration" и результатов ее внедрения служит реализация описанных выше принципов на практике. Субстратом реализации стала SCADA-система, предотвращающая аварии на нефтетрубопроводах. Основной проблемой при разработке системы послужило выделение большого количества времени на ручную сборку модулей и развертывание проекта. Скрипты сборки не решили проблему полностью, так как по-прежнему требовалась традиционная практика контроля качества. Реализация принципа непрерывной интеграции позволила разработчикам в полной мере сконцентрироваться на разработке программного кода и написании тестов к нему.

С целью управления процессом непрерывной интеграции был установлен и сконфигурирован CI-бот GoCD, являющийся бесплатным свободно распространяемым продуктом. Вся команда обладает доступом к GoCD через интернет-браузер и своевременно узнает, что происходит со сборками, а также получает последние результаты тестов и имеет возможность просмотреть

историю внесения изменений. CI-бот автоматически, по несколько раз в день, запускает свои процессы при регистрации изменений программного кода в репозитории TortoiseSVN (в терминологии это называется commit). Работа по внесению изменений завершается автоматически, выводом результатов сборки и развертывания и регистрацией релиза в репозитории.

Таким образом, в данной статье были рассмотрены такие ключевые этапы разработки программного обеспечения, как тестирование, сборка и развертывание. Были выделены наиболее актуальные проблемы, возникающие при выполнении данных этапов, а также было предложено их решение в виде внедрения методологии "Continuous Integration" в процесс разработки. В заключении была приведена реализация данной методологии на примере SCADA-системы.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Непрерывная интеграция и функциональное тестирование: два ключевых фактора разработки качественной информационной системы: [Электронный ресурс] //epam URL: <https://www.epam-group.ru/about/news-and-events/in-the-news/2011/nepreryvnaya-integraciya-i-funkcionalnoe-testirovanie-dva-klyuchevyh-faktora-razrabotki-kachestvennoy-informacionnoy-sistemy-cnews> (Дата обращения 10.05.2019).

2. Чинков М.Ю. Реализация методологии "Continuous Integration" в разработке веб-приложения: [Электронный ресурс] //Библиофонд URL: <https://www.bibliofond.ru/view.aspx?id=897662>.