

Мартынов Т. Е., Соснин А. С.

НЕПРЕРЫВНАЯ ИНТЕГРАЦИЯ И ДОСТАВКА

Тимофей Евгеньевич Мартынов

студент магистратуры

timtim96@bk.ru

Александр Сергеевич Соснин

студент магистратуры

salexandr18@gmail.com

ФГАОУ ВО «Российский государственный профессионально-педагогический университет», Россия, Екатеринбург

CONTINUOUS INTEGRATION AND DELIVERY

Timofej Evgenyevich Martynov

Aleksandr Sergeevich Sosnin

Russian State Vocation Pedagogical University, Russia, Yekaterinburg

Аннотация. В статье рассмотрена такая технология разработки, как непрерывная интеграция и доставка, проблемы выборки инструмента для ее реализации и перспективы этой технологии в ближайшие годы.

Abstract. The article deals with such development technology as continuous integration and delivery, the problem of selecting a tool for its implementation and the prospects of this technology in the coming years.

Ключевые слова: непрерывная интеграция, непрерывная доставка, развертывание, DevOps.

Keywords: continuous integration, continuous delivery, deployment, DevOps.

Платформы непрерывной интеграции и доставки (CI/CD) сейчас находятся на пике популярности. Интерес проявляют все отрасли, которые опираются на модернизацию в цифровой перспективе. Предлагая все более эффек-

тивное программное решение, CI/CD подвергается некоторой критике. В статье рассказано о месте CI/CD в разработке современного программного обеспечения и перспективах его развития в ближайшие годы.

Конвейер качества кода

На данный момент, использование цифровых инструментов в принятии торговых решений играет решающую роль. Наряду с усложнением высокотехнологичной платформы важным фактором стала непрерывность работы критических ИТ-систем. Эта тенденция была затронута и при разработке программного обеспечения.

На этапе разработки возможность постоянно тестировать и доставлять код позволяет увеличить скорость разработки и качество готового решения. Постоянный (несколько раз в день) перенос промежуточного программного кода в общую основную ветку и тестирование результата приняло форму в концепции «непрерывной интеграции и поставки программного обеспечения». Эта инициатива была предложена Грейди Буч в 1991 году.

Принцип работы CI/CD аналогичен конвейеру: методика делает интеграцию, включающую различные типы автоматического тестирования, с последующей доставкой и развертыванием кода для завершенного готового продукта для конечного пользователя.

CI/CD-платформа, на базе которой применяется концепция, поддерживает регулярную автоматическую сборку проекта для быстрого выявления дефектов и интеграционных задач. При стандартном подходе (каскадная разработка программного обеспечения или waterfall-метод), когда разработчики работают независимо над разными частями системы, этап интеграции является окончательным, и при обнаружении ошибок он может непредсказуемо задержать завершение работы. Переход к непрерывной интеграции позволяет снизить трудоемкость работы и делает ее более предсказуемой за счет раннего и непрерывного обнаружения и устранения ошибок и противоречий.

По своей сути, концепция CI/CD реализует идеологии смешивания разработки и эксплуатации программного обеспечения и соответствует основным

принципам Agile в виде рекомендаций по использованию автоматизированного тестирования для быстрой отладки рабочей версии программного обеспечения.

Если Agile — это устранение разрыва между клиентами и разработчиками, а DevOps — это преодоление границ между разработчиками и администраторами, то CI/CD — это воплощение стратегий DevOps на практике.

Обсуждаемость

На текущий момент, CI/CD самый обсуждаемый механизм разработки программного обеспечения, который стремится реализовать почти все этапы. Со временем, скорее всего, появятся какие-то места, где CI/CD признается, как оптимизированный инструмент, со статусом «золотого стандарта», но и waterfall-метод не исчез полностью.

Теперь можно сказать, что методика идеально подходит для любой работы, где разрабатывается новое программное обеспечение, особенно на базе микросервисной архитектуры.

CI/CD не подходит для внесения корректировок в монолитную ИТ-систему без дальнейшей доработки (например, крупная процессинговая система для банка с выпуском обновлений 3–4 раза в год), поскольку такая деятельность потребует реорганизации всей архитектуры.

Другой вопрос, что не каждое место, даже новую архитектуру для новой системы можно подстроить под CI/CD. Пока очевидным ограничением является то, что процедура работы охватывает все, что связано с веб-разработкой, e-commerce, омниканальные решения, одним словом, весь комплекс frontend и middleware компонентов. Для текущей базовой системы каскадная модель по-прежнему является лучшим вариантом.

Особенности развертывания платформы CI/CD

Основная трудность перехода к CI/CD — скорректировать подход, где приложения на первом месте, а технологии только во втором. Необходимо создавать новые процессы, определять новые роли людей, находить точки интеграции существующих и новых приложений. Перенос акцента с программного

и аппаратного обеспечения на людей и организационные аспекты работы стал серьезной проблемой для многих людей.

Второй момент — в CI/CD ответственность разработчика за конечный результат стала намного больше. Он пишет абстрактный код, но и тут же тестирует его своими силами. Так повышаются требования к способностям специалиста.

Не надо путать с классическими системными администраторами, обязанности которых заключаются в настройке оборудования, системы и выполнении настройки основных компонентов системы, с новым поколением сотрудников профиля CI/CD. CI/CD эксперт должен понимать не только фундаментальные, но и ИТ-надстройки — Kubernetes, TeamCity, Jenkins и другие поддержки платформы. Можно сказать, что в CI/CD-концепцию была добавлена стандартная поддержка ИТ-инфраструктуры компании.

Еще одна особенность CI/CD — огромный выбор программных инструментов с открытым исходным кодом. В то же время, многие известные платформы включают в себя наиболее доступные инструменты. CI/CD эксперт обеспечивает благоприятный выбор инструментов, доступных на рынке, ориентируясь на цели проекта, а также на карту их развития.

Нетривиальная задача выбрать инструменты, которые наиболее хорошо подойдут для того, чтобы соответствовать желаниям клиента и специфическому проекту.

В целом, переход на CI/CD является сложным процессом, традиционные компании сталкиваются с рядом трудностей на этом пути. Например, при переподготовке сотрудников пересмотр текущего рабочего процесса неизбежен, и это может привести к недовольству со стороны руководителя.

Бонус и преимущества

В CI/CD необходим быстрый цикл обратной связи, позволяющий им практически сразу узнать, как происходит качественное изменение кода и функциональности продукта. При водопадном подходе можно быстро перейти

на код, но без постоянных проверок ошибок потребуется гораздо больше времени, и они могут быть обнаружены после начала коммерческой эксплуатации, а таких «задержек» в релизе может быть несколько.

Инструменты CI помогают быстро ответить на вопросы о причинах дефектов для каждой фиксации, обеспечивая раннее обнаружение и устранение неполадок. Платформа CI/CD помогает не только быстро протестировать, но и донести до конечного пользователя новый функционал таким образом, что в случае ошибки всегда есть шанс быстро устранить ее или «откатить» этап решения на шаг назад.

Некоторые ошибки в программном обеспечении могут содержать другие, другие включать в себя третьи и так далее. Чем больше ошибок накапливается, тем сложнее их проверить, и они могут привести к неприятным результатам. Между тем, в разработке с использованием CI/CD, автоматические тесты, в случае ошибки, показывают, что нужно исправить. Конечно, на внедрение системы потребуется время, но это поможет программному обеспечению развиваться максимально быстро и удобно.

Автоматический процесс поможет значительно снизить трудозатраты различных подразделений предприятия. Без автоматизации CI/CD ошибки могут возникать из-за человеческого фактора и необходимости ручного управления.

Главные преимущества CI/CD:

- Скорость вывода функционала от запроса клиента до ввода в эксплуатацию. CI/CD позволяет запускать обновление в течение нескольких дней или недель, что значительно быстрее, чем весь календарный год с классическим подходом водопада. Новая услуга – новые конкурентные преимущества бизнеса. Становится возможным не только разрабатывать решения, как у конкурентов, но и достаточно опережать их в разработке и внедрении нового инструмента.

- Возможность выбирать оптимальные варианты благодаря быстрой проверки и большем числе итераций. Если вы откажетесь работать над бесперспективным решением, то сэкономите ресурсы компании.

- Качество конечного результата выше: самотестирование охватывает все аспекты продукта, что сложно реализовать при стандартном доступе к релизу. Все ошибки и узкие места выявляются и устраняются на начальном этапе разработки.

Основные недостатки CI/CD:

- Соклазн перенести на Agile, DevOps и CI/CD все, что связано с корпоративными ИТ-системами, в том числе и на уровне ядра, без получения первичного опыта. Это может серьезно нарушить работу компании, особенно в случае плохой организации перехода на новый механизм.

- Необходимо поддерживать правильный уровень взаимодействия между CI и CD. Быстрый и качественный результат от применения процедуры возможен только после долгой и тщательной настройки взаимодействия команды DevOps, инженеров, экспертов scrum и руководства компании. Самая сложная часть CI/CD – человеческий фактор, установление здорового командования рабочей программой, которая не может быть автоматизирована.

Мода и нехватка

Важно понимать, что все методологии и технологии не являются взаимоисключающими, а дополняют друг друга. Вопрос лишь в том, займет ли механизм и технология определенное место в общей ИТ-среде.

Сейчас мода на CI/CD накладывается на нехватку специалистов. Внедрение технологий с более высоким уровнем качества в таких условиях — не дешевый процесс. В то же время нельзя сказать, что это как-то особенно дорого. В принципе, на рынке присутствуют все необходимые инструменты: программное обеспечение для CI/CD не стоит космических денег. Также возможно освоение CI/CD технологии, в том числе собственными специалистами компании.

Другое дело, что бизнесу зачастую просто некогда вдаваться во все детали, а тем более осваивать их на практике. В этой ситуации самым распространенным окном CI/CD в компании стала новая работа, которая находится под профилем задействованной команды. После запуска проекта установлено, что опыт, инфраструктура и платформа CI/CD применимы и в других областях для решения других задач компании. Такое часто случается в телекоме, ритейле и банках — сначала проект становится «песочницей», где прототипированием занимаются квалифицированные специалисты, а затем результаты их исследований масштабируются на всю организацию.

Критерием для определения правильности выбранной стратегии развития CI/CD является сравнение эффективности процесса с лидерами рынка, которые способны интегрировать новый функционал, протестировать его и привести в действие в течение 2–3 часов.

Подводя итог, CI/CD, можно сказать, лучший метод разработки программного обеспечения для нашего времени, который в своем развитии проходит через фазу активного обсуждения. Это не хорошо и не плохо — это просто этап, который закончится, и тогда мы увидим объективное место системы CI/CD механизма на пути современной разработки программного обеспечения.

Список литературы

1. Continuous Integration для новичков [Электронный ресурс]. – Режим доступа: <https://habr.com/post/352282/> (дата обращения: 22.12.2018).
2. CI/CD: принципы, внедрение, инструменты [Электронный ресурс]. – Режим доступа: <https://medium.com/southbridge/ci-cd-принципы-внедрение-инструменты-f0626b9994c8> (дата обращения: 22.12.2018).
3. Непрерывная интеграция [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Непрерывная_интеграция (дата обращения: 22.12.2018).

4. Что такое непрерывная интеграция [Электронный ресурс]. – Режим доступа: <https://ru.stackoverflow.com/questions/470453/Что-такое-непрерывная-интеграция/471034> (дата обращения: 22.12.2018).

УДК 004.415.04'2

Мартынов Т. Е., Соснин А. С.

**СРАВНЕНИЕ ИНСТРУМЕНТОВ CI/CD: JENKINS, BAMBOO,
CRUISECONTROL, TEAM CITY, TEAM FOUNDATION**

Тимофей Евгеньевич Мартынов

магистрант

timtim96@bk.ru

Александр Сергеевич Соснин

магистрант

salexandr18@gmail.com

ФГАОУ ВО «Российский государственный профессионально-педагогический университет», Россия, Екатеринбург

**COMPARISON OF TOOLS CI / CD: JENKINS, BAMBOO,
CRUISECONTROL, TEAM CITY, TEAM FOUNDATION**

Timofej Evgenyevich Martynov

Aleksandr Sergeevich Sosnin

Russian State Vocation Pedagogical University, Russia, Yekaterinburg

Аннотация. В статье описываются популярные инструменты, реализующие практики CI/CD. Описаны особенности каждого из рассматриваемых инструментов, позволяющие выбрать наиболее подходящий.

Abstract. The article describes popular tools that implement CI/CD practices. Describes the features of each of the analyzed tools, allowing you to select the most suitable one.