



Release

Build (Билд -Сборка) — это версия ПО, готовая к тестированию. Билд устанавливается на стенд тестирования, хранится в системе управления версиями кода (VCS). Если в билде/сборке найдены дефекты, то после их исправления билд собирается заново и ему присваивается новый номер/версия, происходит новая установка на стенд тестирования, снова запускается процесс тестирования и ретест исправленных дефектов.

Release (Релиз – выпуск) – версия ПО, которая уже покрыта одним или несколькими этапами тестирования, в которой все критические дефекты исправлены. Релиз собирают и передают для финального этапа тестирования, после которого следует внедрение или передача заказчику. Релизом также можно назвать временной интервал, в рамках которого разрабатывается и тестируется один или несколько инкрементов (инкремент — это часть кода, меняющая или дополняющая существующий функционал).

Связь между сборкой и выпуском в тестировании ПО.

Один выпуск может иметь несколько сборок, в то время как выпуск основан на сборках.

Разница между сборкой и выпуском в тестировании программного обеспечения:

1. Определение.

Под сборкой понимается автономный программный артефакт, сгенерированный после преобразования исходного кода в исполняемый код, который можно запустить на компьютере. Релиз, с другой стороны, является распространением окончательной версии приложения. Таким образом, эти определения объясняют принципиальную разницу между сборкой и выпуском.

2. Ассоциация с тестированием

Тестирование является основным отличием между сборкой и выпуском. Сборка все еще находится в фазе тестирования (может быть уже протестирована или еще не проверена), но релиз больше не требует тестирования

3. Основа

Команда разработчиков передает сборку команде тестирования, в то время как команда тестирования предлагает релиз своим клиентам.

4. Вхождение

Другое различие между сборкой и выпуском заключается в том, что сборка происходит часто, а выпуск - время от времени.

Итог:

Существует четкое различие между сборкой и выпуском, хотя эти два слова часто используются взаимозаменяемо. **Основное различие между Build и Release** в тестировании программного обеспечения заключается в том, что Build - это версия ПО, которую команда разработчиков передает команде тестирования для целей тестирования, а Release - это ПО, которое команда тестирования передает клиенту.

Релизный цикл.

Релизный цикл - период времени, за который новый функционал проходит путь от принятия решения о его реализации до выхода в продуктивную версию приложения. Методологии разработки программного обеспечения, средства для непрерывной интеграции, системы для релиз-менеджмента, используемые технические средства на различных средах развертывания - все это вкуче и является релизной методологией. За соблюдением релизной методологии обычно следит релизный менеджер.

Обязанности менеджера релиза (неполный перечень) :

- отправляет новые строки на переводы и добавляет готовые переводы;

- организует встречу с представителями команд, участвующих в релизе, где подробно расспрашивает об изменениях, рисках, зависимостях и т. д.;
- создаёт релизную ветку, разворачивает её на тестовых серверах и следит за работоспособностью тестового окружения;
- мониторит логи во время стабилизации, раздаёт найденные баги;
- находится в тесной связи с QA-координатором, который контролирует стабилизационное тестирование, отвечает на его вопросы, помогает найти ответственную команду при нахождении багов;
- контролирует сроки; при невозможности вписаться с таймлайн обсуждает это с владельцами продукта, менеджментом и другими заинтересованными лицами;
- просматривает все изменения в коде после исправления багов;
- отдаёт релиз на стейджинг и отвечает на вопросы OPS;
- в день релиза и последующую неделю мониторит логи;
- при необходимости координирует патчи и хотфиксы, которые являются упрощёнными релизами;
- участвует в ретроспективе релиза.

Что НЕ входит в задачи менеджера релиза:

- развёртывание обновления в продакшене — для этого есть OPS;
- исправление найденных багов — за это ответственны команды.

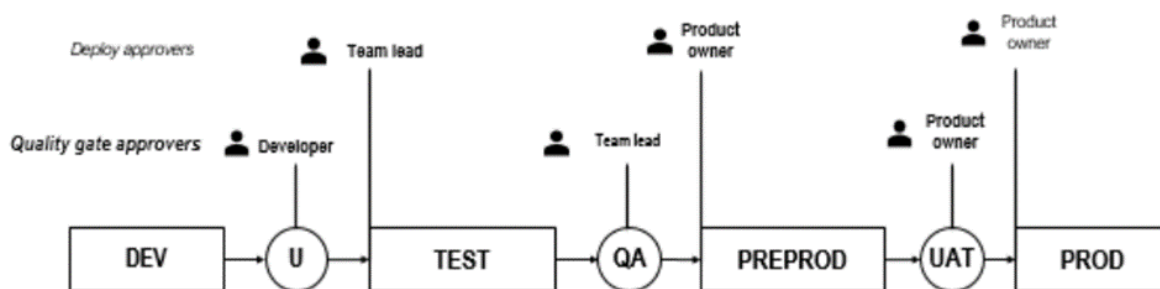
Релиз-менеджмент подразумевает под собой не только теоретически хорошо продуманную стратегию, но и работу с различными инструментами, определенные действия разработчиков или менеджеров:

- Работа с инструментами непрерывной интеграции (Jenkins)
- Работа с инструментами планирования, управления проектами (Jira, Trello, MS Project)
- Работа с системой контроля версий Git
- Планирование релизов, заведение расписания релизов
- Работа с корпоративными мессенджерами (Slack, Skypeforbusiness).

Релизный цикл отдельного ПО

1. Сбор требований заказчиков, анализ и включение требований в релиз – подготовка и согласование новых возможностей, которые требуется добавить в ПО. Составляется список задач и необходимых изменений.
2. Утверждение списка изменений, в т.ч. определяется приоритетность добавления нового функционала ПО.
3. Разработка новой версии ПО.
4. Тестирование новой версии ПО разработчиком.
5. Установка новой версии ПО в тестовый кластер/облачную тестовую среду (по желанию заказчика)
6. Установка новой версии ПО в основной кластер заказчика.

Стандартный релизный цикл



Команда разработки работает по методологии скрам и имеет трехнедельный спринт. При разработке используется четыре среды развертывания - среда разработки (DEV), тестовая среда (TEST), препродуктивная среда (PREPROD) и продуктивная среда - конечный продукт (PROD). Таким образом, каждую третью неделю происходит релиз на продуктивной среде.

Релизу на каждой следующей среде развертывания предшествует релиз на предыдущей. Чтобы он состоялся, должен быть выполнен набор тестов, успешное прохождение которых будет говорить о том, что функционал готов для перехода на следующую среду развертывания. Релиз на каждую из сред развертывания, как правило, должен быть подтвержден определенном членом команды разработки. Для релизов на разные среды развертывания эту роль могут выполнять разные люди. Кроме того, всегда существуют различные факторы, которые могут повлиять на процесс разработки и выпуска релизов: изменение требований бизнесом, внезапно возникшие ошибки, даже болезнь кого-нибудь из команды - все это может сыграть роль. В таких ситуациях нередко релизы приходится сдвигать, перепланировать. Некоторые задачи,

запланированные на текущую итерацию разработки переносятся на следующие, сами релизы выходят с опозданием или отменяются вовсе.

Стадии разработки ПО

Используются для описания степени готовности ПО, может отражать количество реализованных функций, запланированных для определённой версии программы. Стадии либо могут быть официально объявлены и регламентируются разработчиками, либо иногда этот термин используется неофициально для описания состояния продукта.

Pre-Alpha — начальная разработка

Начальная стадия разработки — период времени со старта разработки до выхода стадии альфа. Также так называются программы, не вышедшие ещё в стадию альфа или бета, но прошедшие стадию разработки, для первичной оценки функциональных возможностей в действии. В отличие от альфа- и бета-версий, начальный этап может включать в себя не весь спектр функциональных возможностей программы. В этом случае подразумеваются все действия, выполняемые во время проектирования и разработки программы вплоть до тестирования.

К таким действиям относятся: разработка дизайна, анализ требований, собственно разработка приложения, отладка отдельных модулей.

Alpha — внутренняя разработка

Стадия начала тестирования программы в целом специалистами-тестировщиками, обычно не разработчиками программного продукта, но, как правило, внутри организации или сообществе разрабатывающих продукт. Также это может быть стадия добавления новых функциональных возможностей. Программы на данной стадии могут применяться только для ознакомления с будущими возможностями. Как правило, альфа-тестирование заканчивается заморозкой функциональности и переходит в бета-тестирование.

Beta — общественная разработка

Стадия активного бета-тестирования и отладки программы, прошедшей альфа-тестирование (если таковое было). Программы этого уровня могут быть использованы другими разработчиками программного обеспечения для испытания совместимости. Тем не менее программы этого этапа могут содержать достаточно большое количество ошибок. Поскольку бета-продукт не является финальной версией и публичное тестирование производится на страх

и риск пользователя, производитель не несёт никакой ответственности за ущерб, причинённый в результате использования бета-версии.

Release candidate (релиз-кандидат) — предварительная версия

После альфа и бета-тестирования все возможные ошибки уже устранены и программа практически стабильна. Однако есть ещё вероятность, что обнаружатся баги, поэтому разработчики выпускают программы именно в этой версии – RC. Во многих случаях может выйти несколько версий RC – 1, 2 и т.д.

Release (RTM /от англ. release to manufacturing/, Final, Stable) - Выпуск

Это финальная версия программы, готовая к использованию. В ней исправлены практически все ошибки, она обладает полным функционалом, работа её стабильна и протестирована многими пользователями ранее.

Release to manufacturing - выпуск в производство

Обозначение готовности программного продукта к тиражированию. Это стабильная версия программы, прошедшая все предыдущие стадии, в которых исправлены основные ошибки. RTM предшествует общей доступности (GA), когда продукт выпущен для общественности, готов к массовому розничному распространению. Этот термин не определяет механизм или объём поставки; он лишь указывает, что качество является достаточным для массового тиражирования.

General availability - общедоступность

Общедоступность (англ. general availability) или общепринятость (англ. general acceptance, GA) — стадия маркетинга, на которой завершены все необходимые мероприятия по коммерциализации, и программный продукт доступен для покупки, в зависимости, однако, от языка, региона, электронной или медийной доступности. Деятельность по коммерциализации может включать проверку безопасности и соответствия требованиям, а также локализацию и продвижение по всему миру. Время между выпуском в производство и общедоступностью может составлять от недели до нескольких месяцев. Это время необходимо для завершения всех мероприятий по коммерциализации, требуемых GA. На данном этапе программное обеспечение «вышло в жизнь» (gone live).

Release to web - веб-релиз

Выпуск в интернет (RTW) или веб-релиз является средством доставки программного обеспечения, которое использует интернет для его распространения. При этом изготовитель не задействует никакие физические

носители. Веб-релизы становятся все более распространёнными по мере роста использования интернета.

Support - Поддержка

В течение поддерживаемого срока службы ПО к нему выпускаются сервисные выпуски (service releases), патчи или пакеты обновления, иногда также называемые «промежуточные выпуски» (interim releases).

End-of-life - Окончание срока службы

Когда программное обеспечение больше не продается или не поддерживается, считается, что срок службы продукта истек, что он снят с производства, списан, устарел, заброшен или устарел, но лояльность пользователей может сохраняться еще некоторое время, даже после того, как его платформа устарела.

Milestones.

Milestone - термин, использующийся в управлении проектами, который означает важную веху, ключевой этап, переход на новую стадию в плане разработки. Как правило, этим этапом является завершение какого-либо ключевого мероприятия, подписание важных документов или любые другие значительные действия, предусмотренные планом проекта. Сдвиг контрольной точки приводит к сдвигу всего проекта.

В дополнение к сигнализации о завершении некоего ключевого этапа, milestone также может означать принятие важного ключевого решения, которое может сильно повлиять на судьбу всего проекта. В этом смысле, milestone показывает не только контрольные точки процесса, но и указывает направление движения. В каждый этап разработки ПО добавляется изменение, оказывающее влияние на его функционал в целом, это происходит в пределах каждой вехи. Обычно каждой вехе присваивается порядковый номер. Вехам придается дополнительное значение по сравнению с задачами в плане, чтобы менеджер проекта мог отслеживать задачи, в то время как команда и заинтересованные стороны сосредоточены на дальнейшем прогрессе.

Менеджеры используют вехи для того, чтобы обозначить важные промежуточные результаты, которые должны быть достигнуты в процессе реализации проекта.

Веха имеет малую или нулевую длительность и важное управленческое значение. Вехи отражают значимые для проекта события, например прохождение фазы, получение одобрения на реализацию и т. п.; контрольные

точки более характерны для целей текущего контроля: проведения аудита, иной проверки, сбора отчетов и др.

Контрольная точка, или веха, связана с работой, окончание которой она символизирует. Для эффективного контроля этих точек должно быть достаточно много, и их распределение по жизненному циклу зависит от проекта. Очевидны контрольные точки начала и окончания проекта, перехода между фазами, этапами, проведения контроля, сдачи объектов, получения одобрения, поступления инвестиций и т. д.

Последовательность вех, определенных менеджером, часто называется план по вехам. Даты достижения соответствующих вех образуют календарный план по вехам.

Определение вех устанавливает основу для взаимодействия по согласованию основных стадий разработки проекта, а также для оценки и контроля на высшем уровне и, таким образом, является ключевой частью процесса планирования на раннем этапе.

Определение вех требует ясного понимания ключевых стадий или состояний, через которые проходит проект в течение своего жизненного цикла и четких характеристик точек перехода. Вехи отмечают эти ключевые переходы и, следовательно, обеспечивают естественную серию контрольных точек.

Эти контрольные точки соответствуют специфическим промежуточным целям, требуемым для достижения общей цели. При обзоре выполнения работ вехи позволяют руководству быстро понять, на какой стадии находится проект и оценить, достигнуты ли основные состояния, необходимые для успешного завершения работ.

В отличие от работ, вехи не имеют продолжительности, поэтому они иногда называются событиями. Веха является мерой выполнения (контрольной точкой), ее завершенность имеет только две оценки - выполнена или нет.