

Build и Release

Главное отличие между сборкой и выпуском в тестировании программного обеспечения является то, что Build - это версия программного обеспечения, которую команда разработчиков передает команде тестирования для целей тестирования, а Release - это программное обеспечение, которое команда тестирования передает клиенту.

Разработка программного обеспечения - сложный процесс. Это сложнее, чем написание обычной компьютерной программы, так как заказчик работает с конечным рабочим продуктом. Поэтому важно протестировать программное обеспечение. Тестирование - это процесс проверки и подтверждения правильности работы программного обеспечения в соответствии с требованиями. Это помогает снизить затраты и доставить работающий продукт клиенту. Build и Release - это два термина в разработке и тестировании.

Что такое Build в тестировании ПО

После разработки программного модуля разработчики преобразуют исходные коды в автономную форму или исполняемый код. Затем команда разработчиков передает сборку команде тестирования для выполнения тестирования. Сборка находится в стадии тестирования; возможно, он уже прошел тестирование или нет. Команда тестирования программного обеспечения проверяет эту сборку. Если он состоит из нескольких ошибок и не соответствует требованиям, команда тестирования программного обеспечения отклоняет эту сборку. Сборки происходят до релиза, и они генерируются чаще.

Что такое Release в тестировании ПО

Релиз является окончательным приложением после завершения разработки и тестирования. После тестирования сборки группа тестирования сертифицирует это программное обеспечение и доставляет его заказчику. Для одного выпуска возможно иметь несколько сборок.

Таким образом, это программное обеспечение доставляется заказчику после завершения этапов разработки и тестирования. Более того, релиз основан на сборках и может иметь несколько сборок.

Связь между сборкой и выпуском в тестировании программного обеспечения

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

Разница между сборкой и выпуском в тестировании программного обеспечения

Билд/Сборка — это версия программного обеспечения, 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.

Релизом называют версию программного продукта, 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)

Релизный цикл

Процесс управления релизами актуален для компаний, деятельность которых связана с информационными технологиями.

Обычно работа над программным обеспечением, ИТ-системами и сервисами не заканчивается после этапа внедрения. Ведь улучшение и наращивание возможностей с учетом потребностей пользователей и roadmap продукта — неотъемлемая часть его жизненного цикла.

Понять, какие изменения необходимы на разных этапах, запланировать и реализовать их, ничего не упустив, помогает автоматизация управления релизами.

Что такое управление релизами

Релизами называют запланированные изменения, которые вносят в программное или аппаратное обеспечение.

Управление релизами по ITIL – это комплексный процесс, объединяющий планирование, реализацию, тестирование и внедрение изменений в ИТ-продукт, а также контроль их качества при дальнейшем использовании.

Из каких этапов состоит процесс управления релизами



Ключевые этапы процесса управления релизами

1. Планирование

На этом этапе собираются требования к продукту от клиентов и внутренних заказчиков, анализируется актуальность предложенных изменений с учетом ресурсов, компетенций, времени на реализацию и совместимости с уже существующими функциями ПО.

Приоритетные запросы группируются и разделяются на блоки, задачи и подзадачи. Составляется подробный план релиза со сроками и контрольными точками. Указываются специалисты, которых необходимо привлечь к участию в проекте.

Существует множество методик подготовки к реализации проекта. Одна из таких стратегий – это концепция планирования релизов, которая преимущественно используется в Agile совместно с построением дорожной карты проекта (Roadmap). Менеджер продукта – это «навигационная система» компании. Он определяет курс (создает видение проекта) для команды, а затем организует фактическую работу, необходимую для воплощения этого видения в реальность.

Обычно для этого создается дорожная карта продукта, в которой тоже указываются сроки всех нужных релизов. Она также помогает продумать общую картину. А вот планирование релиза уже отобразит пошагово все детали проекта.

2. Сборка релиза

Организуется процесс разработки и стартует реализация намеченных функциональных изменений. Объемный по содержанию релиз разбивается на спринты и итерации.

3. Тестирование

Проводится тестирование доработок в собственной ИТ-среде. При необходимости формируются группы пользователей, которым предоставляется доступ к бета-версии для выявления критичных багов и первичной «обкатки» новых возможностей. По итогам тестирований производится финальная подготовка релиза с исправлением всех ошибок.

4. Развертывание

Осуществляется выпуск изменений, передача в эксплуатацию, документирование изменений. Выходит новая версия или сам конечный продукт. Все пользователи получают доступ к функциям, которые разрабатывались в релизе.

5. Контроль качества

После развертывания участвующая в релиз менеджменте команда подводит итоги, анализирует выполненное. Если при внедрении новых функций возникает потребность в сопутствующих доработках, специалисты фиксируют их и планируют дальнейшее усовершенствование продукта в рамках нового релиза.

Контроль качества с помощью Quality Gates. Что такое «ворота качества», иначе говоря, проверки качества кода? Каждый день команда разработки пишет код для новых фич, которые могут сломать нашу крепость. Задача QA — тестировать каждую фичу и уменьшать вероятность попадания бага в код продукта. Quality Gates – это автоматические проверки качества, которые устанавливают пороговые значения для продвижения продукта по конвейеру разработки. Каждая из проверок будет отвечать за прохождение важной метрики. Если метрика не соблюдается, «ворота» должны закрыться и не пустить фичу дальше. Пока не пройдут все проверки, фича не будет залита в код продукта, а значит потенциальные баги будут предотвращены.

Пример планирования спринтов и релизов

1. Для формирования плана релиза определяется набор новых фич и/или изменений текущего функционала, которые необходимо внести в продукт.
2. План релиза основывается на четырех факторах: элементах бэклога продукта, оставшегося объема работ, скорости команды и времени. План релиза не является фиксированным. Он меняется вместе с развитием Backlog'a продукта, а также понимания оставшегося объема работ и скорости.
3. Функциональность, которая внедряется в ходе каждого релиза, представляется в виде тем или эпиков (Epic) в Jira. Эпики разбиваются на спринты - небольшие временные периоды, в которые входят задачи, объединенные одной или несколькими микроцелями. Длительность спринта составляет от 1,5 до 3 недель. Все спринты регистрируются в Jira с указанием плановых сроков и дедлайнов вошедших в них задач.
4. Обычно для формирования релиза берутся задачи из Backlog, которые заранее зарегистрированы в Jira и оценены в Story Points.
5. Список задач, которые войдут в релиз, зависит от скорости команды разработчиков. Если скорость команды равна 50 Story Points за спринт из 2-х недель, то и задач в аналогичный спринт должно быть набрано на примерно такое же количество Story Points, при условии, что дата релиза

может быть назначена самой командой. Если дата релиза назначается руководством, то в список задач попадают задачи на такое количество Story Points, которое команда может выполнить за выделенное на релиз время, исходя из своей скорости.

6. Если работа не будет завершена в допустимые сроки (например, в случае фиксированной руководством даты релиза), то функциональность, внедрение которой запланировано в ходе релиза, сокращается. Также в этом случае возможны добавление людей в команду или передача задач менее опытных разработчиков более опытным (если по заключению team lead'a и project lead'a это действительно поможет увеличить скорость разработки, а не уменьшит ее).
7. В конце каждого спринта проводится ретроспектива, в процессе которой высчитывается скорость команды, анализируются отчеты по спринту в Jira, выявляются проблемы в спринте, предлагаются улучшения.

Пре-релиз продукта

Пре-релизные билды собираются в случае необходимости проведения экстренного тестирования до перевода продукта в релиз.

Процесс работы с пре-релизом

1. Задача на сборку пре-релизного билда выполняется согласно правилам принятым в компании
2. Исполнитель скачивает на свою локальную машину исходники проекта и указывает номер пре-релизного билда с учетом номера последнего коммита в master-ветке.
3. Пре-релизный билд собирается и отдается на тестирование.
4. Обнаруженные в процессе тестирования дефекты и необходимые для введения в продукт улучшения и фичи регистрируются в Jira как задачи типов Bug, Improvement или Feature.

Релиз продукта

Как только все задачи, запланированные в релизный билд, закрыты в Jira, продукт переводится в релиз.

Процесс работы с релизом

1. Задача на сборку релизного билда выполняется согласно Правилам сборки и оформления билдов.
2. От master-ветки отбрана ветка release-ветка и именуется соответственно Правилам именования веток.

3. В master-ветку коммитится файл с указанием номера потенциального пре-релиза (например, если создана release-ветка 3.1, то в CommonAssemblyInfo.cs, закоммиченном после создания release-ветки в master-ветку, указывается 3.2.0, без учета номера коммита).
4. В release-ветку коммитится номер релизного билда
5. Релизный билд собирается и отдается на тестирование.
6. Проводится тестирование релизного билда.
7. Если в процессе тестирования не обнаружено дефектов и необходимых для введения в продукт улучшений и фич:
 - последний коммит в release-ветку помечается tag'ом соответственно Правилам именования tag'ов;
 - release-ветка вмерживается в master-ветку соответственно Правилам merge request'ов;
 - релизный билд выводится в продакшн.
8. Если в процессе тестирования обнаружены дефекты и необходимые для введения в продукт улучшения и фичи:
 - эти задачи регистрируются в Jira как Bug, Improvement или Feature с указанием Affects Version (номер билда продукта, в котором обнаружены данные дефекты или необходимые улучшения и фичи);
 - разработчик, который занимается задачей, берет ее в Jira в прогресс;
 - разработчик отбранчевывает от release-ветки feature-ветку, именуюя ее соответственно Правилам именования веток;
 - как только все поставленные в процессе тестирования задачи закрыты в Jira, последний коммит в release-ветку помечается tag'ом соответственно Правилам именования tag'ов
 - release-ветка вмерживается в master-ветку соответственно Правилам merge request'ов;
 - релизный билд выводится в продакшн.
9. Если обнаружены дефекты в релижном билде, выведенном в продакшн:
 - работа над дефектами ведется в release-ветке данного билда;
 - дублируется процесс работы над дефектами, обнаруженными при тестировании релизного билда (пункт 8).
10. При необходимости, исправления дефектов из release-ветки можно вмерживать в master-ветку до формирования продакшн версии.

Жизненный цикл выпуска программного обеспечения

У релиза программного обеспечения есть свой жизненный цикл

Пре-альфа

Первый этап жизненного цикла релиза ПО называется «Пре-альфа» (англ. Pre-alpha). Пре-альфа относится ко всем видам деятельности на этапе разработки ПО до момента тестирования. На этом этапе выполняют такие действия:

Сбор и анализ требований

- Дизайн ПО
- Разработка ПО
- Модульное тестирование

Альфа

Второй этап - «Альфа» (англ. Alpha). На этом этапе впервые проводят комплексное тестирование ПО различными методами. Разработчики тестируют исправность функционирования приложений системы. По на этапе Альфа может бывает нестабильным, то есть может произойти сбой программы или потеря данных. После того, как все тесты проведены, данный этап жизненного цикла релиза ПО завершается утверждением набора функций и параметров ПО.

Третий этап – «Бета» (англ. Beta). Данный этап, как правило, начинается с завершения разработки всех функций ПО. На этапе Бета в программах находят больше технических дефектов, чем на предыдущих этапах (например, скорость работы или производительность системы). Этот этап предполагает проведение тестирования эксплуатационной пригодности. Процесс предоставления пользователям бета-версии программы называется Бета-релиз. По сути, это тот этап, когда программное обеспечение становится доступным для внешних пользователей, а не только для разработчиков внутри организации.

Закрытое и открытое Бета-тестирование

Четвертый этап – «**Закрытое и открытое Бета-тестирование**». То есть разработчики выпускают или закрытую версию ПО, или открытую. Закрытая Бета-версия программы доступна лишь для узкого круга пользователей, которые протестируют ПО. Соответственно, открытая Бета-версия доступна широкому кругу лиц, которые могут протестировать программу по своей инициативе. После чего, они сообщают разработчикам о возникших ошибках и иногда предлагают дополнить программу какими-то функциями, которые, по их мнению, должны присутствовать в финальной версии программы. Примерами основных открытых бета-версий программ могут служить:

Тестирование открытых Бета-версий применимо в двух случаях:

Пятый этап называется **«Релиз-кандидат»** (англ. Release candidate, RC)- это Бета-версия продукта, который имеет достаточно потенциала для того, чтобы стать финальной версией. На этом этапе продукт станет финальным релизом, в случае, если не возникнут серьезные дефекты. На данном этапе происходит стабилизация функционирования ПО – все функции разработаны, закодированы и протестированы. Этот релиз превращается в релиз с утвержденным кодом (code complete), когда вся команда разработчиков убеждена, что ни один новый код не будет добавлен в данную программу.

Релиз для производства

Шестой этап называется **«Релиз для производства»** (англ. «Release to manufacturing, RTM). Термин RTM, известный также, как «становящийся золотом» (англ. Going Gold), используется, когда продукт может предоставляться конечным пользователям. Чаще всего, подразумеваются розничные продажи широкому числу покупателей. Термин RTM также может означать, что продукт был предоставлен клиентам для инсталляции на своих аппаратных устройствах. Этот термин не определяет способ и объем доставки ПО потребителям, он всего лишь указывает на то, что на этом этапе качества продукта достаточно для массового распространения. Этот этап предшествует этапу жизненного цикла релиза ПО под названием **«Общедоступный Релиз»**.

Общедоступный Релиз

Седьмой этап - **«Общедоступный Релиз»** (англ. General availability, GA). Означает, что вся деятельность по коммерческой реализации завершена и ПО доступно широкой общественности или через Интернет, или с помощью физических носителей. Коммерческая деятельность может включать, но не ограничивается, организацией дистрибуции, подготовкой маркетинговых материалов на разных языках, определением целевых рынков, выполнением окончательных тестов. На этом этапе ПО «обретает жизнь». То есть данный релиз – это «живая» финальная версия конкретного продукта. На данном этапе ПО отличается своей стабильностью и практически не имеет дефектов. Такая программа пригодна для массового использования конечными потребителями. Этот релиз также известен как **«золотой релиз»** (англ. Gold Release).

«Веб-релиз» - означает, что для распространения программы используют Интернет, то есть не используются физические носители. В последние годы данный вид релиза стал более популярным, в связи с распространением Интернета. После всех вышеперечисленных этапов наступает этап

«промежуточных релизов». После внедрения программы, спустя определенный промежуток времени она будет нуждаться в поддержке.

Окончание срока службы

Завершающий этап **«Окончание срока службы»** (англ. End-of-life). Означает, что программа больше не продается и не поддерживается. ПО морально устаревает, но иногда лояльность пользователей может продлить программе жизнь на еще какой-то промежуток времени.

Milestone (майлстоун)

Milestone(майлстоун)— 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.