

Описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения “Платформа Jetskills”, в том числе устранение неисправностей, выявленных в ходе эксплуатации программного обеспечения, совершенствование программного обеспечения, а также информацию о персонале, необходимом для обеспечения такой поддержки.

Оглавление

Процесс разработки	3
Модель жизненного цикла.....	3
Определение требований и планирование	4
Оценка рисков проекта	5
Проектирование, разработка и проведение оценки	5
Интеграция новой версии ПО в продуктовую среду	6
Тестирование	6
Автоматическое тестирование	6
Ручное тестирование	7
Персонал	7
Обучение и квалификация персонала	9
Техническая поддержка	9
Совершенствование и модернизация ПО	10
Сообщения о неисправности.....	11
Эксплуатация	11

Процесс разработки

В процессе разработки программного обеспечения “Платформа JetSkills” (далее ПО) включает в себя следующие основные этапы:

- разработка и согласование ТЗ;
- согласование структуры ПО;
- разработка исходного кода ПО, проведение тестирования программных модулей;
- разработка программной документации;
- проведение интеграционного тестирования и устранения замечаний;
- проведение регрессионного тестирования и устранения замечаний;
- адаптация ПО к среде реализации.

Модель жизненного цикла

Для разрабатываемого ПО в качестве модели жизненного цикла используется спиральная модель, соответствующая масштабу и сложности проекта (рис. 1).

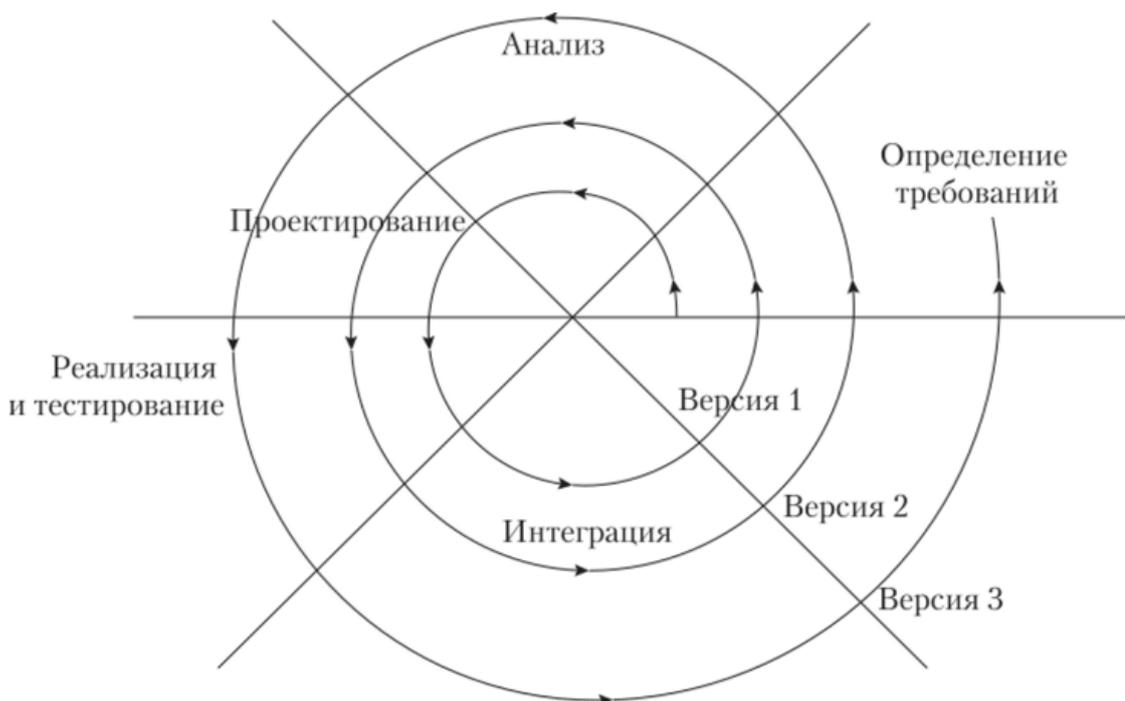


рис. 1 Спиральная модель жизненного цикла ПО

Спиральная модель схожа с инкрементной моделью, но в ней уделяется больше внимания оценки и разрешения рисков. Спиральная модель подразделяет реализацию проекта на четыре этапа:

- определение требований и планирование;
- оценка рисков;
- проектирование, разработка и проведение оценки;
- интеграция новой версии ПО в продуктовую среду.

Проект каждый раз заново проходит через эти четыре стадии при создании новой версии или фрагмента ПО (что соответствует одному витку спирали в данной модели).

Базовый виток спирали, начинающийся на этапе постановки задач, включает в себя **определение требований** и **разработку архитектуры ПО**.

Базовое определение требований

При определении требований к ПО определяется общая концепция разрабатываемого программного продукта, и на ее основе строится базовая структура проекта, оценивается его выполнимость и связанные с ним риски, описываются соответствующие подходы к конфигурационному управлению и технологиям.

Базовая разработка архитектуры

При разработке архитектуры определяется архитектурный проект системы, в соответствии с которым выполняется идентификация элементов ПО и удовлетворяются заданные требования. При определении верхнего уровня архитектуры системы должны быть идентифицированы составные части технических средств, программных средств и ручных операций. Должно быть учтено, что все системные требования распределяются между этими составными частями. Составные части конфигурации технических средств, программных средств и ручных операций должны последовательно идентифицироваться этими составными частями.

Каждый последующий виток строится на основе базового.

Определение требований и планирование

В процессе определения требований в качестве исходных данных используются цели, поставленные в разделе плана проекта, описывающем требования высокого уровня.

Каждый программный компонент должен иметь собственную Спецификацию требований.

Требования к ПО определяют функционал программного компонента, производительность, точность, временные характеристики работы, затраты ресурсов используемого оборудования, работоспособность в нестандартных условиях и при перегрузках. В Спецификации требований к программному обеспечению описываются алгоритмы и математические методы.

Результатами этапа являются план конфигурационного управления, план реализации качества ПО, план реализации проекта и календарный план, содержащий подробный список запланированных работ грядущего этапа, а также предварительная оценка трудозатрат на последующих этапах.

Оценка рисков проекта

Риск – это любое событие, которое может помешать реализации проекта в соответствии с планом или его успешному завершению. Риски можно идентифицировать из разных источников. Некоторые из них могут быть довольно очевидными и будут выявлены до начала проекта. Другие будут идентифицированы в течение жизненного цикла проекта, и риск может быть идентифицирован любым участником проектом. Некоторые риски будут присущи самому проекту, в то время как другие будут результатом внешних воздействий, которые полностью неподконтрольны команде проекта.

Проектирование, разработка и проведение оценки

На этапе проектирования в качестве исходных данных используются требования, определенные в принятой спецификации. По каждому требованию определяется элемент или набор элементов проектирования по результатам согласования с заказчиком.

Элементы проектирования подробно описывают требуемый функционал ПО и, как правило, включают в себя схемы функциональной иерархии, схемы расположения элементов визуализации на экране, таблицы правил, схемы деловых процессов, псевдокод, а также схему всех потоков данных с полным словарем данных. Эти элементы проектирования предназначены для описания ПО в объеме, достаточном для того, чтобы квалифицированные специалисты могли разработать ПО с минимальной потребностью в дополнительных данных.

На этапе разработки ПО в качестве исходных данных используются элементы проектирования, описанные в принятом плане разработки. По каждому элементу определяется артефакт или набор артефактов ПО. Артефакты ПО включают в себя (но не ограничиваются ими) меню, диалоговые окна, формы для управления данными, форматы отчетных данных и специализированные процедуры, и функции.

В процессе разработки разработчики используют систему управления версиями исходного кода Git.

Git – это программное обеспечение, свободно распространяемое на условиях универсальной общедоступной лицензии GNU версии 2.

Каждый рабочий каталог в Git – это полноценный репозиторий, содержащий всю историю проекта с возможностью отслеживания версий, не зависящий от доступа к сети или центральному серверу.

В процессе разработки используются следующие окружения:

- Development - окружение разработки, используется программистами;
- Test - окружение отладки функционала, интеграционное тестирование, согласно требований;
- Stage - окружение отладки всей системы, проведение регрессионного тестирования;

- Production - продуктивное окружение, использование конечным пользователем.

Git позволяет быстро создавать и осуществлять слияние отдельных ветвей проекта и включает в себя специальные инструменты для визуализации и навигации по нелинейной истории разработки. Основным принципом в Git является предположение, что слияние изменения будет производиться чаще, чем его написание, так как оно распределяется для оценки несколькими разработчиками.

ПО разрабатывается несколькими разработчиками одновременно. Когда разработчик начинает внедрение нового функционала или отладку, он забирает последнюю версию проекта из системы Git и разрабатывает функционал в Development окружении. Прежде чем залить новую версию в Test окружение, менеджер версий ПО направляет измененное ПО на проверку другим разработчикам, выбранным произвольно. После одобрения кода ПО отправляется на слияние. Менеджер версий ПО пытается совместить новые изменения с обновленными ветвями проекта. При неудачном слиянии новый код отправляется разработчикам, чтобы они одобрили общее обновление. При успешном слиянии новая ветвь отправляется на Test окружение для его последующего тестирования тестировщиками.

Интеграция новой версии ПО в продуктивную среду

На последнем этапе производится релиз новой версии системы в продуктивную среду с последующим приемочным тестированием заказчиком, согласно предъявляемым требованиям к данной версии ПО.

Тестирование

При тестировании ПО используется как автоматическое, так и ручное тестирование.

Автоматическое тестирование

Используется методология TDD (test-driven development или процесс разработки через тестирование), при которой разработчиками, на этапе разработки, покрывается автоматическими тестами разрабатываемый функционал модульно. Основная цель данного метода - достичь максимального покрытия всего функционала, оперативно обнаружить ошибки при внедрении новых функций. Это позволяет придерживаться принципам развертывания CI/CD (Continuous Integration/Continuous Delivery). Кроме того используется внешняя система, разработанная специалистами QA компании, позволяющая в автоматическом режиме проверить работу модулей ПО между собой (Интеграционное тестирование)

Ручное тестирование

Используется система управления тестированием, что позволяет хранить все тестовые сценарии в едином месте, а также планировать и составлять тестовые планы по тестированию ПО. При тестировании ПО функционал вначале тестируется изолированно на Test окружении, при успешном прохождении тестовых сценариев на Test окружении проводится регрессионное тестирование всего функционала ПО на Stage окружении.

Такой подход к тестированию позволяет минимизировать риски и обеспечить предоставление пользователям стабильно работающее ПО на продуктовой среде.

Персонал

В процессах разработки, поддержки, модернизации, сопровождения, гарантийного обслуживания и устранения неисправностей “Платформы Jetskills” задействована команда “ООО Унивеб” численностью 20 человек:

Должность	Кол-во	Ключевые навыки
Технический директор	1	создание и реализация ИТ-стратегии по развитию команды, инфраструктуры и процессов; оптимизация архитектуры и инфраструктуры компании.
Руководитель проекта	1	взаимодействие с отделом разработки и тестирования; сбор потребностей и формирование задач; разработка ТЗ, календарных планов; описание бизнес-процессов; контроль сроков исполнения задач (как бизнеса, так и команд разработчиков); распределение нагрузки между участниками команды с учетом приоритетов и компетенций, формирование релизов.
Архитектор программного обеспечения	1	понимание аспектов информационной безопасности разрабатываемых решений; опыт проектирования архитектуры сложных решений;

		знание и опыт практического применения технологий: баз данных, серверов приложений, брокеров сообщений и интеграционных шин, и других.
Дизайнер	2	проектирование сценариев взаимодействия пользователя и продукта; генерация дизайнерских решений; проектирование интерфейсов; осуществление комплексного анализа интерфейсов; налаживание взаимодействия в команде при работе над интерфейсом;
Backend разработчик	4	реализация новой функциональности продукта; поддержка существующего кода; анализ требований, обсуждение, оценка и декомпозиция задач; участие в проработке архитектурных решений; проведение Code Review;
Frontend разработчик	4	реализация новой функциональности продукта; поддержка существующего кода; анализ требований, обсуждение, оценка и декомпозиция задач; участие в проработке архитектурных решений; проведение Code Review;
Руководитель отдела QA	1	планирование, организация и ведение процесса функционального тестирования; анализ и оптимизация существующих процессов функционального тестирования; внедрение новых технологий и практик в процесс функционального тестирования; организация взаимодействия с другими подразделениями компании; обучение и развитие сотрудников; автоматизация текущей деятельности (<u>инструменты</u> тестирования, автотестирование); подготовка отчетности и сбор метрик.
QA инженер	1	разработка сценариев тестирования, адаптация и обновление имеющихся тест-кейсов для автоматизации; автоматизированное тестирование API back-end (Jmeter, Java);

		<p>формирование отчетов по результатам проведенного тестирования;</p> <p>внедрение процессов CI/CD в автоматизированное тестирование;</p> <p>составление багов в системе управления проектами и задачами;</p> <p>ручное тестирование (при необходимости).</p>
Manual QA	3	<p>ручное тестирование UI;</p> <p>анализ документации;</p> <p>разработка и поддержка тест-кейсов;</p> <p>взаимодействие с командой разработчиков;</p> <p>оперативное составление баг-репортов.</p>
Специалист технической поддержки	2	<p>обработка обращений пользователей;</p> <p>ведение базы знаний по обращениям;</p> <p>формирование инструкций и FAQ;</p>

Фактический почтовый адрес центрального офиса:

г.Москва, м. Кропоткинская, Соимоновский проезд, дом 7, строение 1, подъезд № 2

Телефон для связи: +7 (499) 643 82 01

Почта: support@jetskills.ru

Обучение и квалификация персонала

С целью повышения уровня теоретических знаний, совершенствования практических навыков и умений сотрудников организации проводится регулярное повышение квалификации сотрудников. Как с использованием сил кампании (обучающие внутренние вебинары, использование материалов JetSkills), так и внешних ресурсов (прохождение курсов, посещение конференций).

Техническая поддержка

Контакты технической поддержки: support@jetskills.ru

Режим работы технической поддержки: с 10:00 до 18:00 (Московское время)
по будним дням

Пользователи могут отправить заявку на электронную почту технической поддержки по адресу support@jetskills.ru. По факту получения письма

автоматически формируется задача в системе управления проектами и задачами, которая назначается на специалиста технической поддержки. Дальнейшая работа включает в себя следующие пункты:

- 1) Специалист технической поддержки анализирует проблему задачи по ее влиянию на систему и интерфейсные связи по следующим аспектам:
 - a) типу (корректировка/ошибка пользователя, модернизация, ошибка работы ПО, профилактика или адаптация к новым условиям);
 - b) объему (размеру изменения, времени на реализацию изменения);
 - c) критичности (возможность работы с ПО, влиянию на производительность, безопасность или защиту).
- 2) Специалист технической поддержки воспроизводит или верифицирует возникшую проблему, если информации недостаточно для воспроизведения и верификации, специалист технической поддержки связывается с пользователем и запрашивает дополнительную информацию.
- 3) На основе проведенного анализа специалист технической поддержки разрабатывает варианты реализации оказания поддержки пользователю:
 - a) проблема на стороне пользователя - оказывается помощь пользователю в устранении ошибки;
 - b) проблема на стороне ПО и уровень критичности Blocker (Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с ПО или ее ключевыми функциями становится невозможна) - задача переводится в соответствующий статус, после чего разработчиком производится документирование исправлений в hotfix версии и последующий релиз этой версии на Production окружение;
 - c) проблема на стороне ПО и уровень критичности Critical (Ошибка при которой не правильно работает ключевая бизнес логика, проблема приведшая к временному падению сервера или приводящая в нерабочее состояние некоторую часть ПО) - задача переводится в соответствующий статус, после чего разработчиком производится документирование исправлений в critical версии, которые вначале проходят тестирование на Stage окружении и только при успешном тестировании релиз этой версии публикуется на Production окружении.
 - d) проблема на стороне ПО и уровень критичности Hot Feature (ПО необходима критическая доработка, которая улучшит и сделает ПО более доступным) - задача переводится в соответствующий статус, после чего проходит все стадии разработки ПО в плановом порядке.

Совершенствование и модернизация ПО

Совершенствование и модернизация программы происходит путем доработки интерфейса и внедрения новых функций. Увеличивается производительность

программы, и минимизируется время на взаимодействия между органами управления интерфейса всего комплекса и пользователем.

Фактическое улучшение всей программы отображается в виде изменения младших разрядов номера версии ПО с учетом доработок.

Пользователь может самостоятельно повлиять на совершенствование продукта. Для этого необходимо направить техническое предложение на адрес электронной почты support@jetskills.ru. Предложение будет рассмотрено, в случае признания его эффективности будет добавлено в план разработки и соответствующие изменения появятся в ПО.

Сообщения о неисправности

В процессе эксплуатации программного обеспечения могут возникнуть сообщения о неисправности. В случае их возникновения необходимо осуществить процедуру передачи информации о характере ошибки в службу технической поддержки. Обращение можно оставить посредством отправки сообщения на электронную почту support@jetskills.ru.

Устранением неисправностей занимается квалифицированный персонал компании ООО «Унивеб» в количестве 20 человек (см. п. Персонал)

Эксплуатация

Система эксплуатируется в установленной для нее эксплуатационной среде в соответствии с документацией пользователя. Система доступна в сети интернет по адресу - <https://platform.jetskills.ru/>.