



Приложение "Galaxy Chat" ("Чат Galaxy")

Жизненный цикл программного обеспечения

Аннотация

Документ содержит описание жизненного цикла приложения "Galaxy Chat" ("Чат Galaxy").

Данный документ содержит информацию актуальную на момент составления. ООО "Мобилфон" не гарантирует отсутствие ошибок и неточностей и оставляет за собой право вносить изменения и правки в данный документ в случае возникновения необходимости.

Приложение "Galaxy Chat" ("Чат Galaxy") является интеллектуальной собственностью ООО "Мобилфон".

Содержание

Аннотация	2
Содержание	3
Этапы жизненного цикла программного обеспечения	4
Планирование	5
Реализация	6
Тестирование	7
Документирование	8
Внедрение	10
Сопровождение	11

Этапы жизненного цикла программного обеспечения

К основным этапам жизненного цикла разработки приложения "Galaxy Chat" ("Чат Galaxy") относятся:

- Планирование;
- Реализация;
- Тестирование;
- Документирование;
- Внедрение;
- Сопровождение.

Разработка программного обеспечения (ПО) выполняется циклами. Длительность цикла разработки может варьироваться от недели до нескольких недель и зависит от индивидуальных предпочтений команд разработки, оценок эффективности и производительности, объемов и сложности текущих задач.

Гибкий подход к разработке на основе циклов позволяет поддерживать баланс между меняющимися требованиями рынка программного обеспечения, запросами пользователей и качеством программного обеспечения. Данный подход также даёт возможность минимизировать зависимости между разными командами разработки, при этом зависимости могут возникать при реализации задач, которые требуют разработки и изменения интерфейсов приложения, либо при реализации серверных задач, которые требуют внедрения новой функциональности на клиентском приложении.

Техническая поддержка, модернизация ПО, устранение ошибок и обеспечение выполнения всех этапов жизненного цикла ПО осуществляется силами штатных сотрудников в количестве 44 человек.

Планирование

Планирование осуществляется руководителем проекта совместно с руководителями отделов-инициаторов задач при учете экспертизы отделов разработки. При оценке приоритетности задачи учитываются следующие факторы:

- Ожидаемый эффект на пользовательскую аудиторию;
- Техническая сложность задачи;
- Текущая загруженность команд разработки;
- Соответствие долгосрочной стратегии развития приложения.

Для каждой разрабатываемой задачи формируется техническое задание, которое содержит:

- Описание требований к реализуемой функциональности;
- Макеты пользовательских интерфейсов и/или описание логики взаимодействия пользователей с интерфейсом;
- Список команд, участвующих в разработке;
- Оценку количества требуемых циклов разработки;
- Описание требуемых тестов;
- Описание протокола приёмки;
- Ожидаемый срок внедрения изменений.

После формирования технического задания задача отправляется на планирование индивидуальной команды разработки, которая ответственна за разработку данной задачи. На планировании определяется круг лиц непосредственно ответственных за реализацию задач, устанавливается очерёдность выполнения задач в рамках текущего цикла разработки и, в случае необходимости, вносятся изменения и уточнения в техническое задание.

Реализация

Этап реализации включает в себя следующие шаги:

- Обсуждение различных вариантов реализации;
- Консультации с непосредственным постановщиком задачи;
- Разработку дизайна;
- Кодирование.

Тестирование

В зависимости от задачи после реализации могут проводиться различные типы тестирования. Минимальным базовым набором тестов является:

- Выполнение ручного тестирования командой контроля качества на предмет соответствия техническому заданию и отсутствия ошибок реализации;
- Запуск юнит тестирования различных компонентов программного обеспечения;
- Запуск программ контроля качества разрабатываемого кода;
- Запуск регрессионного тестирования.

В случае необходимости дополнительно может быть проведено интеграционное тестирование с целью оценки корректности взаимодействия различных компонент программного обеспечения.

Также при реализации крупных изменений в программном обеспечении может быть проведено нагрузочное тестирование с целью оценить влияние новой функциональности на общую производительность программного обеспечения.

При реализации новой функциональности команда разработчиков также разрабатывает новые тесты.

В случае выявления ошибок в ранее написанном коде, создаются новые наборы тестов для выполнения регрессионного тестирования.

Документирование

Этап документирования может включать в себя несколько различных этапов:

- Регистрация изменений в реестре изменений;
- Документирование изменений во внутренней базе знаний, обеспечивающей работу сотрудников поддержки приложения;
- Документирование изменений во внутренней документации команд разработки;
- Внесение изменений во внутренние стандарты;
- Внесение изменений в руководства для пользователей приложения.

В зависимости от особенностей разрабатываемой функциональности этапы документирования могут пропускаться.

Регистрация изменений в реестре изменений происходит для всех изменений, которые затрагивают логику функционирования приложения, которая непосредственно влияет на поведение пользователей. Как правило, данный этап пропускается в ситуации, когда вносятся инфраструктурные или архитектурные изменения технического характера, направленные на улучшения технического состояния приложения.

Документирование изменений во внутренней базе знаний основной целью преследует повышение качества оказываемой поддержки пользователям. Своевременное и полное документирование во внутренней базе знаний позволяет быстро решать проблемы и затруднения пользователей при работе с приложением, а также оперативно реагировать на возникающие ошибки в работе приложения, что, в свою очередь, непосредственно влияет на уровень удовлетворённости пользователей от использования приложением.

Документирование изменений во внутренней документации команд разработки прежде всего направлено на улучшение процесса разработки, стандартизации подходов к разработке, а также формализации принимаемых архитектурных решений, что увеличивает качество разработки, повышает производительности и способствует быстрой интеграции новых сотрудников в производственные процессы.

Ключевые особенности функционирования приложения, протоколы взаимодействия различных компонентов приложения, дизайн система описаны в виде внутренних стандартов. Данные документы являются ключевыми для описания логики взаимодействия различных элементов приложения, как например протокол взаимодействия клиента с сервером через шифрованное соединения. Все изменения в данных документах строго регламентируются.

Внесение изменения в руководства для пользователей происходит по факту изменения логики функционирования приложения. Наряду с изменением внутренней базы знаний это позволяет повысить качество пользовательского, а также снимает нагрузку с сотрудников службы поддержки, так как пользователь может решить значительную часть своих затруднений после прочтения руководства.

Внедрение

Внедрение изменений происходит в автоматическом режиме после прохождения всех необходимых тестов и принятия решения о выпуске новой функциональности. Небольшие изменения могут быть добавлены в приложения сразу для всех пользователей. Крупные изменения, которые потенциально могут затронуть большое количество пользователей, в случае возникновения проблем внедряются постепенно для части пользователей посредством канареечного развёртывания, что позволяет минимизировать риски снижения качества функционирования приложения.

Сопровождение

Процесс сопровождения является непрерывным после реализации той или иной функциональности. Служба поддержки выполняет сопровождение приложения в части получения обратной связи от пользователей, выявления ошибок в функционировании приложения, приеме заявок на улучшения и изменения приложения.

Далее в случае обнаружения проблем и ошибок с работой приложения информация передаётся в отдел контроля качества, задачей которого является детальное исследование и диагностирование причин неполадок. В случае подтверждения существования проблемы создаётся новая задача на устранение неисправностей, которая назначается к исполнению в очередной цикл разработки. В ситуации, когда ошибка носит критический характер, она устраняется сразу после обнаружения отделом контроля качества силами отдела разработки, который ответственен в данный цикл разработки за устранение текущих ошибок.