

Описания процессов, поддерживающих поддержание жизненного цикла программного обеспечения LegoCar, способов устранения неисправностей, информации о персонале, информации о совершенствовании ПО

1. Процесс разработки и совершенствования программы автоматизации маркетинга «LegoCar»

1.1 Данные о процессе разработки программного продукта LegoCar

Процесс разработки построен по итеративной модели. Проект при этом подходе в каждой фазе развития проходит повторяющийся цикл PDCA: Планирование — Реализация — Проверка — Корректировка (англ. plan-do-check-act cycle). Цель каждой итерации — это получение версии ПО, включающей в себя как новые или переработанные возможности, реализованные в ходе текущей итерации, так и функциональность всех предыдущих итераций. Результат же финальной итерации содержит всю требуемую функциональность продукта.

В связи с большой долей инновационных и исследовательских задач, выбрана методология разработки, которая позволяет в ходе реализации менять состав функций или особенности (детали) их реализации, если это способствует достижению бизнес-результата. Реализуемый программный продукт разбит на модули, разработка ведётся циклами (релизами) по 4 недели, в течение которых происходит разработка новых функциональных возможностей в одном или нескольких модулях программного продукта, а также работа над исправлением возникающих технических ошибок.

Процесс разработки программного обеспечения LegoCar включает в себя следующие основные этапы:

1 Этап - проектирование решения

В рамках этапа формируется техническое задание на доработку нового функционала по запросу клиентов или на основе внутренних потребностей команды сопровождения продукта, где расписывается способ решения поставленной задачи. Требования программному продукту и к особенностям разработки программного обеспечения (ПО) определяются на данном этапе. На основе технического задания формируются задачи для дизайнеров, frontend-разработчиков и backend-разработчиков.

Компетенции: менеджер-проекта, UX/UI дизайнер, директор по продукту, аналитик.

Используемые инструменты: инструменты моделирования информационных систем (например, Diagram.io), инструменты документирования (например, Bitrix24), система управления задачами и проектами (YouTrack).

Результат этапа: техническое задание на разработку релиза в системе управления задачами и проектами.

2 Этап - программная реализация

В рамках данного этапа разрабатывается адаптивный дизайн пользовательского интерфейса, вёрстка графического интерфейса и написание программного кода, обеспечивающего выполнение запланированных в релизе функций.

В процессе разработки ПО в качестве распределенной системы управления версиями исходного кода используется инструмент 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.

Концепция Git возникла на основе опыта, полученного при управлении большим распределенным проектом разработки при работе над Linux, а также особенностей работы с файловыми системами, и острой потребности в создании жизнеспособной системы в кратчайшие сроки. Все это привело к следующим особенностям реализации: 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. Ветви в Git занимают очень мало места: они являются лишь ссылками на отдельные коммиты. С помощью родительского коммита может быть построена полная структура ветвей.

Git предоставляет каждому разработчику локальную копию всей истории разработки, и изменения копируются из одного такого репозитория в другой. Такие изменения импортируются в виде дополнительных ветвей разработки, и их слияние может осуществляться так же, как и для ветви, разработанной локально.

Репозитории могут быть опубликованы с помощью HTTP, FTP, rsync или протокола Git через обычный сокет, ssh или HTTP. Git также имеет эмулятор сервера CVS (системы одновременных версий), который позволяет использовать существующие клиенты CVS и плагины IDE для доступа к репозиториям Git. Репозитории подверсий и SVK можно использовать напрямую с помощью git-svn.

История хранится в Git таким образом, что идентификатор конкретной версии («коммит» в терминологии Git) зависит от полной истории разработки, предшествовавшей данному коммиту. После его публикации невозможно изменить старые версии незаметно. Эта структура похожа на дерево хешей, но с наличием дополнительных данных в узлах и листовых вершинах.

ПО разрабатывается несколькими специалистами. Когда разработчик начинает внедрение нового функционала или отладку, он забирает последнюю версию проекта из системы Git. После выполнения задачи новая версия ПО хранится на его локальном хосте.

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 ПО, чтобы они одобрили общее обновление.

Компетенции: менеджер проекта, директор по продукту, фронтенд-разработчик, бэкенд-разработчик, верстальщик, UX/UI дизайнер.

Используемые инструменты: система контроля версий (например, Gitlab), среда разработки (например, PhpStorm, Pycharm), графический редактор (например, Figma).

Результат этапа: релиз продукта готовый к тестированию и отладке работы.

3 Этап - тестирование и отладка

В рамках этапа тестирования происходит подготовка релиза программы к публикации, исправление ошибок, наполнение контентом, внесение изменений в работу функций, если это необходимо для получения наилучшего результата.

При успешном слиянии новая на этапе разработки ветвь в GIT отправляется на тестирование. Тестировщики, менеджеры проекта или специалисты отдела сопровождения проверяют обновленное ПО разными методами, и при успешном прохождении тестов ветвь разработки подготавливается к релизу.

Также ПО проверяется в тестовой среде. Если все тесты и компиляция проходят успешно, версия считается готовой к релизу. В противном случае менеджер версий ПО находит проблему и возвращает ветвь разработчикам для исправления.

Компетенции: менеджер проекта, директор по продукту, фронтенд-разработчик, бэкенд-разработчик, верстальщик, UX/UI дизайнер, специалист отдела сопровождения.

Используемые инструменты: система контроля версий (например, gitlab), среда разработки (например, visual studio), инструменты автоматизированного тестирования (например, selenium).

Результат этапа: дистрибутив программного продукта, готовый к внедрению и эксплуатации.

4 Этап - внедрение

На данном этапе релиз программы (или отдельного модуля) передаётся в промышленную эксплуатацию путём разворачивания новой версии программы (модуля) в облачной инфраструктуре (SaaS).

Компетенции: аналитик, менеджер проекта, директор по продукту, DevOps инженер.

Используемые инструменты: система контроля версий (например, Gitlab), инструменты автоматизированной выгрузки изменений программного продукта (например, GitLab CI).

Результат этапа: программный продукт опубликован в общем доступе для использования клиентами.

1.2. Данные о процессах совершенствования (модернизации) программного продукта LegoCar

Совершенствование и модернизация программы происходит путем доработки интерфейса и внедрения новых функций. Цель совершенствования программного продукта заключается в непрерывном улучшении текущих показателей эффективности: скорость работы программных модулей, удобство и скорость формирования коммерческих предложений, корректировки для соответствия стандартам поисковых систем и т.д.

Совершенствование ПО автоматизации маркетинга LegoCar начинается со сбора требований для улучшения текущего функционала или разработки нового.

Процесс совершенствования ПО инициируется или согласно плановому развитию, утвержденному стратегией развития ПО, или на основе обратной связи и пожеланий клиентов ООО «Легокар», которые используют ПО в своей деятельности.

В плановом развитии ПО выделяются следующие этапы:

- выявление, анализ и исправление ошибок в функционировании текущей программной версии продукта с помощью внутреннего тестирования ПО или при входящем обращении от клиента;

- добавление нового функционала с учётом ситуации на рынке или в основных бизнес-процессах потребителей продукта, а также на основе требований третьих лиц, которые прямо или косвенно влияют на маркетинговую деятельность клиентов ООО «Легокар»;

- определение и фиксация новых аналитических метрик для подтверждения выдвинутых гипотез в ходе формирования требования для развития продукта;
- обновление справочной документации.

В модернизации ПО, основанной на итогах анализа обратной связи от клиентов ООО «Легокар», можно выделить следующие этапы:

- сбор заявок на устранение выявленных недочётов или предложений по совершенствованию текущей функциональности;
- аналитика обратной связи. После проведения аналитики определяется целесообразность модернизации на основе обратной связи с учётом планового развития;
- обратная связь, принятая по аналитическим соображениям и внесенная в план развития, проходит все этапы внедрения и демонстрируется на тестовом стенде ПО;
- при успешной проверке новых функций обновления размещаются в продуктивной среде;
- на финальном этапе обновлений осуществляется доработка и обновление справочной информации.

Обращаем внимание, что учет обратной связи клиентов ООО «Легокар» не является заказной разработкой программного обеспечения и не требует дополнительной оплаты от клиента. Клиенты вправе направить предложения о совершенствовании продукта, а ООО «Легокар» вправе рассмотреть предложения, но ООО «Легокар» самостоятельно принимает решение о целесообразности или нецелесообразности его учета и производстве обновлений. Для каждой части требований выполняются все необходимые процессы, работы и задачи, указанные в п.1.1. настоящего документа. Например, анализ требований, создание или изменения архитектуры, разработка программного кода, проведение тестирования (функционального, регрессионного, нагрузочного).

1.3. Данные о команде разработки

№	Квалификация	Количество
1	Ведущий программист	1
2	Программист	4
3	Менеджер-проекта	1
4	Верстальщик	1
5	UX-UI дизайнер	1
6	Тестировщик	1
7	DevOps инженер	1
8	Директор по продукту	1
9	Ведущий специалист по маркетингу	1
ИТОГО		12

1.4. Адрес проведения работ и контакты

Фактический адрес проведения работ по разработке ПО: 460507 Оренбургская область, м.р-н Оренбургский, с.п. Пригородный Сельсовет, п Пригородный, ш Нежинское, дом 22

Контактный телефон разработчиков: 8 800 600 19 67,

Электронная почта: info@lego-car.ru

2 Процесс поддержки Программы автоматизации маркетинга «LegoCar»

Для продукта предусмотрена техническая поддержка. Поддержку осуществляет правообладатель – ООО «Легокар» своими силами, а также при необходимости с привлечением третьих лиц, находящихся на территории Российской Федерации.

2.1. График работы технической поддержки

Запросы принимаются круглосуточно 7 дней в неделю на электронную почту support@lego-car.ru.

Период обработки обращений клиентов с 07:00 до 17:30 (время МСК) в рабочие дни. Если запрос поступил вне указанного времени работы, то их обработка начнется с момента начала рабочего дня. Критичные запросы обрабатываются дежурными специалистами также вне рабочих часов по возможности.

2.2. Регламент работы технической поддержки

В процессе тестирования и эксплуатации ПО могут возникнуть сообщения о неисправности. В случае их возникновения необходимо осуществить процедуру передачи информации о характере ошибки в службу технической поддержки ООО «Легокар».

Для оформления заявки на устранения неисправности можно воспользоваться следующими средствами связи:

- оставить запрос на электронную почту support@lego-car.ru;
- связаться по общему телефону 8 800 600 19 67;
- связаться с персональным менеджером, закрепленным за клиентом, по рабочим контактам по телефону, в мессенджерах или по почте на домене @lego-car.ru.

Порядок приёма и обработки обращений

Все обращения принимает специалист отдела сопровождения, который квалифицирует обращение и передает в соответствующий отдел для анализа, реализации или исправления технической ошибки.

Каждый запрос проходит следующие этапы:

1 Классификация обращения и определение проблемы

Запросы квалифицируются на следующие типы:

- консультация. Запрос решается специалистом отдела сопровождения путем предоставления консультации по пользованию ПО в устном или письменном виде, а при необходимости с привлечением средств видео-конференций;
- ошибка. Запрос передается менеджеру проекта для занесения в систему управления задачи и дальнейшего решения силами отдела разработки;
- предложение. Запрос передается директору продукта для рассмотрения и принятия решения о целесообразности включения данного предложения в план развития проекта. Также клиенту могут быть предложены альтернативные решения.

Обращения на техническую поддержку должно соответствовать следующим требованиям:

- в теме Обращения должно быть краткое описание запроса;

- в теле Обращения должно быть подробное описание запроса, информация о пользователе (ФИО, контакты, по которым связаться для уточнения информации или сообщить о решении проблемы);

- в случае Ошибки, в зависимости от характера ошибки, необходимо предоставить максимально полную информацию, которая может включать в себя описание процедуры, во время которой произошла Ошибка, сообщение об Ошибке дословно (приложить скриншот с ошибкой), либо фактический результат описанных действий, расходящийся с ожиданием, действие пользователя, вызывающее Ошибку, описание операционных систем, на которых возникает неполадка.

2 Поиск и предоставление решения (опционально)

Технические ошибки передаются в отдел разработки после того, как менеджер проекта проанализирует проблему и составит техническое задание для специалистов отдела разработки.

Для разрешения Обращения может быть использован тот тип коммуникаций, который обеспечит наилучший результат: удаленное подключение к компьютеру пользователя (при наличии технической возможности), телефонный звонок или отправка сообщения электронной почты.

В результате данного этапа проходит процесс обновления программного кода и оповещение клиента о решении его запроса. В случае, если от клиента не поступает дополнительных данных по этой ошибке или сообщения о том, что проблема не решена, то запрос закрывается. В случае, если клиент сообщает, что проблема так и не решена, то запрос отправляется на доработку в отдел разработки менеджером проекта.

2.3. Данные о команде сопровождения

№	Квалификация	Количество
1	Программист	2
2	Менеджер-проекта	1
3	Специалист отдела сопровождения	5
4	Контент-менеджер	3
ИТОГО		11

2.4. Адрес проведения работ по сопровождению и контакты

Фактический адрес проведения работ по технической поддержке ПО: 460507 Оренбургская область, м.р-н Оренбургский, с.п. Пригородный Сельсовет, п Пригородный, ш Нежинское, дом 22

Контактный телефон: 8 800 600 19 67,

Электронная почта: support@lego-car.ru