ООО «3В Сервис»
РФ, 127051, Москва, ул. Трубная 25 стр 1 офис 6
Тел./ф (495) 221-22-53
www.3v-services.com

Утверждаю
генеральный директор
ООО «3В Сервис»

_____ Петухов В.Н.

Среда динамического моделирования технических систем SimInTech™

План обеспечения качества программного обеспечения

Модуль генерации кода для систем реального времени

ШИФР ГК16ОК

Москва, 2016

Аннотация

В данном программном документе приведено описание мероприятий по обеспечению качества разработки программного обеспечения.

Указаны основные этапы разработки и верификации ПО.

СОДЕРАЖНИЕ

Ведение

Программа для ЭВМ “Среда динамического моделирования SimInTech”, (сокращенное название «SimInTech»), свидетельство о регистрации №2010617758 - современная среда интеллектуальной системы автоматизированного **проектирования** (САПР), предназначенная для детального исследования и анализа нестационарных процессов в системах автоматического управления, в следящих приводах и роботах, в любых технических системах, описание динамики которых может быть реализовано методами структурного моделирования.

SimInTech обеспечивает создание алгоритмов управления в виде функционально-блочных диаграмм. ПО содержит в себе математическое ядро для проведения динамического расчета созданного алгоритма управления путем задания входных воздействий и анализа изменений внутренних параметров и выходных значений во время моделирования.

SimInTech является базовым программным обеспечением для разрабатываемого модуля. Комплексная система моделирования систем управления и программирования приборов включает в себя:

- модуль генерации кода для автоматической генерации исходных кодов и исполняемых модулей;
- среду разработки для проектирования алгоритмов управления в виде наглядной функционально-блочной диаграммы;
- систему исполнения программ для приборов, для выполнения сгенерированных при помощи генератора кода исполняемых модулей на приборах.

Под прибором следует понимать программируемые логические контроллеры (ПЛК) в составе промышленных компьютеров, работающие под управление POSIX-совместимых операционных систем реального времени.

Предметом данного плана качества является разработка модуля для SimInTech, обеспечивающего автоматическое создание программ на языке Си для исполнения в системах управления реального времени.

1. Основание для разработки

Основанием для разработки является:

Договор N 437 - 01 от 26.07.2013 по теме: «Разработка программного обеспечения верхнего уровня программно-технического комплекта средств автоматического управления».

Заказчик ООО «Московский завод «ФИЗПРИБОР».

План особо важных работ по доработке программного обеспечения на 2015 год. Утвержден 15.02.2013. ООО «3В Сервис».

2. Термины и определения

2.1 Платформа – Среда динамического моделирования технических систем SimInTech™ среда разработки программного обеспечения на предметно-ориентированном языке программирования.

2.2 ПО – разрабатываемое программное обеспечение «Модуль генерации кода для систем реального времени»

2.3 Заказчик – Общество с ограниченной ответственностью «Московский завод «Физприбор»

2.4 Компания – Общество с ограниченной ответственностью «3В Сервис», ООО «3ВС»

2.5 Руководитель (Компании) – Генеральный директор Компании согласно Уставу

2.6 Администрация (Компании) – Генеральный директор и лица, относящиеся к Администрации в соответствии со штатным расписанием

2.7 Проект – проект разработки и создания ПО

Участники проекта – специалисты Компании и Заказчика, вовлеченные в Проект

Руководство по качеству – Руководство по качеству Компании РК СМК 01-2016

3. Общие принципы обеспечения качества ООО «3ВС»

ООО «3ВС» создано в соответствии с Гражданским кодексом Российской Федерации, Федеральным законом «Об открытых акционерных обществах» и иными законодательными и нормативными правовыми актами Российской Федерации.

Средством достижения высокого уровня осуществляемой ООО «3ВС» деятельности» является разработанная, документально оформленная, внедренная и поддерживаемая в рабочем состоянии система менеджмента качества (СМК), соответствующая требованиям международных стандартов ISO серии 9000. Системы качества являются эффективным инструментом для достижения главной цели любой организации – удовлетворения требований потребителя.

Непрерывный процесс обеспечения качества создается посредством разработки и выполнения проектных перспективных, текущих и оперативных планов по качеству, в которых предусматривается совершенствование имеющихся технологических процессов и методик выполнения работ, корректировка действующих и разработка новых нормативных документов СМК при выполнении работ и оказании услуг на новых объектах.

Документирование и анализ несоответствий, материальное и моральное стимулирование за обеспечение и повышение качества позволяет повысить ответственность и заинтересованность руководителей, специалистов и исполнителей в высоком качестве выполняемых работ.

Гарантия качества продуктов.

Непрерывный контроль качества реализуется с помощью тестирования. Данный процесс предполагает создание тестов для каждого ключевого сценария, реализуемого в системе. Качество системы проявляется, прежде всего, в количестве успешных и неуспешных сценариев, что как раз и выявляется в процессе тестирования. Тестирование и разработка новых тестовых сценариев проводятся на каждой итерации проекта. Наборы сценариев и программных скриптов дорабатываются итеративно вместе с создаваемым продуктом.

Непрерывный контроль качества приводит к следующим позитивным результатам:

- оценка состояния проекта приобретает большую объективность, т. к. оценивается реальное функционирование системы, а не качество проектной документации;
- оценка проекта позволяет раскрыть несоответствия в требованиях, моделях и реализации;
- тестирование акцентирует внимание на тех сторонах работы системы, которые имеют наибольшую важность и повышенный риск;
- дефекты выявляются на ранних стадиях, что снижает затраты на их устранение;
- автоматизированное тестирование обеспечивает высокий уровень функциональности системы, надежности и производительности.

4. Необходимые ресурсы

Администрация Компании постоянно решает вопросы, связанные с обеспечением ресурсами всех видов деятельности, включая разработку ПО.

К указанным ресурсам относятся:

- человеческие ресурсы;
- вычислительная техника;
- специальное программное обеспечение;
- инфраструктура.

Ресурсная база подвергается постоянной оценке на достаточность и адекватность стратегическим целям и может быть изменена и улучшена под конкретные проекты. Все орг.-штатные единицы Компании обеспечены квалифицированным персоналом, соответствующим оборудованием, программным обеспечением и необходимыми производственными условиями. Планы по ресурсообеспечению разрабатываются Руководителем Компании на год в соответствие с потребностями подразделений и перспективным планом развития.

4.1 Человеческие ресурсы

Для разработки ПС были задействованы специалисты, сотрудники Компании, с опытом работы в сфере разработки программного обеспечения не менее 5 лет.

Указанные специалисты участвовали в предыдущих проектах Компании по разработке основных архитектурных блоков и модулей Платформы. Таким образом, при разработке ПО обеспечена преемственность опыта работы, что исключило, так называемые ошибки «малого возраста» и позволило в полной мере использовать наработанный положительный опыт эксплуатации Платформы.

Все штатные сотрудники Компании – специалисты с высшим техническим образованием, имеющие большой опыт работы в соответствующих профессиональных областях. Задействованный в процессе создания ПС персонал не реже 1 раза в год проходит программы и курсы повышения квалификации или специальные курсы, позволяющие поддерживать уровень профессиональных знаний на соответствующем передовым знаниям и практикам уровне. В том числе, Администрация Компании выделяет ресурсы на проведение

внутренних циклов повышения квалификации, которые проводятся на базе Компании приглашенными специалистами и внешние циклы повышения квалификации, которые проводятся на базе ведущих образовательных учреждений повышения квалификации.

Планирование повышения квалификации программистов осуществляет Руководитель Компании ежегодно.

По итогам сбора и анализа требований Заказчика по содержанию и срокам, предъявляемым к проекту разработки ПС, а также анализа государственной нормативной документации в сфере регулирования соответствующей деятельности, была проведена оценка потребности в человеческих ресурсах проекта разработки ПС. По результатам оценки, с целью реализации проекта в установленные сроки, была сформирована проектная команда из состава сотрудников Компании, которая состоит из:

Специализация	**Количество человек**
Руководитель проекта	1
Аналитик	1
Ведущий разработчик	2
Разработчик	2
Специалист по тестированию	2
Технический писатель	1

4.2 Вычислительная техника

Для разработки ПО Использовались персональные компьютеры с следующей конфигурацией:

- Процессор Intel с частотой не менее 1500 МГц
- Оперативная память не менее 4 Гб
- Жесткий диск не менее 500 Гб

В для групповой работы используется 3 сервера распложённых в дата центре и содержащий набор необходимых виртуальных машин. Конфигурация серверов:

- Процессор 4-ядерный Intel Xeon E5-2623V4
- Оперативная память 32 Гб DDR-4
- Массив жестких дисков 32 Тб

В для групповой работы используется ц сервера распложённых в офисе дата центре и содержащий набор необходимых виртуальных машин. Конфигурация серверов в офиссе:

- Процессор 4-ядерный Intel Xeon E5-2623V4
- Оперативная память 32 Гб DDR-4
- Массив жестких дисков 8 Тб

4.3 Специальное программное обеспечение

Для разработки ПО используется набор специальных программного обеспечения - инструментальных средств для разработки. Средства разработки включает себя:

Delphi XE7 – среда разработки программного обеспечения.

Redmine – система управления проетами.

GIT – среда для хранения и управления версиями.

OTRS – среда работы с пользователями ПО.

4.4 Инфраструктура

Офис организации расположен по адресу Трубная 25 к5 офис 6

Офис оборудован рабочими местами, оборудованными с учетом требований безопасности и охраны труда с доступом в интернет, и необходимым офисным оборудованием. В общее оборудование входит:

1) Принтеры, сканер;
2) Локальное серверное хранилище;

Для организации работы используется местная локальная сеть в офисе ООО «3В Сервис» и сеть интернет для организации связи с удаленными рабочими местами.

Управление деятельностью:

Руководство ООО «3ВС» определяет потребности в соответствующей инфраструктуре (здания, производственные помещения, вспомогательное оборудование, транспорт, связь, энергоресурсы), а также поддерживает её рабочем состоянии.

5. Менеджмент разработки ПО

5.1 Жизненный цикл разработки ПО

При разработки любого программного обеспечения в ООО «3В Сервис» используется стандартизированный жизненный цикл, каждый этап которого, документируется и отслеживается.

Жизненный цикл разработки программного обеспечения состоит из следующих этапов:

- Формирование и управление требованиями к ПО
- Разработка
- Тестирование
- Приемочные испытания у заказчика
- Техническая поддержка эксплуатация

На рисунке 1 изображены этапы разработки ПО.

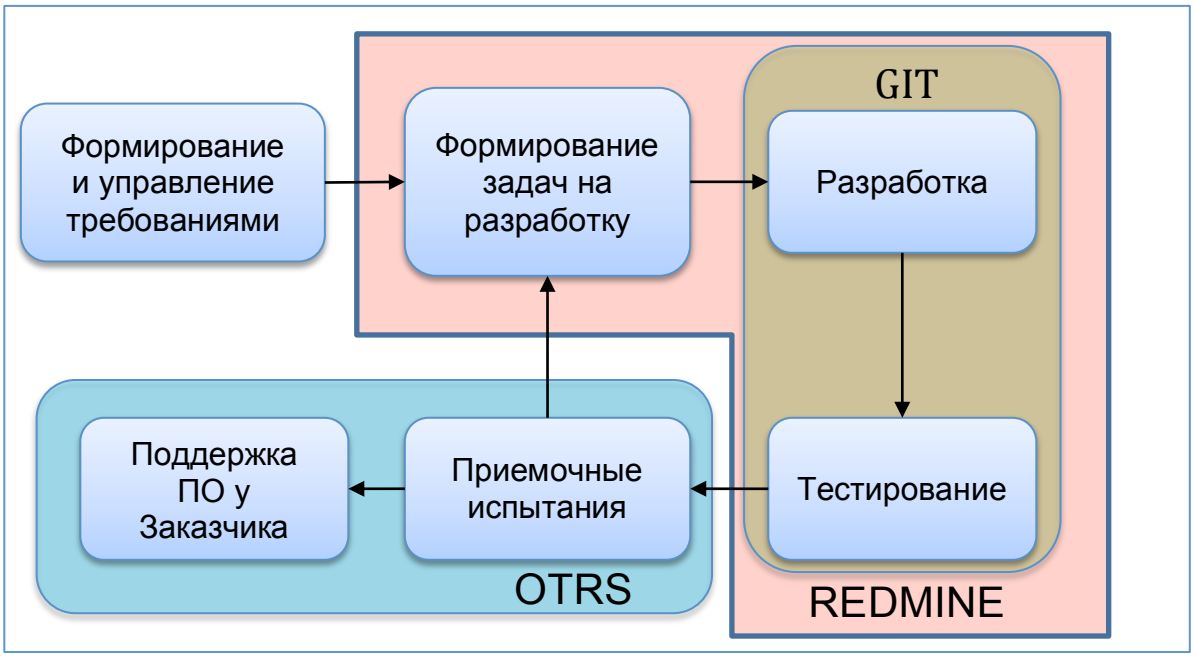

Рисунок 1. Этапы разработки ПО и используемые ресурсы

5.2 Формирование и управление требованиям к ПО

5.2.1 Принципы управления требованиями к ПО

С целью обеспечения качества в проекте разработки ПС реализованы следующие принципы управления требованиями:

Иерархичность требований. Требования формируют древовидную структуру, от верхнеуровневых (общих) требований к никзоуровневым (детальным). Требования структурированы по типу.

Ясность (недвусмысленность, определенность, однозначность спецификаций). Требование обладает свойством ясности, если оно сходным образом воспринимается всеми участниками проекта.

Корректность и согласованность (непротиворечивость). Требования не должны противоречить требованиям своего уровня иерархии и требованиям «родительского» уровня. Так требования пользователей не должны противоречить бизнес-требованиям, а функциональные требования – требованиям пользователя.

Необходимость и полезность при эксплуатации. Необходимыми следует считать свойства, без выполнения которых невозможно, либо затруднено выполнение автоматизированных бизнес-функций пользователей; полезными при эксплуатации следует считать любые свойства, повышающие эргономические качества продукта.

Осуществимость (выполнимость). Выполнимость требования определяется разумным балансом между ценностью (степенью необходимости и полезности) и потребными ресурсами.

Модифицируемость. Обеспечение возможности переработки требований и поддержание истории изменений для каждого положения. Все положения уникально помечены и обозначены. Каждое требование единожды записывается в спецификацию требований. Организовано сохранение спецификации в базе данных инструмента управления требованиями, что делает их пригодными для повторного использования.

Трассируемость. Трассируемость требования определяется возможностью отследить связь между ним и другими артефактами информационной системы (документами, моделями, текстами программ и пр.).

Упорядоченность по важности и стабильности. Приоритет требования представляет собой количественную оценку степени значимости (важности) требования. Стабильность требования характеризует прогнозную оценку неизменности требований во времени.

5.2.2 Требования проекта

В рамках разработки ПО разрабатываются следующие требования:

Наименование требования	**Описание**
Функциональные требования	Данный тип требований включает необходимый набор функциональности, который должен быть реализован программным обеспечением
Требования платформы	Требования к ПО, которые диктуются архитектурной Платформой на базе которой создается ПО. В данном случае это «Среда динамического моделирования технических систем «SimInTech»
Требования к интерфейсу	Требования к интерфейсу ПО
Соответствие законодательным нормам и стандартам	Возможные требования законодательства, под которые попадает разрабатываемое ПО

5.2.3 Этапы формирования и управления требованиям

В рамках проекта рассматриваются следующие этапы управления требованиями:

1 этап анализа на стороне Заказчика, в рамках которого происходит взаимодействие представителей Заказчика и Компании, и происходит фиксация первичных требований;

2 этап анализа на стороне Компании, в рамках происходит уточнение требований к разрабатываемому ПС, исходя из предполагаемой архитектуры ПО и требованиями к «Средой динамического моделирования технических систем «SimInTech»;

3 этап анализа по итогам приемочных испытаний ПО в результате которых Заказчик может внести дополнительные требования к ПО или изменить сформулированные ранее требования.

5.2.4 Процесс управления требованиями

В рамках каждого этапа управления требованиями реализовывается следующий процесс управлениями требования:

- Сбор требований;
- Разработка документов;
- Создание тестовых сценариев;
- Проектирование системы.

5.3 Модель управления жизненным циклом разработки ПО

С целью обеспечения принципа менеджмента качества «Процессный подход» руководства по качеству Компании были принят к реализации «Итерационный подход» к разработке программного обеспечения. Качество разработки также обеспечивается Системой управления разработкой и Системой управления версиями.

5.3.1 Процесс управления проектами

Подход предполагает разбиение жизненного цикла проекта на последовательность итераций, каждая из которых напоминает «мини-проект», включая все процессы разработки в применении к созданию меньших фрагментов функциональности, по сравнению с проектом в целом. Цель каждой итерации — получение работающей версии программной системы, включающей функциональность, определённую интегрированным содержанием всех предыдущих и текущей итерации. Результат финальной итерации содержит всю требуемую функциональность продукта. Таким образом, с завершением каждой итерации продукт получает приращение — инкремент — к его возможностям, которые, следовательно, развиваются эволюционно.

Данный подход позволяет контролировать каждый прирост функционала

5.3.2 Система управления разработкой

В роли Системы управления разработкой выступает Redmine — открытое серверное веб-приложение для управления проектами и задачами, которое предоставляет следующие функциональные возможности:

- ведение нескольких проектов;
- гибкая система доступа, основанная на ролях;
- система отслеживания ошибок;
- диаграммы Ганта и календарь;
- ведение новостей проекта, документов и управление файлами;
- оповещение об изменениях с помощью RSS-потоков и электронной почты;
- вехи для каждого проекта;
- форумы для каждого проекта;
- учёт временных затрат;

- настраиваемые произвольные поля для инцидентов, временных затрат, проектов и пользователей;
- лёгкая интеграция с системами управления версиями (SVN, CVS, Git, Mercurial, Bazaar и Darcs);
- создание записей об ошибках на основе полученных писем;
- поддержка множественной аутентификации LDAP;
- возможность самостоятельной регистрации новых пользователей;
- многоязычный интерфейс (в том числе русский);
- поддержка СУБД MySQL, Microsoft SQL Server, PostgreSQL, SQLite, Oracle.

5.3.3 Система контроля версий

В роли Системы контроля версий выступает GIT – распределённая система управления версиями. Git поддерживает быстрое разделение и слияние версий, включает инструменты для визуализации и навигации по нелинейной истории разработки. Git предоставляет каждому разработчику локальную копию всей истории разработки, изменения копируются из одного репозитория в другой.

Git относится к классу распределенных систем управления версиями. Основные преимущества распределённых систем — их гибкость и значительно бóльшая (по сравнению с централизованными системами) автономия отдельного рабочего места. Каждый компьютер разработчика является, фактически, самостоятельным и полнофункциональным сервером, из таких компьютеров можно построить произвольную по структуре и уровню сложности систему, задав (как техническими, так и административными мерами) желаемый порядок синхронизации. При этом каждый разработчик может вести работу независимо, так, как ему удобно, изменяя и сохраняя промежуточные версии документов, пользуясь всеми возможностями системы (в том числе доступом к истории изменений) даже в отсутствие сетевого соединения с сервером. Связь с сервером или другими разработчиками требуется исключительно для проведения синхронизации, при этом обмен наборами изменений может осуществляться по различным схемам.

5.3.4 Описание процесса разработки ПС

Для разработки программного обеспечения, используются итерационный подход, заключающийся в повторении заранее определенных и документированных последовательностей итераций.

- Формирование списка (бэклога) задач первой итерации на основе спецификации требований. Данный список задач вносится в Систему управления разработкой без назначения на конкретных исполнителей.
- Распределение задач между исполнителями (разработчиками).
- Разработка, фиксация результатов разработки в Системе контроля версий.
- Передача откомпилированных результатов разработки на тестирование, возврат задач в разработку, в случае обнаружения ошибок по итогам тестирования (см. п 5.4.1).
- Передача протестированных результатов первой итерации разработки на приемочное тестирование Заказчику (см. п. 5.4.3).
- Формирование списка (бэклога) задач следующей итерации на основе спецификации требований и результатов приемочного тестирования, которая наращивает функционал разрабатываемого ПС.

При этом жизненный цикл задач в Системе управления разработкой выглядит следующим образом: новая задача – назначение ответственного – разработка ПС, выгрузка результатов в Git – тестирование – сборка – передача результатов Заказчику.

При таком подходе существует возможность отслеживания и обеспечения качества на каждом этапе создания ПС.

При итерационном подходе, ошибки разработки не накапливаются, а выявляются путем тестирования как внутренними силами так и силами Заказчика в рамках пилотного тестирования им результатов работ.

5.4 Тестирование ПС

С целью обеспечения качества результатов разработки применяются следующие виды тестирования.

- Модульное тестирование
- Интеграционное тестирование
- Системное тестирование

5.4.1 Модульное тестирование

Модульное тестирование проводится, для каждой функций ПО созданной в рамках итерационного этапа. Для каждого этапа создается тестовые задачи, которые обеспечивают проверку выполнения созданного функционала ПО. Выполняется анализ работы созданной функции на предмет ее соответствия спецификации требованиям помещенным в систему Redmine, в качестве задачи. В случае если анализ подтверждает соответствие спецификации требований происходит закрытие задачи в Redmain.

В случае если тестирование и (или) анализ показывают несоответствии спецификации требование задача возвращается разработчику с замечаниями и описанием выявленных отклонений от спецификации требований.

5.4.2 Интеграционное тестирование

Интеграционное тестирование в качестве входных данных использует модули, над которыми было проведено модульное тестирование, группирует их в более крупные множества, выполняет тесты, определённые в плане тестирования для этих множеств, и представляет их в качестве выходных данных и входных для последующего системного тестирования.

Целью интеграционного тестирования является проверка соответствия проектируемых единиц функциональным, приёмным и требованиям надежности. Тестирование этих проектируемых единиц — объединения, множества или группы модулей — выполняется через их интерфейс, с использованием тестирования «чёрного ящика».

На этом этапе разрабатываемые модули тестируются в составе среды на базе которой происходит разработка ПО.

5.4.3 Системное тестирование

Системное тестирование – это тестирование ПО, выполняемое на полностью интегрированном ПО, с целью проверки соответствия ПО исходным требованиям.

Также при системном тестировании важно установить, адекватно ли работает ПО при заведомо неправильных действиях пользователя. Для этого применяются методы позитивного и негативного тестирования. Для реализации позитивного тестирования создаются сценарии, эмитирующие заведомо правильные действия пользователя, и изучается реакция системы. В случае негативного тестирования создаются сценарии, эмитирующие заведомо неверные действия пользователя и так же изучается реакция системы. различных функций системы.

5.4.4 Приемочные испытания

Приемочные испытания проводятся с целью получения обратной связи от Заказчика о готовности ПО к промышленной эксплуатации. Для обеспечения обратной связи по итогам приемочного тестирования используется Система взаимодействия с заказчиком.

Приемочные испытания проводятся в две стадии:

- альфа-тестирование – тестирование внутри ООО «3В Сервис» разработчиками ПО и специалистами не включёнными в разработку ПО
- бэта-тестирование – тестирование выполняемые тестировщиками и представителями заказчика.
-

5.4.4.1 Альфа тестирование

Альфа тестирование проводиться разработчиками ПО.

Первый этап альфа-тестирования выполняется разработчиком ПО, с помощью подготовленных сценариев или с помощью среды для быстрого выявления ошибок. В случае выявления ошибок возможен как возврат в доработку, так передача ошибки специалистам по тестированию для дополнительного исследования в среде.

Втрой этап альфа тестирования осуществляется членами команды ООО «3ВС», не задействованными в разработке. При этом пользовали могут использовать альфа версию разрабатываемого ПО как для проверки разрабатываемого функционала так и для решения собственных задач, возможно напрямую не связанных с разрабатываемым функционалом.

На данном этапе проверяется как непосредственно разрабатываемый функционал так и пользовательский интерфейс.

На третьем этапе производиться ad-hoc testing — вид тестирования, который выполняется без подготовки к тестам, без определения ожидаемых результатов, проектирования тестовых сценариев.

5.4.4.2 Бетта тестирование

Отличительной особенностью является то что для него привлекаются добровольные пользователи (потенциальные пользователи) которые проводят испытания в интересах заказчика.

Цель бэтта-тестирования потребителем это - обеспечение необходимого обратной связи, которая дает конечным пользователям даёт перспективу решения возникающих вопросов разработчиками ПО. Пользователям передается версия ПО для опытной эксплуатации.

Пользователи всегда имеют

Для организации конструктивной обратной связи, а также в рамках выполнения требований системы менеджмента качества, по автоматизации деятельности ООО «3ВС» была введена система обработки заявок OTRS.

5.5 Система взаимодействия с заказчиком

В роли системы взаимодействия с заказчиком выступает OTRS – открытая система обработки заявок. Система предоставляет следующий функционал:

- Персональный доступ специалистов заказчика к службе поддержки.
- Формирование заявок на поддержку.
- Взаимодействие с заказчиком через веб-приложение;
- Фиксация заявок на поддержку

OTRS - Open source Ticket Request System, является web приложением по учету (приему и обработке) заявок между руководством компании и потребителями. Система OTRS Создает свой электронный почтовый ящик в котором фиксирует запросы и далее сохраняет их в своей базе данных. создает автоответ для каждого нового запроса и отправляет его заказчику. Для каждого запроса OTRS создает прямую ссылку - номер заявки.

Также, следует отметить тот факт, что ни одно сообщение клиента не будет отредактировано дважды, поскольку система автоматически блокирует заявку, для которой создается ответ.

OTRS успешно решает следующие задачи:

Регистрацию обращений. Сотрудникам необходимо знать, что обращение было; пользователям необходимо понимать, что обращение дошло по адресу и не потеряно. Кроме того, для пользователя удобно, чтобы любые обращения можно было оформить одним и тем же унифицированным способом.

Учет и обработку обращений. Сотрудники должны знать, какие обращения имеются в каждый момент и предпринимать действия для их выполнения; пользователи должны понимать, что происходит с их обращением.

Автоматизацию выполнения заявок. Существует тезис о том, что любая проблема может быть решена компетентным специалистом в течение 1 часа. Нужно только, чтобы проблема сразу попала в руки такого специалиста. Поэтому если автоматизировать обработку поступающих обращений, можно резко сократить путь обращения к соответствующему специалисту.

Человеческий фактор, в меньшей степени оказывает влияние на выполнение заявок потому, что о ней банально знают другие сотрудники, а не только исполнитель. Заказчику

нужно только установить у себя систему OTRS, авторизироваться и направлять замечания и пожелания исполнителю.