

Общество с ограниченной ответственностью «Мионит»



**Описание процессов, обеспечивающих поддержание
жизненного цикла программного обеспечения
«МионитCRM»**

2026 г.

Оглавление

1	Регламент оказания технической поддержки.....	3
1.1	Список терминов.....	3
1.2	Общие положения.....	3
1.3	Режим работы технической поддержки.....	5
1.4	Порядок подачи и обработки обращений.....	5
2	Техническая поддержка в системе Intraservice	9
2.1	Регистрация участников процесса.....	9
2.2	Выбор сервиса и оформление заявок.....	9
2.3	Определение статуса заявки.....	11
2.4	Определение приоритета.....	12
2.5	Работа с инцидентами.....	12
2.6	Работа с проблемами.....	13
2.7	Работа с запросами на изменения	14
2.8	Недокументированное поведение системы и ошибки	14
3	Регламент процесса разработки.....	16
3.1	Термины.....	16
3.2	Участники.....	16
3.3	Инструменты	17
3.4	Процедуры.....	17
3.4.1	Планирование.....	17
3.4.2	Ежедневная работа участников команды.....	18
3.4.3	Ежедневный контроль.....	20
3.4.4	Демонстрация.....	20
3.4.5	Ретроспектива	21
3.5	Организация работы с кодом.....	21
3.5.1	Code review	21
3.5.2	Работа с ветками в системе контроля версий	22
3.6	Вынужденный простой сотрудников.....	22

1 Регламент оказания технической поддержки

1.1 Список терминов

«Техническая поддержка» – информационно-консультационная поддержка Заказчика по вопросам обслуживания рабочего продукта, предоставляемая в рабочее время по электронной почте и иным каналам связи на основании регистрируемых обращений.

«Запрос на оказание технической поддержки» или «Обращение» – обращение Заказчика за технической поддержкой к Исполнителю по каналам, указанным в настоящем Регламенте, относящееся к использованию Программного Продукта.

«Релиз» – одно или более изменений в системе, построение, тестирование и развертывание которых выполняется совместно.

«Инцидент» – какое-либо существенное, незапланированное событие, которое происходит во время работы программного обеспечения, и требует последующего исследования и поправки.

«Устранение инцидента» – восстановление работоспособности (выполнение всеми пользователями доступных функций), выдача инструкций по разрешению сходных ситуаций, также, при необходимости описание проблемы с присвоением идентификационного номера и указание релиза, в рамках которого проблема будет устранена.

«Проблема» – причина одного или нескольких Инцидентов.

«Обращение» – письменное обращение по каналу, в случае возникновения инцидента.

«Программный продукт (ПП)» – совокупность программных средств Исполнителя, обеспечивающих реализацию целевых функций («МионитCRM»).

«Рабочее время» – промежуток времени с 8:00 до 17:00 по московскому времени за исключением выходных и праздничных дней, согласно законодательству РФ.

«Рабочий час» – астрономический час в пределах Рабочего дня.

«Время реакции» на обращение означает срок между моментом приема обращения от Заказчика и моментом первого ответа Исполнителя по обращению.

1.2 Общие положения

Настоящий Регламент устанавливает порядок оказания услуг по технической поддержке рабочего продукта Службой технической поддержки Общества с ограниченной ответственностью ООО «Мионит».

Фактический почтовый адрес, по которому осуществляется процесс технической поддержки: Общество с ограниченной ответственностью «Мионит», 414000, Астраханская область, г Астрахань, Казанская ул, д. 44, кв. 4.

Перечень задач, решаемых в рамках технической поддержки:

- оказываются консультации по вопросам имеющегося функционала, предоставления ссылок для скачивания;
- оказываются общие консультации по выбору оборудования для работы с программным продуктом;
- оказываются консультации по установке программного продукта.
- даются рекомендации по типовым проблемам, возникающим при установке;
- разъясняются вопросы настройки программного продукта в случае возникновения затруднений при работе с документацией;
- решаются общие вопросы по установке и настройке рабочего продукта на сервере, принимаются запросы по совершенствованию функционала продукта;
- принимаются сообщения об обнаруженных ошибках в программном продукте;
- осуществляется помощь в поиске и устранении проблем в случае некорректной установки обновления;
- консультационная помощь в поиске и устранении причин, вызвавших сбой в работе, если выявленная причина (или ошибка) находится в коде программного продукта, устраняется в следующих обновлениях;
- если обнаружена критическая ошибка в работе продукта (и подтверждена сотрудниками технической поддержки) – оказывается консультация по восстановлению нормальной работы продукта.

Для обычных и критичных обращений установлено минимальное время реакции специалистов технической поддержки:

Категория обращения	Время реагирования
Обычное	Время реагирования - в течение 8 рабочих часов

Критическое	Время реагирование - в течение 2 рабочих часов
-------------	--

Заказчик обращается в Службу технической поддержки путем направления Обращения с описанием возникшей проблемы.

Исполнитель оказывает помощь в восстановлении работоспособности рабочего продукта при возникновении ошибок в работе, которые подтверждены сотрудниками технической поддержки Исполнителя.

При возникновении ошибок, решение которых не может быть осуществлено в краткие сроки специалистом технической поддержки, данные проблемы решаются Архитектурным комитетом, состоящим из представителей Заказчика и Исполнителя.

1.3 Режим работы технической поддержки

По обращениям техническая поддержка оказывается по рабочим дням с 8 до 17 часов по московскому времени, кроме выходных и праздничных дней (по производственному календарю РФ).

Время решения вопроса может зависеть от сложности решаемой проблемы и необходимости передачи вопроса в отдел разработки.

Время реакции определяется общей загрузкой технической поддержки. Реакция сотрудников службы технической поддержки на поступление дополнительной информации должна быть не дольше максимального времени реакции, указанного в данном Регламенте.

Вопросы, поступающие в техническую поддержку, решаются в порядке очередности поступления обращений.

В процессе сопровождения задействованы следующие специалисты:

- администратор системы;
- специалист технической поддержки;
- программист-разработчик.

1.4 Порядок подачи и обработки обращений

Основанием для выполнения работ является обращение от Заказчика. Обращение может быть создано в форме письма с вопросом на электронную почту support@mionit.org.

В обращении должны быть точно сформулированы вопросы, требующие разъяснения, и описаны проблемы, требующие решения. Для более оперативного решения вопросов обращение должно включать следующую информацию:

- наименование организации;
- описание проблемы и пошаговое описание действий по воспроизведению проблемы (по возможности);
- вопросы желательно задавать, используя терминологию документации по программному продукту;
- номер используемой версии программного продукта.

Службой технической поддержки может быть запрошена дополнительная информация по настройкам ПО, используемым версиям, файлы обмена данными, скриншоты и т.д.

По каждому электронному письму или звонку, принятому службой технической поддержки, высылается на адрес пользователя письмо с подтверждением о принятом обращении.

При отправке обращения в службу технической поддержки Заказчик получает уведомление о его регистрации и уникальный номер созданной заявки не позднее максимально возможного времени реакции. Номер полученной заявки необходимо сохранять в теме письма во время последующей переписки со службой технической поддержки по данному вопросу.

Созданная заявка доступна Заказчику для просмотра в Системе регистрации заявок. Количество пользователей Системы регистрации заявок со стороны Заказчика – 5 (пять).

Информация по заявке в Системе ежедневно актуализируется. По заявке отображаются следующие данные:

- прогнозируемый срок решения задач по заявке;
- описание причины возникновения проблемы;
- классификация проблемы (ошибка в ПО, проблема в инфраструктуре Заказчика, дубликат заявки, заявка на доработку и кастомизацию ПП);
- статус работ по заявке («Ведутся работы», «Ожидается реакция со стороны заказчика», «Ожидается реакция со стороны инфраструктурной службы», «Требуется развертывание», «Проблема решена», «Заявка закрыта»);
- для проблем, классифицированных как «Заявка на доработку и кастомизацию ПП» указывается оценка времени разработки и описание

функционала, которые требуют дополнительного согласования с Заказчиком.

- для проблем, классифицированных как «Дубликат» указывается ссылка на ранее выполненную заявку.

При создании обращения или при отправке обращения по электронной почте можно включать скриншоты и графические пояснения, которые могут помочь в решении проблемы. Скриншоты должны быть подготовлены в форматах: JPG, GIF, PNG.). Все скриншоты или фотографии рекомендуется не включать в текст письма, а прикладывать в виде вложений к письму.

При подаче обращения по e-mail обращение должно содержать корректную информацию о зарегистрированном пользователе продукта: адрес электронной почты, логин в системе и т.п. Указанная информация используется для однозначной идентификации пользователя. Обратите внимание, обращение будет принято только в случае, если письмо отправлено с электронного адреса Заказчика, который зафиксирован в базе данных Исполнителя, в противном случае необходимо обратиться в отдел продаж Исполнителя и сообщить новый адрес электронной почты.

Ответы на стандартные, часто задаваемые вопросы, могут быть даны в виде ссылок на соответствующую страницу документации по программному продукту, на скачивание руководств, сайты разработчиков программного обеспечения.

Решение вопросов по обращениям может быть отложено по следующим основным причинам:

- невозможно повторить описанную проблему на аналогичной конфигурации оборудования;
- пользователь не может предоставить достаточно информации или не может предоставить информацию своевременно для решения проблемы;
- вопрос требует доработки функционала и/или выпуска обновления для программного продукта;
- пользователь выполняет действия в нарушение технических требований по установке и использованию программного продукта, внесены изменения в исходный код программного продукта, программный продукт используется на оборудовании, не рекомендованном специалистами Исполнителя;
- используется нелицензионная копия программного продукта;

- вопрос выходит за рамки технической поддержки (например, вопрос не по программным продуктам Исполнителя);
- вопрос задан некорректно, и решение проблемы затягивается из-за несвоевременного предоставления информации по обращению.

При отправке обращений по электронной почте возможно возникновение проблемных ситуаций с работой сторонних почтовых сервисов или спам-фильтров. Обращение считается принятым, только если вы получили письмо о регистрации вашего обращения с уникальным номером заявки. Это означает, что письмо прошло проверку на антиспам и было зарегистрировано в системе технической поддержки. В случае возникновения проблем с доставкой почтовых сообщений рекомендуется сообщить об этом сотрудникам Исполнителя с указанием электронного адреса, с которого высылалось письмо.

Заявка получает статус «Заявка закрыта» после получения подтверждения от Заказчика об отсутствии вопросов по этой заявке. В случае отсутствия ответа Заказчика о закрытии заявки, в течение 14 (четырнадцати) календарных дней заявка также считается завершенной.

2 Техническая поддержка в системе Intraservice

Все взаимодействие технической поддержки документируется в системе IntraService

В рамках Intraservice работают две роли пользователей:

- сотрудники клиентов;
- сотрудники технической поддержки ООО «Мионит».

2.1 Регистрация участников процесса

Первичная регистрация сотрудников клиента производится после отправки сообщения через почту на support@mionit.org. На сайт приходит заявка со стандартными настройками, а пользователю на почту приходит логин и пароль для входа на сайт.

Так же регистрация пользователей может производиться администратором технической поддержки ООО «Мионит» по изначальной договоренности с клиентом.

2.2 Выбор сервиса и оформление заявок

После авторизации пользователя открывается стартовая страница системы – раздел Главная, на которой отображается список сервисов.

Сервисы служат для логического объединения заявок в системе. Сервисы могут соответствовать:

- клиенту (например, «Поддержка филиала в Новосибирске»);
- контракту (например, «Поддержка по договору 1С-4503/11»);
- оказываемой услуге (например, «Ремонт оргтехники»).

Сотрудники клиентов по умолчанию видят только свой сервис и общую очередь.

Сервис общей очереди служит для создания заявок, приходящих по e-mail.

Администратор видит все сервисы, может их создать / удалить и т.д.

При переходе к сервису имеется возможность создать новую заявку или удалить старую.

Для оформления заявки необходимо:

- перейдите в раздел «Заявки»;
- нажмите на кнопку «Создать заявку» (если этой кнопки нет, значит для вашей роли на этом сервисе недостаточно полномочий);
- выбрать сервис, в котором будет создана заявка;

- определить тип заявки:
 - стандартный: создание заявки без определенного типа;
 - инцидент: создание заявки, при незапланированном прерывании или снижении качества ИТ-услуги. Сбой конфигурационной единицы, который еще не повлиял на услугу, также является инцидентом;
 - проблема: причина одного или нескольких инцидентов. Обычно при создании записи о проблеме причина неизвестна, и за дальнейшее её расследование отвечает процесс управления проблемами;
 - с созданием пользователя: при необходимости создания заявки с последующим созданием нового пользователя;
 - запрос на изменение: формальное предложение на выполнение изменения. Запрос на изменение включает в себя детали предложенного изменения и может быть записан в бумажном или электронном виде;
 - запрос на обслуживание: запрос пользователя на информацию, или консультацию, или на стандартное изменение, доступ к ИТ-услуге.
- заполните обязательные поля (название, статус, приоритет- обязательные поля для (отмечены звездочкой));
- при необходимости вставьте подтверждающие документы (скриншоты, логи и т.д.). Если этой возможности нет, значит для вашей роли на этом сервисе недостаточно полномочий;
- нажмите «Сохранить» или «Сохранить и вернуться».

Доступность к просмотру/изменению полей определяется ролью пользователя на сервисе и признаком «Участник заявки». Пользователь становится участником заявки, если он:

- является заявителем (если он сам создал заявку или диспетчер создал заявку от его имени);
- назначен наблюдателем (пользователь имеет возможность получения уведомлений по заявкам. При наличии прав доступа наблюдатель может просматривать и изменять соответствующие поля заявки) назначается заявителем или администратором;
- назначен исполнителем (сотрудник, назначенный администратором для решения проблемы).

Задание любого участника заявки (заявителя, исполнителя и наблюдателя) всегда ограничивается общим списком пользователей, назначенных на данный сервис.

2.3 Определение статуса заявки

Статусы заявки делятся на:

- «Открыта»;
- «Требуется уточнения»;
- «В работе»;
- «Выполнена»;
- «Закрыта»;
- «Отменена»;
- «Отложена».

По умолчанию ставится статус «Открыта». Пользователь имеет возможность выбирать такие статусы, как: «Открыта», «Отменена», «Закрыта».

Статус «Открыта» ставится при создании заявки клиентом.

Статус «Требуется уточнения» ставится исполнителем или администратором в случае необходимости дополнительной информации от заявителя с обязательным указанием вопроса (объекта уточнения).

Статус «В работе» ставится администратором, когда назначен сотрудник, ФИО которого указано в колонке «Исполнитель», и исполнитель в данный момент обрабатывает заявку.

Изменение, предназначенное для решения проблемы, должно быть рассмотрено при оценке результатов внедрения до закрытия проблемы. Если изменение дало ожидаемый результат, проблема может быть закрыта, и в базе данных о проблемах ее статус будет администратором изменен на статус «Выполнена»

Статус «Отложена» ставится администратором или исполнителем, если заявка отложена на определенный срок (причиной может быть отсутствие реакции клиента на статус «Требуется уточнения» на протяжении длительного периода или недоступность клиента).

Статус «Отменена» устанавливается пользователем или администратором, если проблема была разрешена пользователем самостоятельно или если был найден ответ на поставленный вопрос.

Статус «Закрыта» устанавливается только пользователем (в случае предоставленного решения проблемы или ответа на поставленный вопрос). Так же с разрешения клиента, администратор может установить данный статус заявки, если было найдено решение и обговорено с заявителем.

2.4 Определение приоритета

Приоритет является одним из обязательных полей для заполнения, так как от него напрямую зависит очередность в списке заявок.

Приоритет делится на:

- низкий – ошибка должна быть исправлена, ее наличие не является критичной, и не требует срочного решения;
- средний – ошибка должна быть исправлена, ее наличие не является критичной, но требует обязательного решения.
- высокий – ошибка должна быть исправлена, ее наличие может привести к критической ошибке для проекта, требует обязательного решения.
- критический – ошибка должна быть исправлена как можно быстрее, т.к. ее наличие является критической для проекта.

2.5 Работа с инцидентами

Инцидент – незапланированное прерывание или снижение качества ИТ-услуги. Сбой конфигурационной единицы, который еще не повлиял на услугу, также является инцидентом.

При создании заявки, по изначальному согласованию с клиентом, тип заявки указывается «Инцидент».

После создания новой заявки администратор обязан ознакомиться с ней. При неполном описании проблемы или необходимости уточнить ее детали, администратором изменяется статус с «Открыта» на «Требуется уточнения» и обязательно заполняет поле «Комментарий», указывая, какая именно информация необходима. После чего, пользователю на почту автоматически отправляется письмо с новыми изменениями. Получая ответ от пользователя, статус вновь меняется на «Открыта».

Если инцидент типовой и администратор способен справиться с ним самостоятельно, без необходимости обращаться за помощью к специалистам отдела разработок, применяются соответствующие шаги по решению проблемы, после чего статус переводится в «Выполнена». На почту пользователю приходит отбивка о выполнении заявки, и, в случае проверки пользователем инцидент не

обнаруживается, пользователь должен перевести статус заявки на «Закрито». В противном случае пользователь меняет статус на «Открыто»

Если администратор не может сам решить инцидент, он выносится на ежедневном собрании с соответствующей командой, занимающейся разработкой данного ПО. После чего выносится информация по номеру заявки и необходимым действиям.

При принятии инцидента в работу, разработчиками создается запрос на сайте mionit.visualstudio.com и передается ссылка запроса администратору технической поддержки. Всю полученную информацию администратор указывает в заявке (уточняет вопросы, заданные разработчиком, меняет статус заявки, вставляет ссылку на заявку и т.д.)

При получении ссылки от разработчиков, администратор вставляет её в поле «Ссылка в TFS» соответствующей заявки и меняет статус заявки на «Выполнена».

При повторных проявлениях инцидентов, замеченных и на других устройствах/у других пользователей инцидент переходит в тип - Проблема.

2.6 Работа с проблемами

Проблема - причина одного или нескольких инцидентов. Обычно при создании записи о проблеме причина неизвестна, и за дальнейшее её расследование отвечает процесс управления проблемами.

При обнаружении однотипных или близких по типу инцидентов, администратор должен создать заявку с типом «Проблема». После чего необходимо зайти в заявки со всеми подобными инцидентами и сделать их дочерними к созданной «заявке-проблеме» (+ Действия -> Родитель - дочерняя -> Сделать дочерней к другой заявке -> Выбрать родительскую заявку). Все дочерние заявки переводятся администратором в статус «Закрита».

Также можно из созданной «заявки-проблемы» прикрепить «заявки-инциденты» (+Действия -> Родитель - дочерняя -> Сделать родительской к другой заявке -> Выбрать дочернюю заявку). В заявках со статусом «Проблема», как и в «Инцидент», используется один механизм работы:

- проблема выносится на ежедневном собрании с соответствующей командой, занимающейся разработкой данного ПО;
- в соответствующей группе Техническая поддержка выносится информация по номеру заявки и необходимым действиям;

- при принятии проблемы в работу, разработчиками создается запрос на сайте mionit.visualstudio.com и передается ссылка запроса администратору технической поддержки;
- всю полученную информацию администратор указывает в заявке (уточняет вопросы, заданные разработчиком, меняет статус заявки, вставляет ссылку на заявку и т.д.);
- при получении ссылки от разработчиков, администратор вставляет её в поле «Ссылка в TFS» соответствующей заявки и меняет статус заявки на «Выполнена».

В случае, когда в проблеме имеются несколько дочерних заявок с собственными ссылками TFS, ссылки переносятся в родительскую заявку, а дочерние заявки переходят в статус «Выполнена». Если все дочерние заявки закрыты, то в родительской меняется статус на «Выполнена». Заявки без ссылок TFS не закрываются.

2.7 Работа с запросами на изменения

Запрос на изменение поступает при желании внести более удобные механизмы выполнения задач, изменение внешнего вида и т. д.

Предложения, поступившие от клиентов/пользователей, выносятся на обсуждение с разработчиками. При целесообразности предлагаемых изменений/дополнений, сотрудники отдела разработок вносят предложения в план работы, а заявка переходит в соответствующий статус «Выполнена».

2.8 Недокументированное поведение системы и ошибки

При обработке заявки необходимо понять, к какой категории относится проблема (является ли это недокументированным поведением системы или ошибкой).

Ошибка (баг) – ошибка в программе или системе, которая выдает неожиданный или неправильный результат.

При обнаружении ошибки специалист техподдержки обращается в отдел разработок для создания нового бага. После создания ссылки на ошибку, администратор добавляет ссылку в поле «Ссылка TFS» и закрывает заявку с соответствующим статусом и комментарием.

Недокументированное поведение системы (фича) – особенность, характерная черта или какая-нибудь недокументированная дополнительная возможность программы.

При обнаружении такого недокументированного поведения системы, специалист техподдержки обращается в отдел разработок для создания РВІ. После создания ссылки на РВІ, администратор добавляет ссылку в поле «Ссылка TFS» и закрывает заявку с соответствующим статусом и комментарием.

Для понимания этапа решения проблемы администратор отслеживает предоставленные ссылки на ошибку или недокументированное поведение системы.

Статусы:

- approved (одобренный);
- committed (порученный);
- done (готовый);
- new (новый);
- removed (удалённый).

3 Регламент процесса разработки

Фактический почтовый адрес, по которому осуществляется процесс разработки: Общество с ограниченной ответственностью «Мионит», 414000, Астраханская область, г Астрахань, Казанская ул, д. 44, кв. 4.

3.1 Термины

- Backlog – упорядоченный по приоритету журнал детальных требований к продукту и перечень рабочих задач;
- Product backlog item (PBI) – элемент, описывающий выполнимое детальное функциональное или архитектурное требование;
- Спринт – фиксированный по времени этап проекта;
- Ветка – набор версий исходных файлов в системе контроля версий с единой историей;
- Коммит – операция по закреплению новой версии исходных файлов в системе контроля версий;
- Pull request – запрос на перенос изменений исходных файлов из одной ветки в другую;
- Ревью (Code review) – процедура обеспечения внутреннего качества продукта путем чтения и критики исходного текста программы;
- Тестовые сценарий (test case) – документ, описывающий действия тестировщика и ожидаемые ответы разрабатываемой системы;
- Сборка (build) – автоматическая процедура создания дистрибутива продукта из исходных файлов.

3.2 Участники

В процессе разработки задействованы следующие специалисты:

1. Руководитель проекта (РП) – единственное в проекте лицо, несущее ответственность за формирование и контроль рамок проекта и коммуникацию с заказчиком. Основной представитель заказчика для команды, основной представитель компании для заказчика.
2. Аналитик (дополнительная для РП роль) – лицо, осуществляющее бизнес и системный анализ, детализирующее, фиксирующее и контролирующее исполнение требований командой разработки.

3. Команда проекта (разработчики, тестировщики) – осуществляют работы по реализации программного обеспечения и контролю внутреннего и внешнего качества.
4. Scrum-master – член команды проекта, ответственный за соблюдение процесса разработки в команде, модератор общения участников команды в рамках процедур процесса.
5. СТО (технический директор) – распределяет ресурсы между проектами, принимает решения о технологическом стеке.

3.3 Инструменты

1. Microsoft Team Foundation Server (TFS)
2. Microsoft Teams (MS Teams)
3. Time-tracker

3.4 Процедуры

3.4.1 Планирование

Планирование – совещание команды, аналитиков и РП, проводимое в начале спринта. До планирования:

РП должен согласовать состав и доступность членов команды на спринте с СТО и сообщить об этом scrum-master и другим членам команды.

Аналитики и/или РП предоставляют приоритизированный backlog. Из PBI вынесенных вверх backlog (предлагаемых на спринт) должно четко и ясно следовать зачем данная функциональность заказчику, роли пользователей(системы), участвующие в PBI, критерии приемки, основной и альтернативные сценарии.

Scrum-master выставляет фокус-фактор команде.

В случае нарушения условий аналитиком и/или РП (входные данные не предоставлены в полном виде или возникли вопросы по ним, которые не могут быть разрешены сразу) команда вправе перенести планирование итерации на следующий день. Возможен простой команды в этом случае.

Фаза 1: аналитик и/или РП предлагают и объясняют PBI команде, отвечают на вопросы по ним в порядке приоритета в backlog. Если PBI не вызывает вопросов у команды, команда его принимает и переводит в статус Approved. Длительность фазы ~ 1 час.

Фаза 2: Команда разбивает PBI на задачи, оценивает их, и формирует Sprint Backlog.

Длительность фазы ~ 2 часа.

Правила:

- разработчики и тестировщики оценивают задачу в идеальных часах, эта оценка не должна учитывать время, затрачиваемое на исправление багов и замечаний, code review;
- желательная длительность задач не превышает 6 часов;
- тестировщик включает в оценку написание тестовых сценариев и их прогонку;
- разработчики ставят и оценивают задачи на code review (но не на исправлении задач code review);
- участники команды оценивают необходимые работы по развертыванию, если эти работы не автоматизированы и требуют существенных усилий;
- в процессе оценки команда может запросить информацию у РП и/или аналитика, в конце участники договариваются по PBI, которые будут взяты в спринт.

Итог: В backlog спринта перенесены PBI с оцененными командой задачами. Трудоемкость в спринте не должна нарушать ограничений загруженности команды, т. е. должна быть исполнима в теории.

PBI перемещаются в колонку Develop доски PBI.

Некоторые оцененные PBI (из Approved) могут не попасть в спринт, но в процессе по договоренности между РП и членами команды их можно взять.

Если запланированные работы по PBI не выполнены в спринт, невыполненные задачи переносятся в новый спринт.

3.4.2 Ежедневная работа участников команды

Основная работа участников команды начинается после планирования и продолжается до проведения демонстрации продукта.

Типовой процесс исполнения PBI:

- Разработчик берет задачу (задачи) и переносит ее(их) в InProgress.
- Разработчик создает отдельную ветку от ветки development.
- Разработчик выполняет работы по техническому проектированию и кодированию в рамках созданной ветки.

Следует соблюдать следующие рекомендации:

- делать коммиты достаточно часто;
 - не коммитить некомпилируемый код;
 - перед каждым коммитом запускать модульные тесты;
 - при появлении опережающих коммитов в ветке development сливать их в свою ветку из development;
 - как правило, перевод разработчиком задачи в Done должен сопровождаться коммитом в созданную ранее ветку;
- Разработчик показывает выполненную работу аналитику и/или РП. При демонстрации багов быть не должно. Если в PBI есть явно указанные основной и альтернативные сценарии, то они должны быть продемонстрированы.
 - Аналитик и/или РП подтверждает выполнение требований или дает замечания разработчику. Если замечаний нет, PBI перемещается в колонку Review. Аналитики/РП сообщает об этом команде в MS Teams. Если замечания есть разработчик исправляет их и снова показывает работу Аналитику/РП.
 - Разработчик создает pull request на человека, за которым закреплена задача code review.
 - Разработчик, начавший выполнение code review, обязан перевести задачу на code review в In Progress и на ежедневной обновлять оценку в поле Remaining Work (как и для обычной задачи) до исправления замечаний и принятия изменений в ветку development.
 - При выявлении замечаний осуществлявший code review разработчик возвращает PBI в колонку Develop и сообщает о результате ревью команде в MS Teams.
 - При получении замечаний, разработчик исправляет их за счет фокус своего фокус- фактора и отправляет на ревью повторно.
 - После приемки ревью и слияния новой ветки с development осуществлявший ревью разработчик переносит PBI в колонку Test и сообщает об этом команде в чате MS Teams.
 - Тестировщик меняет состояние своих задач и оценку remaining work ежедневно по тем же правилам, что и разработчик.
 - Тестировщик не зависимо от программиста готовит тестовые сценарии по требованиям в PBI.

- При появлении готового к проверке PBI тестировщик может начать задачу по его тестированию переводя ее в состояние In Progress. Задача по тестированию будет оставаться в состоянии In Progress до успешного завершения тестирования.
- Тестировщик запускает сборку ветки development.
- Тестировщик проверяет выполненный PBI по тестовым сценариям. Если найдены ошибки, то тестировщик сообщает об ошибке команде в MS Teams и возвращает PBI в колонку Develop в TFS.
- Если ошибок нет, тестировщик завершает свои задачи по тестированию и переносит PBI в колонку Ready сообщает об этом команде в MS Teams.
- Протестированные PBI переводятся в Done РП при отсутствии обоснованных замечаний с его стороны к качеству исполнения и несоответствию формализованным требованиям до конца спринта.

3.4.3 Ежедневный контроль

Члены команды в конце рабочего дня обязаны изменить значение поля Remaining work у своих задач в состоянии In Progress в соответствии с новой оценкой объема оставшейся работы или перевести задачу в состояние Done.

Также в конце рабочего дня разработчики обязаны отчитаться по трудозатратам на проект в тайм-трекере.

Ежедневно в заранее установленное время для контроля работы участников команды проводится daily meeting. На daily meeting присутствуют члены команды, аналитики и РП. Модерирует daily meeting scrum-master. Daily meeting должен составлять 15-20 минут. Каждый участник говорит, что сейчас делает, есть ли проблемы, когда планирует завершить начатые работы.

Дополнительно на daily meeting демонстрируется диаграмма сгорания и доска с задачами спринта. Если эти инструменты индицируют проблемы, им должны быть найдены причины и намечены возможные способы выравнивания ситуации.

3.4.4 Демонстрация

Демонстрация разработанного проводится в конце итерации командой. На демонстрации в обязательном порядке присутствуют участники команды, аналитики и РП. На демонстрацию могут быть приглашены представители

заказчика, представители других подразделений компании. Состав приглашенных лиц определяет РП.

Демонстрацию ведет один заранее (на планировании) выбранный от команды человек. Еще один выбранный от команды человек (или аналитик) фиксирует обратную связь по продемонстрированному функционалу. Итоговый протокол демонстрации публикуется в MS Teams.

Состав демонстрируемых фич определяется командой по согласованию с РП накануне демонстрации.

3.4.5 Ретроспектива

Ретроспектива проводится в конце спринта, после проведения демонстрации. Обязательно участие всех членов команды, аналитиков, РП в течение всего времени. Модерирует и обеспечивает проведение ретроспективы scrum-master. Среди участников ретроспективы выбирается человек, ведущий протокол ретроспективы. Итоговый протокол ретроспективы публикуется в MS Teams. Желательная длительность ретроспективы 1-2 часа.

Каждый из участников должен подготовиться к ретроспективе, рассмотрев протокол предыдущей ретроспективы, проанализировав недостатки/достоинства своей работы и работы своих коллег, и рассказать следующее:

- что получилось, что не получилось (почему) у него и у команды на спринте;
- если не выполнены все задачи, объяснить почему и сказать, когда будут выполнены, предложить способы недопущения подобных отклонений в дальнейшем;
- описать проблемы, предложения, замечания, если они есть в процессе, проекте.

В конце ретроспективы scrum master подводит итоги, оглашая предложения по изменению параметров процесса в команде, перераспределению ролей, ресурсному обеспечению команды (материально-техническому и/или кадрово-компетентностному).

3.5 Организация работы с кодом

3.5.1 Code review

Необходимыми условиями для выполнения code review являются:

- наличие всех необходимых комментариев в коде;
- успешная сборка самого решения;

- выполнение всех модульных тестов.

В случае невыполнения всех необходимых условий code review отклоняется ревьювером сразу без разбора кода.

Если разработчик внес изменения файл, то проверяющий может смотреть весь файл, который затронули изменения. Проверяющий вправе заставить исправлять код разработчика, даже если не он автор этих изменений.

3.5.2 Работа с ветками в системе контроля версий

В проекте имеются минимум две ветки:

- master;
- development.

Ветка Master содержит в себе исходный код, build которого развернут у заказчика.

Ветка Development предназначена для разработки новых фич, ведения нового проекта. Для ветки master создается политика, которая ограничивает возможность изменения данной ветки, теперь master можно изменить только через pull request с обязательным выполнением build (в build запускаются также модульные тесты). Коммиты в ветку master должны помечаться тегами версии.

Если реализуется новый функционал, то код правится в ветке development. Для разработки новых фич создаются новые ветки от development и далее через pull request переливаются в ветку development, после чего удаляются. Перед публикацией Заказчику в эксплуатируемую среду ветка development должна быть перелита в ветку master.

Если у клиента нашли критический баг должна быть создана ветка от ветки master для устранения ошибки. Далее ветка переливается в master и оттуда в development.

3.6 Вынужденный простой сотрудников

Простой, вызванный нехваткой или ожиданием задач, сотрудник должен отмечать в тайм-трекере как «Простой», даже если занимался потенциально полезной для проекта деятельностью.

Во время вынужденного сотрудника сотрудник не должен:

- отвлекать других сотрудников;
- активно демонстрировать свое бездействие другим сотрудникам;

- вносить изменения в проект по своей воле, не согласуя свои действия и не ставя в известность РП, аналитиков и других членов команды.

Во время вынужденного простоя сотрудник может:

- предложить свою помощь сотруднику, у которого есть задачи;
- предложить свою помощь в проведении code review или дополнительном тестировании (в т. ч. автоматизированном);
- предложить свою помощь в создании технической документации;
- предложить варианты ликвидации технического долга;
- предложить варианты улучшения инфраструктуры/инструментов разработки;
- заняться самообразованием по рабочей специализации.