
ООО «Рога, копыта и программы»

УТВЕРЖДАЮ
Директор

_____"_____"_____" 2012г.

**ОРГАНИЗАЦИЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ
РАЗРАБОТКИ, ВНЕДРЕНИЯ И СОПРОВОЖДЕНИЯ ПРОГРАММНЫХ
ПРОДУКТОВ**

**TFS. РАБОЧИЕ ЭЛЕМЕНТЫ.
СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ**

Сведения о документе

Версия:	01.01
Редакция:	12.11.2015 21:55
Дата выпуска:	???
Основание:	Распоряжение № ...
Дата утверждения:	???
Взамен:	-

АННОТАЦИЯ

Настоящий документ является частью организационного обеспечения выполнения проектов разработки программного обеспечения. В документе представлены сценарии использования (применения) рабочих элементов MS TFS. За основу взят шаблон MSF for CMMI.

СОДЕРЖАНИЕ

АННОТАЦИЯ	2
СОДЕРЖАНИЕ	3
СПИСОК ИЗМЕНЕНИЙ.....	5
1. ВВЕДЕНИЕ.....	5
1.1. Цель документа	5
1.2. Сфера применения	5
1.3. Нормативная основа.....	5
1.4. Ссылки	5
1.5. Определения, акронимы и сокращения.....	5
2. ОБЩАЯ СХЕМА ПРИМЕНЕНИЯ РАБОЧИХ ЭЛЕМЕНТОВ	6
2.1. Роли, области, рабочие элементы.....	6
2.2. Ключевые моменты WorkFlow	7
3. УПРАВЛЕНИЕ ТРЕБОВАНИЯМИ.....	8
3.1. Создание требования, состояние: Поступившее	8
3.2. Переход из Поступившее в В работе	8
3.3. Состояние: В работе	9
3.4. Переход из В работе в Закрыто.....	9
3.5. Переход из Закрыто в Закрыто и проверено	9
4. УПРАВЛЕНИЕ ЗАДАЧАМИ.....	9
4.1. Создание Задачи, состояние: Поступившее.....	10
4.2. Состояние: Поступившее.....	10
4.3. Переход из Поступившее в В работе	10
4.4. Состояние: В работе.....	10
4.5. Переход из В работе в Закрыто.....	10
4.6. Состояние: Закрыто	10
4.7. Состояние: Закрыто	11
4.8. Переход из Закрыто в Закрыто и проверено	11
4.9. Переход из Закрыто в Поступившие.....	11
4.10. Переход из Закрыто в Поступившие	11
5. УПРАВЛЕНИЕ ТЕСТИРОВАНИЕМ.....	12
5.1. Создание Ошибки, состояние: Поступившее	12
5.2. Состояние: Поступившее.....	12
5.3. Переход из Поступившее в В работе	13
5.4. Состояние: В работе.....	13
5.5. Переход из В работе в Закрыто.....	13

5.6.	Состояние: Закрыто	13
5.7.	Состояние: Закрыто	13
5.8.	Переход из Закрыто в Закрыто и проверено	13
5.9.	Переход из Закрыто в Поступившие.....	13
6.	УПРАВЛЕНИЕ ИЗМЕНЕНИЯМИ.....	14
6.1.	Создание Запроса на изменение	14
6.2.	Состояние: Поступившее.....	14
6.3.	Переход из Поступившее в В работе	15
6.4.	Состояние: В работе.....	15
6.5.	Переход из В работе в Закрыто	15

СПИСОК ИЗМЕНЕНИЙ

№ п/п	Дата	Автор	Изменение

1. ВВЕДЕНИЕ

1.1. Цель документа

Документ содержитTODO

1.2. Сфера применения

Документ предназначен для использования в области разработки прикладного программного обеспечения и документации.

Документ является частью документационного обеспечения организационной системы производства программного обеспечения.

1.3. Нормативная основа

В качестве нормативной основы при разработке данного Документа использованы стандарты:

1. ГОСТ 34.003-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Термины и определения.
2. ГОСТ Р ИСО/МЭК 12207-99 Информационная технология. Процессы жизненного цикла программных средств.
3. Guide to Software Engineering Base of Knowledge (SWEBOK). IEEE Computer Society, 2004¹

1.4. Ссылки

4. Visual Studio Application Lifecycle Management. Планирование и отслеживание проектов. MSF для улучшения процесса CMMI. <http://msdn.microsoft.com/ru-ru/library/dd997574%28v=vs.100%29>
5. Организация производственных процессов разработки, внедрения и сопровождения программных продуктов. Глоссарий. ООО «Рога, копыта и программы». Таганрог 2012

1.5. Определения, акронимы и сокращения

Основные определения приведены в документе [5].

¹ Русский перевод: Сергей Орлик. «Основы Программной Инженерии (по SWEBOK)» разработаны на базе IEEE Guide to SWEBOK® 2004, включает перевод <http://SWEBOK.SOrlik.ru>

2. ОБЩАЯ СХЕМА ПРИМЕНЕНИЯ РАБОЧИХ ЭЛЕМЕНТОВ

2.1. Роли, области, рабочие элементы

Перечень ролей определен в документе [5].

В рамках процесса разработки планируется использовать следующие рабочие элементы: Требование (**Requirement**), Задача (**Task**), Ошибка (**Error**), Запрос на Изменение (**Change Request**), Тестовый случай (**Test Case**) [4].

Объединение рабочих элементов в области представлено на Рис. 1, взаимоотношение ролей, рабочих элементов и областей в Табл. 1.

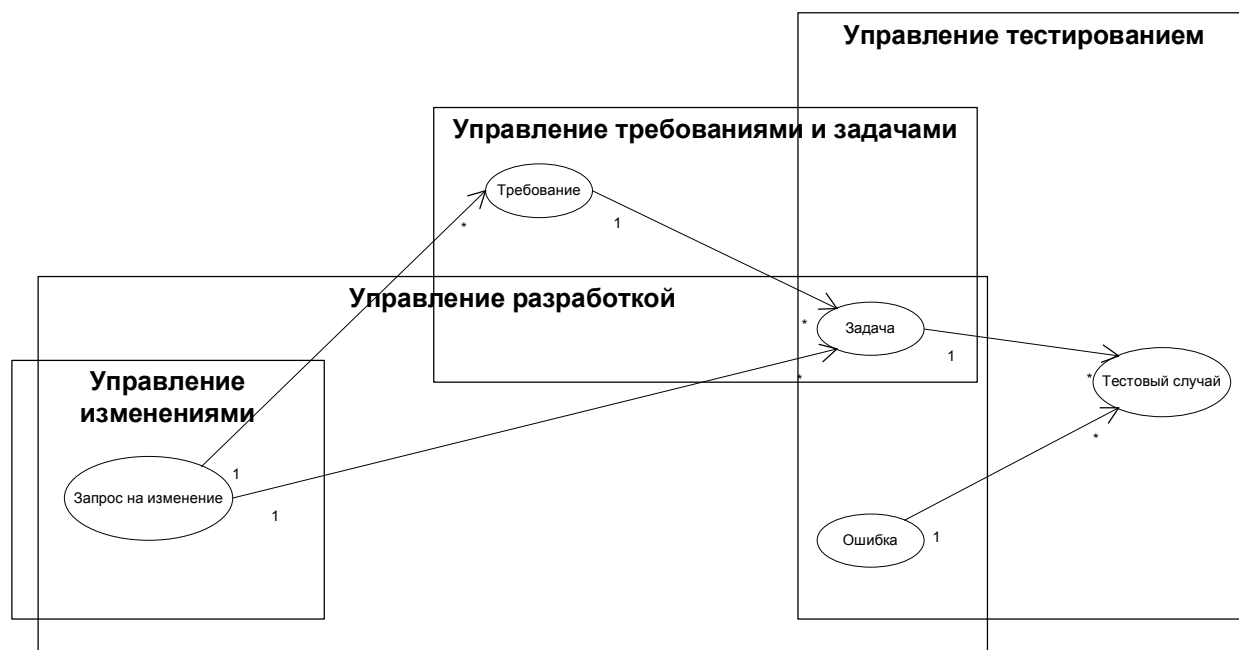


Рис. 1 Роли, области, рабочие элементы

Табл. 1 Роли, области, рабочие элементы

Область -----	Управление требованиями и задачами	Управление тестированием	Управление разработкой	Управление изменениями
Рабочий элемент ----- Роль	Требование (R) Задача (T)	Тест (ТС) Ошибка (E)	Задача (T) Ошибка (E)	Запрос на изменение (CR)
Менеджер	R: назначение итерации			CR: определение типа назначение исполнителя назначение итерации
Аналитик	R: создание T: создание			CR: закрытие E: создание
Проектировщик	T: создание T: назначение исполнителя			
Разработчик	T: назначение исполнителя		T: закрытие E: закрытие	
Тестировщик		ТС: создание E: создание T: подтверждение E: подтверждение R: подтверждение		
Сопровожденец	CR: создание			
Заказчик	CR: создание			

2.2. Ключевые моменты WorkFlow

На вход аналитику поступают рабочие элементы:

- Запрос на изменение, из него всегда создаются Требования, Ошибки или Запрос отвергается. Может быть возвращен за доп информацией
- Требование

На вход аналитика поступают рабочие элементы в состоянии «Поступившие», они их сами перевод в «В работу».

На вход проектировщикам поступают рабочие элементы:

- Требование, из него всегда создаются Задачи
- Задача, может быть возвращено за доп информацией

На вход проектировщика поступают рабочие элементы в состоянии «Поступившие», они их сами перевод в «В работу».

На вход разработчикам поступают рабочие элементы:

- Задача, из Требований всегда создаются Задачи,
- Ошибка, (???есть вопрос, что Задачи, чьи Ошибки пришли тоже возвращаются)

На вход разработчикам поступают рабочие элементы в состоянии «Поступившие», они их сами перевод в «В работу».

На вход тестировщикам поступают рабочие элементы:

- Задача,
- Ошибка, после исправления разработчикам или за доп информацией
- Запрос на изменение, из него всегда создаются Ошибки или Запрос отвергается ???вопрос

На вход тестировщикам поступают рабочие элементы в состоянии «Поступившие», они их сами перевод в «В работу».

3. УПРАВЛЕНИЕ ТРЕБОВАНИЯМИ

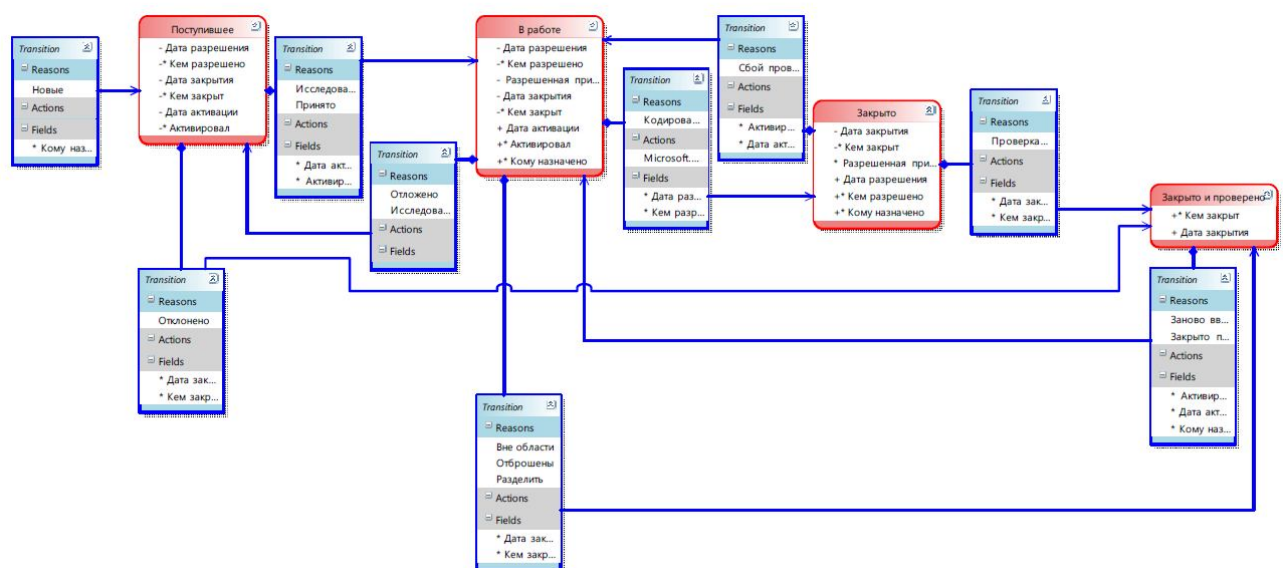


Рис. 2 Жизненный цикл рабочего элемента «Требование»

3.1. Создание требования, состояние: Поступившее

Кто: Аналитик. Как часть его работы по постановке задачи или как результат реакции на одобренный Запрос на изменение (см. «Управление изменениями»).

Причина: **Новое**

3.2. Переход из Поступившее в В работе

Кто: Менеджер. После обсуждения и включения требования в очередную итерацию по результатам совместного решения Группы разработки (Менеджер, Аналитик, Проектировщик, Разработчик, м.б. Сопроводитель).

Выбрать он может легко – запрос «дай требования находящиеся в состоянии Проступившие, и без итерации».

Причина: Новое

Исполнитель: как правило² Проектировщик.

Атрибуты: итерация

Тестировщик по мере появления новых требований и задач, *назначенных на очередную итерацию*, создает Тестовые случаи и *связывает их с задачами или требованиями*.

3.3. Состояние: В работе

Кто: Проектировщик. Создает Задачи и связывает их с этим Требованием
Создание задачи (задач) для Требования *обязательно*.

3.4. Переход из В работе в Замкнуто

Кто: Проектировщик

Причина: все задачи созданы

Исполнитель: Тестирующий

3.5. Переход из Зажато в Зажато и проверено

Кто: Тестировщик. Обычно перед передачей версии ПП заказчику, после того как все задачи, созданные в реализацию данного требования, закрыты и проверены.

Выбрать он может так – запрос «дай требования находящиеся в состоянии Закрыто, назначенные на определенную итерацию». А потом (надо еще подумать как ???TODO) по каждому требованию из списка запрос: «все-ли дочерние задачи требования закрыты и проверены?».

Причина: все задачи, созданные в реализацию данного требования, закрыты и проверены.

4. УПРАВЛЕНИЕ ЗАДАЧАМИ

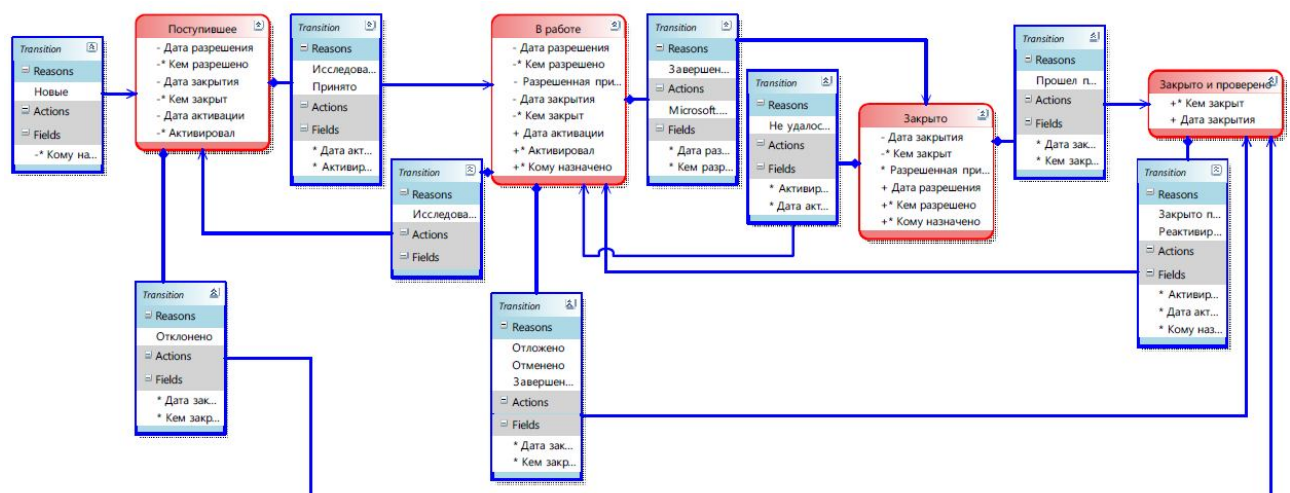


Рис. 3 Жизненный цикл рабочего элемента «Задача»

² Допустима ситуация, когда Задачи на основе Требования создает сам Аналитик. В этом случае Требование назначается на него и он осуществляет создание Задач.

4.1. Создание Задачи, состояние: Поступившее

Причина: **Новое**

Кто: Проектировщик. В процессе анализа требований. Создаваемая задача *обязательно* связана с Требованием.

Исполнитель: Разработчик ???

Тестировщик по мере появления новых требований и задач, *назначенных на очередную итерацию и Разработчику*, создает Тестовые случаи и *связывает их с задачами или требованиями*.

4.2. Состояние: Поступившее

Кто: Разработчик. Созданные задачи назначаются конкретным исполнителям (разработчикам, тестировщикам, ...), это, как правило³, делает⁴ Разработчик. Возможно указывается оценка трудозатрат.

Выбрать – запрос «дай задачи назначенные на Разработчика или не назначенные никому и находящиеся в состоянии Проступившие, можно с учетом итерации».

Исполнитель: конкретный разработчик

Атрибуты: оценка трудозатрат

4.3. Переход из Поступившее в В работе

Причина: ...

Кто: конкретный разработчик. Конкретный разработчик начинает работать над задачей.

Выбрать – запрос «дай задачи назначенные на меня и находящиеся в состоянии Проступившие, можно с учетом итерации».

Исполнитель: остается конкретный разработчик

Атрибуты: оценка трудозатрат

4.4. Состояние: В работе

Кто: конкретный разработчик.

Выбрать – запрос «дай задачи назначенные на меня и находящиеся в состоянии В работе, можно с учетом итерации».

Кодирует, м.б. если задача большая пусть указывает процент завершения.

4.5. Переход из В работе в Закрото

Причина: задача реализована

Кто: конкретный разработчик

Исполнитель: Тестировщик

4.6. Состояние: Закрото

Кто: Тестировщик.

Выбор – запрос «дай все закрытые задачи назначенные Тестировщику или не назначенные никому, можно с учетом сборки».

³ По крайней мере в отношении задач программирования.

⁴ Хорошо, если при этом он укажет и трудоёмкость задачи.

Исполнитель: конкретный тестировщик

4.7. **Состояние:** Закрыто

Кто: конкретный тестировщик.

Выбор – запрос «дай все закрытые задачи назначенные мне, можно с учетом сборки».

После передачи сборки на тестирование осуществляется ее тестирование путем исполнения тестовых случаев.

4.8. **Переход из** Закрыто **в** Закрыто и проверено

Кто: конкретный тестировщик.

Причина: **Проверка пройдена**

4.9. **Переход из** Закрыто **в** Поступившие

Вариант №1 (упрощённый)

Кто: конкретный тестировщик. Причина: **Сбой проверочного теста**

Исполнитель: Разработчик или конкретный исполнитель этой Задачи

В атрибуте Задачи ??? детально описывает дефект. Ошибку *не создает*.

Вариант №2 (развёрнутый)

Кто: конкретный тестировщик.

Причина: **Сбой проверочного теста**

Исполнитель: Разработчик или конкретный исполнитель этой Задачи

Создает рабочий элемент Ошибка, в котором детально описывает дефект. Связывает Ошибку (как дочерний элемент) с Задачей. Подробнее см. в «Управление тестированием».

??? Разработчик или конкретный исполнитель этой Задачи получает «взад» Задачу и еще Ошибку по этой задаче. В процессе обсуждения были мысли: оставлять Задачу у Тестировщика в состоянии **Закрыто**; оставлять Задачу у Тестировщика в состоянии **В работе**.

5. УПРАВЛЕНИЕ ТЕСТИРОВАНИЕМ

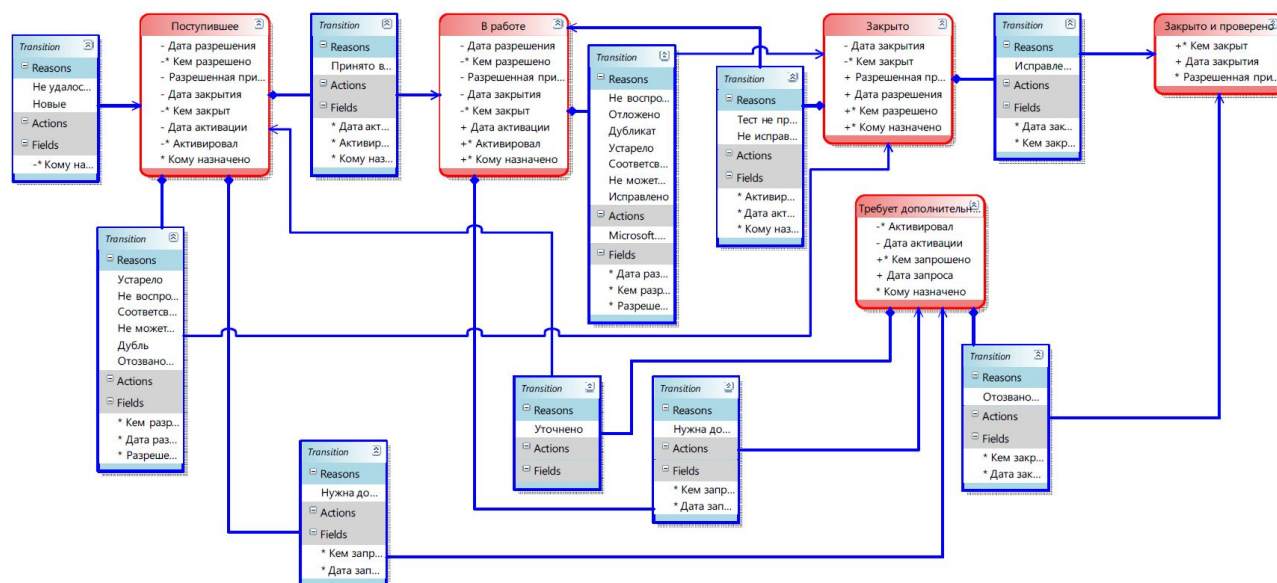


Рис. 4 Жизненный цикл рабочего элемента «Ошибка»

5.1. Создание Ошибки, состояние: Поступившее

Причина: **Новое**

Кто: Тестировщик. Рабочий элемент Ошибка появляется только как результат работы Тестировщика: при тестировании очередной сборки или при проверке запроса, поступившего от пользователей через сайт поддержки и представленного рабочим элементом **Запрос на изменение**, который создал сопровождаец (см. «Управление изменениями»).

Если рабочий элемент Ошибка создан при тестировании очередной сборки, то в нем могут быть ссылки на исполненные тестовые случаи. Состояние рабочего элемента Ошибка: **Поступившее**, исполнитель: Разработчик Задачу при тестировании которой возникла ошибка Тестировщик откатывает в **В работу**, и указывает в качестве исполнителя предыдущего разработчика.

Возможен вариант и без создания Ошибки – откатить задачу в Поступившие и назначить тому кто делал. И конечно описать проявление дефекта. Против этого вариант Макс возражает

Исполнитель: ...

5.2. Состояние: Поступившее

Кто: Разработчик. Созданные ошибки назначаются конкретным исполнителям. Возможно указывается оценка трудозатрат.

Выбрать – запрос «дай ошибки назначенные на Разработчика или не назначенные никому и находящиеся в состоянии Поступившие, можно с учетом итерации».

Исполнитель: конкретный разработчик

Атрибуты: оценка трудозатрат

5.3. Переход из Поступившее в В работе

Причина: ...

Кто: конкретный разработчик. Конкретный разработчик начинает работать над ошибкой.

Выбрать – запрос «дай ошибки назначенные на меня и находящиеся в состоянии Проступившие, можно с учетом итерации».

Исполнитель: остается конкретный разработчик

Атрибуты: оценка трудозатрат

5.4. Состояние: В работе

Кто: конкретный разработчик.

Выбрать – запрос «дай ошибки назначенные на меня и находящиеся в состоянии В работе, можно с учетом итерации».

5.5. Переход из В работе в Закрыто

Причина: ошибка исправлена

Кто: конкретный разработчик

Исполнитель: Тестировщик или конкретный тестировщик ???

5.6. Состояние: Закрыто

Кто: Тестировщик. Назначает конкретного тестировщика.

Выбор – запрос «дай все закрытые ошибки назначенные Тестировщику или не назначенные никому, можно с учетом сборки».

Исполнитель: конкретный тестировщик

5.7. Состояние: Закрыто

Кто: конкретный тестировщик.

Выбор – запрос «дай все закрытые ошибки назначенные мне, можно с учетом итерации».

5.8. Переход из Закрыто в Закрыто и проверено

Кто: конкретный тестировщик.

Причина: **Проверка пройдена**

5.9. Переход из Закрыто в Поступившие

Кто: конкретный тестировщик. Причина: **Сбой проверочного теста**

Исполнитель: Разработчик или конкретный исполнитель этой Ошибки.

Тестировщик проверяет исправление ошибки и в случае исправления меняет состояние на **Закрыто и проверено**, в случае если ошибка не исправлена – меняет состояние на **Поступившее** и назначением ее на того исполнителя, который был ранее.

6. УПРАВЛЕНИЕ ИЗМЕНЕНИЯМИ

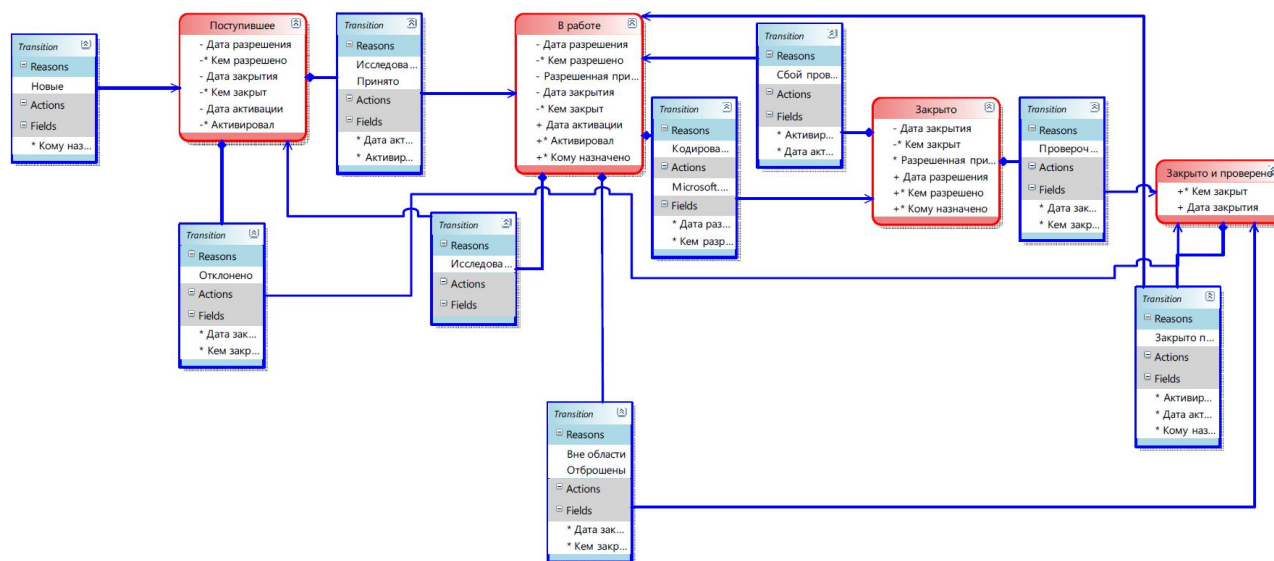


Рис. 5 Жизненный цикл рабочего элемента «Заявка на изменение»

Рабочий элемент «Заявка на изменение» служит для взаимодействия с внешним (по отношению к группе разработки) миром и обеспечивает возможность внесения изменений в продукт в процессе его сопровождения.

6.1. Создание Заявки на изменение

Состояние: **Поступившее**

Кто: создание заявки на изменение может осуществить:

- специалист службы сопровождения на основе заявки пользователя с сайта;
- менеджер или аналитик по результатам общения с заказчиком или пользователем;
- можно рассмотреть случай, когда и заказчик будет допущен к созданию.

Причина: **Новое**

6.2. Состояние: Поступившее

Кто: Менеджер. Поступившие Заявки на изменение анализируются Менеджером, осуществляется классификация Заявки на изменение (см. категории сопровождения⁵ (Categories of Maintenance) в соответствии с ГОСТ ИСО/МЭК

⁵ Категории сопровождения (Categories of Maintenance) в соответствии с ГОСТ ИСО/МЭК 14764:

Корректирующее (corrective maintenance): Реактивное изменение программного продукта, выполняемое после его поставки для корректировки обнаруженных проблем (несоответствий, ошибок). Подобные изменения исправляют программный продукт для того, чтобы он соответствовал установленным требованиям.

Адаптивное (adaptive maintenance): Изменение (модификация) программного продукта после поставки, обеспечивающее его работоспособность в измененных или изменяющихся условиях (среде). Адаптивное сопровождение обеспечивает модернизацию, позволяющую вносить изменения в эксплуатационную среду программного средства. Данные изменения должны быть внесены для сохранения работоспособности продукта в изменяемой среде. Например, операционная система может быть модернизирована и некоторые изменения должны быть внесены в продукт для его адаптации к новой операционной системе.

Профилактическое (preventive maintenance): Модификация программного продукта после поставки в целях обнаружения и корректировки имеющихся в нем скрытых ошибок для предотвращения явного проявления этих ошибок при эксплуатации данного продукта.

Совершенствующее (perfective maintenance): Модификация программного продукта после поставки для повышения его рабочих характеристик или улучшения сопровождаемости. Совершенствующее (полное или расширение функциональных возможностей) сопровождение обеспечивает модернизацию (усовершенствование) продукта в интересах пользователя, уточнение соответствующих программных документов и его перепрограммирование для улучшения рабочих характеристик, сопровождаемости и других атрибутов программного средства.

14764) и, возможно, назначение на итерацию. При необходимости Менеджер организует совещание «чрезвычайной тройки» в составе (Менеджер, Аналитик, Разработчик, Проектировщик) в рамках, которого выполняются действия по классификации и назначению итерации.

Выбрать – запрос «дай Запросы на изменение находящиеся в состоянии Проступившие, и без итерации».

Исполнитель:

Запрос на изменение, реализующий адаптивное, профилактическое, совершенствующее сопровождение, должен быть назначен Аналитiku для создания Требования (см. «Управление требованиями»).

Запрос на изменение, являющийся корректирующим, назначается Тестировщику для создания Ошибки (см. «Управление тестированием»).

«Корректирующий» запрос, назначается разработчику для исправления (Запрос выступает в роли Ошибки) или, если требуются доп. исследования может быть назначен Тестировщику для создания Ошибки (см. «Управление тестированием»).

6.3. Переход из Поступившее в В работе

Кто: Аналитик или Тестировщик берет себе.

Выбрать – запрос «дай Запросы на изменение находящиеся в состоянии Проступившие, и назначенные мне».

Исполнитель: Аналитик или Тестировщик.

6.4. Состояние: В работе

Кто: Аналитик. Осуществляет анализ и создает Требование или Задачу. После этого Запрос на изменение закрывается.

???TODO или не создает, это потребуетсa детальнее расписать потом

Кто: Тестировщик. Осуществляет анализ и создает Ошибку. . После этого Запрос на изменение закрывается.

???TODO или не создает, это потребуетсa детальнее расписать потом

Выбрать – запрос «дай Запросы на изменение находящиеся в состоянии Проступившие, и назначенные мне».

Исполнитель:

6.5. Переход из В работе в Закрыто

Кто: Аналитик или Тестировщик

Причина: Требование(я) или Ошибка(и) созданы или не созданы, т.е. Запрос на изменение отвергнут ???TODO детальнее потом.