



Сертифицированный тестировщик

Программа обучения Продвинутого уровня

Руководитель тестирования

Версия 2012 RU-2

International Software Testing Qualifications Board



Уведомление об авторских правах

Уведомление об авторских правах © International Software Testing Qualifications Board (далее просто ISTQB®)

Авторские права © 2017 авторы перевода 2012 Маргарита Трофимова (руководитель группы), Александр Александров (редактор), Андрей Конушин, Елена Костина, Александр Мешков, Александра Титова.

Рабочая группа Продвинутого уровня Rex Black (председатель), Judy McKay (вице-председатель), Graham Bath, Debra Friedenberg, Bernard Homès, Kenji Onishi, Mike Smith, Geoff Thompson, Tsuyoshi Yumoto; 2010-2012.

История изменений

Версия	Дата	Комментарии
ISEB v1.1	04 сентября 2001	ISEB Practitioner Syllabus
ISTQB 1.2E	Сентябрь 2003	ISTQB Advanced Level Syllabus из EOQ-SG
V2007	12 октября 2007	Certified Tester Advanced Level syllabus version 2007
D100626	26 июня 2010	Внесение изменений, принятых в 2009 году, разделение глав на отдельные модули
D101227	27 декабря 2010	Принятие изменений в формате и правок, не влияющих на суть содержания
D2011	31 октября 2011	Изменения по разделению программы обучения, переработаны цели обучения и текст для согласования с целями. Добавлены бизнес-цели.
Alpha 2012	09 февраля 2012	Внесение комментариев от национальных коллегий по октябрьскому релизу.
Beta 2012	26 марта 2012	Внесение комментариев от национальных коллегий по альфа-релизу.
Beta 2012	07 апреля 2012	Бета-версия для GA
Beta 2012	08 июня 2012	Отредактированная версия для национальных коллегий
Beta 2012	27 июня 2012	Внесены комментарии от EWG и по Глоссарию
RC 2012	15 августа 2012	Релиз-кандидат – включены финальные правки от национальных коллегий
GA 2012	19 октября 2012	Финальные правки для релиза GA
RSTQB 2017	30 марта 2017	Перевод на русский язык
2012 RU-2	12 мая 2024	Приведение к шаблону ISTQB v4.0 Обновление логотипа RSTQB

Содержание

Уведомление об авторских правах	2
Благодарности	7
0. Предисловие к программе обучения	8
0.1. Цель документа	8
0.2 Обзор.....	8
0.3 Цели обучения, доступные проверке	8
1. Процесс тестирования – 420 минут	9
1.1 Введение	10
1.2 Планирование, мониторинг и контроль тестирования	10
1.2.1. Планирование тестирования	10
1.2.2 Мониторинг и контроль тестирования	12
1.3 Анализ тестирования	12
1.4 Проектирование тестов.....	14
1.5 Реализация тестов	15
1.6 Выполнение тестов	16
1.7 Определение критериев выхода и отчетности	16
1.8 Активности по завершению тестирования	17
2. Управление тестированием – 750 минут	19
2.1 Введение	20
2.2 Управление тестированием в контексте	21
2.2.1 Понятие заинтересованных лиц тестирования.....	21
2.2.2 Дополнительные активности жизненного цикла разработки программного обеспечения и рабочие продукты	22
2.2.3 Согласование активностей тестирования и других активностей жизненного цикла	23
2.2.4 Управление нефункциональным тестированием	26
2.2.5 Управление тестированием, основанным на опыте.....	27
2.3 Тестирование, основанное на рисках, и другие подходы приоритизации тестов и распределения трудозатрат.....	27
2.3.1 Тестирование, основанное на рисках	28
2.3.2 Методы тестирования, основанного на рисках	32
2.3.3 Другие методы для выбора тестов	36
2.3.4 Приоритизация тестов и распределение усилий в процессе тестирования	37
2.4 Тестовая документация и другие рабочие продукты	38
2.4.1 Политика тестирования.....	38
2.4.2. Стратегия тестирования.....	39

2.4.3. Главный план тестирования	41
2.4.4. Уровневый план тестирования	42
2.4.5. Управление рисками проекта	42
2.4.6 Другие рабочие продукты тестирования	43
2.5 Оценка затрат на тестирование	44
2.6 Определение и использование метрик тестирования	45
2.7 Бизнес-ценность тестирования	51
2.8 Распределенное, аутсорсинговое и внутреннее тестирование	52
2.9 Управление на основе промышленных стандартов	53
3. Рецензирование – 180 минут	55
3.1 Введение	55
3.2 Управленческие рецензирования и аудиты	57
3.3 Управление рецензированием	57
3.4 Метрики рецензирования	59
3.5 Управление формальным рецензированием	60
4. Управление дефектами – 150 минут	62
4.1 Введение	62
4.2 Жизненный цикл дефекта и жизненный цикл разработки программного обеспечения	62
4.2.1 Жизненный цикл дефекта и его состояния	63
4.2.2 Управление отчетами о неверных дефектах или дубликатах	64
4.2.3 Кросс-функциональное управление дефектами	64
4.3 Информация, содержащаяся в отчете о дефекте	65
4.4 Оценка возможности процесса с помощью отчетов о дефектах	66
5. Улучшение процесса тестирования – 135 минут	68
5.1 Введение	68
5.2 Процесс улучшения тестирования	69
5.2.1 Введение в улучшение процесса	69
5.2.2 Типы улучшений процесса	69
5.3 Улучшение процесса тестирования	70
5.4 Улучшение процесса тестирования с TMMi	71
5.5 Улучшение процесса тестирования с TPI Next	72
5.6 Улучшение процесса тестирования с CTP	72
5.7 Улучшение процесса тестирования со STEP	73
6. Инструменты для тестирования и автоматизация – 135 минут	74
6.1 Введение	74
6.2 Выбор инструмента	74

6.2.1	Инструменты с открытым исходным кодом	74
6.2.2	Инструменты, сделанные на заказ.....	75
6.2.3	Возврат инвестиций (ROI).....	76
6.2.4	Процесс выбора	77
6.3	Жизненный цикл инструмента тестирования.....	79
6.4	Метрики инструмента тестирования.....	80
7.	Командные навыки – Формирование команды – 210 минут	81
7.1	Введение	81
7.2	Личные навыки.....	82
7.3	Развитие команды тестирования	83
7.4	Внедрение тестирования в организации	85
7.5	Мотивация	87
7.6	Коммуникации	88
8.	Ссылки.....	90
8.1	Стандарты	90
8.2	Документы ISTQB	90
8.3	Торговые знаки	90
8.4	Книги	91
8.5	Прочие ссылки	91

Благодарности

Перевод версии документа 2012 года выполнен Рабочей Группой Продвинутого уровня Russian Software Testing Qualifications Board: Маргарита Трофимова (руководитель группы), Александр Александров (редактор).

Благодарим команду переводчиков: Андрей Конушин, Елена Костина, Александр Мешков, Александра Титова

Этот документ был создан рабочей группой Программы обучения Руководитель тестирования Продвинутого уровня ISTQB: Rex Black (Председатель), Judy McKay (Заместитель председателя), Graham Bath, Debra Friedenberg, Bernard Homès, Paul Jorgensen, Kenji Onishi, Mike Smith, Geoff Thompson, Erik van Veenendaal, Tsuyoshi Yumoto.

Рабочая группа благодарит команду редакторов и национальные коллегии за их вклад и замечания.

Состав рабочей группы «Руководитель тестирования Продвинутого уровня» на момент завершения программы обучения продвинутого уровня состоял из (в алфавитном порядке): Graham Bath, Rex Black, Maria Clara Choucair, Debra Friedenberg, Bernard Homès (Заместитель председателя), Paul Jorgensen, Judy McKay, Jamie Mitchell, Thomas Mueller, Klaus Olsen, Kenji Onishi, Meile Posthuma, Eric Riou du Cosquer, Jan Sabak, Hans Schaefer, Mike Smith (Председатель), Geoff Thompson, Erik van Veenendaal, Tsuyoshi Yumoto.

В редактировании и предоставлении замечаний принимали участие: Chris van Bael, Graham Bath, Kimmo Nakala, Rob Hendriks, Marcel Kwakernaak, Rik Marselis, Don Mills, Gary Mogyorodi, Thomas Mueller, Ingvar Nordstrom, Katja Piroué, Miele Posthuma, Nathalie Rooseboom de Vries, Geoff Thompson, Jamil Wahbeh, Hans Weiberg.

Документ был официально выпущен Генеральной Ассамблеей ISTQB 19 октября 2012 года.

0. Предисловие к программе обучения

0.1. Цель документа

Программа обучения представляет собой основу международной сертификации на квалификацию «Руководитель тестирования Продвинутого уровня» в области тестирования программного обеспечения. ISTQB предоставляет программу обучения:

1. Национальным коллегиям для перевода на национальный язык и аккредитации организаторов обучения. Национальные коллегии могут приспособить программу обучения к особенностям конкретных языков и определить ссылки для адаптации к местным публикациям.
2. Экзаменационным комиссиям для формирования экзаменационных вопросов на национальном языке, адаптированные к целям обучения каждого курса.
3. Организаторам обучения для разработки учебной программы и определения соответствующих методов обучения.
4. Кандидатам на получение сертификатов для подготовки к экзамену (в рамках программы обучения или независимо).
5. Международному сообществу разработки ПО и систем для повышения уровня профессионализма при тестировании ПО и систем, и использования как основы для книг и статей.

ISTQB может разрешить другим лицам использовать эту программу обучения в своих целях при условии, что они обратятся за письменным разрешением.

0.2 Обзор

Продвинутый уровень состоит из трех отдельных программ обучения:

- Руководитель тестирования
- Тест-аналитик
- Технический тест-аналитик

Документ «Обзор продвинутого уровня ISTQB» включает следующую информацию:

- Результат деловой деятельности для каждой программы обучения
- Обзор программы обучения
- Связи между программами обучения
- Описание познавательных уровней (К-уровни)
- Приложения

0.3 Цели обучения, доступные проверке

Цели обучения согласуются с бизнес-результатами и используются при сдаче экзамена для сертификации руководителей тестирования Продвинутого уровня. В большинстве случаев все части этой программы обучения доступны проверке на уровне К1. Цели обучения на уровнях К2, К3 и К4 отображаются в начале соответствующей главы.

1. Процесс тестирования – 420 минут

Ключевые слова

Критерии выхода; тестовые сценарии; завершение тестирования; тестовое условие; контроль тестирования; проектирование тестов; выполнение тестов; реализация тестов; протокол тестирования; планирование тестирования; процедура тестирования; автоматизированный сценарий тестирования; итоговый отчет о тестировании.

Задачи обучения процессу тестирования

1.2 Планирование, мониторинг и контроль тестирования

ТМ-1.2.1 (К4) Проанализировать потребности системы в тестировании для того, чтобы спланировать активности по тестированию и работу с продуктом, которые позволят достигнуть целей тестирования.

1.3. Анализ тестирования

ТМ-1.3.1 (К3) Использовать трассируемость для проверки полноты и целостности определенных условий тестирования в отношении объектов тестирования, стратегии тестирования и плана тестирования.

ТМ-1.3.2 (К2) Определить факторы, которые могут повлиять на уровень детализации, на котором тестовые условия могут быть определены, а также преимущества и недостатки определения тестовых условий на детальном уровне.

1.4 Проектирование тестов

ТМ-1.4.1 (К3) Использовать трассируемость для проверки полноты и целостности разработанных сценариев тестирования в отношении определенных тестовых условий.

1.5 Реализация тестов

ТМ-1.5.1 (К3) Использовать риски, приоритизацию, тестовое окружение, зависимости данных и ограничения для разработки графика выполнения тестов, которые являются полными и целостными в отношении целей тестирования, стратегии тестирования и плана тестирования.

1.6 Выполнение тестов

ТМ-1.6.1 (К3) Использовать трассируемость для отслеживания хода тестирования в отношении полноты и целостности с целями тестирования, стратегией тестирования и планом тестирования.

1.7 Оценка критериев выхода и разработка отчетности

ТМ-1.7.1 (К2) Объяснить важность сбора точной и достоверной информации во время процесса тестирования для поддержки достоверной отчетности и оценки критериев выхода.

1.8 Активности по завершению тестирования

ТМ-1.8.1 (К2) Кратко изложить четыре группы активностей по завершению тестирования.

ТМ-1.8.2 (К3) Реализовать ретроспективу проекта для оценки процессов и улучшаемых областей.

1.1 Введение

Программа обучения «Базовый уровень ISTQB» описывает базовый процесс тестирования, который включает следующие активности:

- планирование и контроль;
- анализ и проектирование;
- реализация и выполнение;
- оценка критериев выхода и отчетность;
- Активности по завершению тестирования.

Программа обучения «Базовый уровень» указывает, что мероприятия процесса могут совмещаться или происходить параллельно, хотя логически последовательны. Обычно требуется привязка основных мероприятий в рамках системы и проекта.

Для дополнительного уточнения и оптимизации процесса, большего соответствия жизненному циклу разработки ПО, содействию мониторингу и контролю тестирования в рамках Продвинутого уровня некоторые из этих активности рассматриваются по отдельности. К активностям относятся:

- планирование, мониторинг и контроль;
- анализ;
- проектирование;
- реализация;
- выполнение;
- оценка критериев выхода и отчетность;
- активности по завершению тестирования.

1.2 Планирование, мониторинг и контроль тестирования

В этом разделе уделяется особое внимание процессам планирования, мониторинга и контроля тестирования. Как отмечалось на Базовом уровне, эти действия относятся к роли руководителя тестирования.

1.2.1. Планирование тестирования

Для каждого уровня тестирования планирование начинается в начале процесса тестирования и продолжается на протяжении проекта вплоть до завершения активностей в рамках уровня. Оно включает в себя определение активностей и ресурсов, необходимых для достижения целей и задач, определенных в стратегии тестирования. Планирование тестирования также включает выявление методов сбора и отслеживания метрик, которые будут использоваться при управлении проектом, определение степени соответствия плану и оценке достижения целей. При определении важных метрик на этапах планирования можно выбрать инструменты, запланировать график обучения и разработать примерную документацию.

Стратегия (или стратегии) тестирования, выбранные для проекта, помогает выявить задачи, которые нужно выполнить в ходе этапов планирования. Например, при использовании стратегии на основе рисков (см. Главу 2) анализ рисков используется для управления процессом планирования тестирования с целью снижения идентифицированных продуктовых рисков и для помощи в планировании на случай чрезвычайных ситуаций. Если выявлено большое число вероятных и потенциально серьезных дефектов, связанных с безопасностью, то нужно приложить серьезные усилия к тестированию безопасности. Кроме того, если будет установлено, что в спецификации проектирования обнаружены серьезные дефекты, то процесс планирования тестирования может

привести к дополнительному статическому тестированию (рецензированию) спецификации проектирования.

Информация о риске может также быть использована для определения приоритетов различных активностей по тестированию. Например, в тех случаях, когда производительность системы является риском высокого уровня, то тестирование производительности может проводиться сразу, как только интегрированный код станет доступным. Похожим образом, если применяется реактивная стратегия, то планирование создания концепции тестирования и программных средств динамического тестирования, таких как исследовательское тестирование, может быть оправдано.

Кроме того, на этапе планирования тестирования руководитель тестирования четко формулирует подход к тестированию, который включает задействованные уровни тестирования, цели и задачи каждого уровня, а также методы тестирования, используемые на каждом уровне тестирования. Например, при тестировании, основанном на риске, для авиационных электронных систем оценка риска предусматривает требуемый уровень покрытия кода и, следовательно, определяет, какие методы тестирования могут быть использованы.

Могут существовать сложные связи между базисом тестирования (например, конкретными требованиями или рисками), тестовыми условиями и тестами, которые покрывают их. В рабочем продукте часто существуют связи «многие-ко-многим». Это необходимо понимать для осуществления эффективного мониторинга, планирования и контроля тестирования. Выбор инструмента также зависит от понимания принципов работы ПО.

Связи могут также существовать между результатами, полученными командой разработчиков и командой тестировщиков. Например, матрица трассируемости может нуждаться в отслеживании связей между элементами детальной спецификации проектирования проектировщиков системы, бизнес-требованиями бизнес-аналитиков и результатами тестирования, полученными командой тестировщиков. Если должны быть спроектированы и использованы низкоуровневые сценарии тестирования, то требование, определенное на этапе планирования, детализированное в документах для команды разработки, должно быть согласовано перед созданием сценариев тестирования. При следовании жизненному циклу на основе методологии Agile неформальные активности по передаче информации могут использоваться для взаимодействия между командами перед началом тестирования.

В плане тестирования возможно перечисление конкретных свойств ПО, которые входят в область тестирования (основанное на анализе рисков, если необходимо), а также явно определенных свойств, которые выходят за рамки области тестирования. В зависимости от уровня формализации и соответствующей документации проекта каждая функция может быть связана с соответствующей спецификацией проектирования тестов.

На этом этапе может также существовать требование к руководителю тестирования работать с архитектором проекта для определения начальной спецификации тестового окружения, проверки доступности требуемых ресурсов, гарантий того, что люди, которые будут конфигурировать окружение, заинтересованы в этом и имеют представление о стоимости работ и сроках, требуемых для подготовки и настройки тестового окружения.

Наконец, должны быть определены все внешние зависимости, связанные с соглашением об уровне предоставления услуг (SLA) и, если необходимо, должен быть установлен начальный контакт. Примерами зависимостей являются запросы на ресурсы к внешним группам, которые зависят от других проектов (работающих в рамках программы), внешних разработчиков ПО или партнеров по разработке, команды внедрения и администраторов баз данных.

1.2.2 Мониторинг и контроль тестирования

Для обеспечения эффективного контроля руководителю тестирования необходимо создать график тестирования и схему мониторинга. Схема должна включать подробные метрики и цели, которые должны связать статус рабочих продуктов тестирования, запланированные активности и стратегические цели.

Для небольших и несложных проектов может быть относительно легко связать рабочие продукты тестирования, запланированные активности и стратегические цели. Обычно для достижения этого нужно определить более детальные цели. Они могут включать метрики и критерии, соответствующие целям тестирования и покрытию базиса тестирования.

Важно иметь возможность связать статус рабочих продуктов тестирования и активностей, связанных с базисом тестирования, способом, который является понятным и подходящим для участников проекта и участников со стороны бизнеса.

Определение целей и измерение прогресса на основе тестовых условий и групп тестовых условий можно использовать в качестве способа их достижения, связывая рабочие продукты тестирования ПО с базисом тестирования при помощи тестовых условий. Трассируемость, настроенная таким образом, чтобы получать отчет о её статусе, делает сложные связи, существующие между результатами разработки, базисом тестирования и рабочими продуктами по тестированию ПО, более прозрачными и доступными для восприятия.

Иногда детальные измерения и цели, которые участники хотят мониторить, напрямую не связаны с функциональностью или спецификацией, особенно если документации мало или она вовсе отсутствует. Например, участник со стороны бизнеса может быть больше заинтересован в создании покрытия за счет операционного бизнес-цикла, хотя спецификация определена в терминах функциональности системы. Вовлеченность участников проекта со стороны бизнеса на ранних этапах проекта может помочь определить критерии оценки и цели, которые можно использовать не только для обеспечения лучшего контроля проекта, но также для актуализации и определения дополнительных активностей тестирования в рамках всего проекта. Например, критерии оценки участников и цели могут привести к структуризации проектирования тестов и реализации тестирования рабочего продукта и/или проверке графика выполнения тестирования ПО для облегчения точного мониторинга хода тестирования для этих критериев оценки. Эти цели также помогают обеспечить отслеживаемость конкретного уровня тестирования и имеют возможность предоставлять информацию о связи между различными уровнями тестирования.

Контроль тестирования — это постоянная активность. Она включает сравнение с планом фактического прогресса выполнения активностей по тестированию и устранение недостатков, когда это необходимо. Контроль тестирования позволяет управлять тестированием для выполнения целей, стратегии и задач, включая при необходимости активности по планированию тестирования. Предпринимаемый анализ управляющих данных зависит от детализации информации в предполагаемой информации.

Содержание документов по планированию тестирования и активностям по контролю рассматривается в Главе 2.

1.3 Анализ тестирования

В программе обучения «Базовый уровень» изложен анализ тестирования и проектирования тестов. В отличие от него «Продвинутый уровень» рассматривает их как отдельные активности, учитывая, что они могут быть реализованы параллельно, интегрированно или итерационно для упрощения выполнения проектирования тестов рабочего продукта.

Глоссарий определяет термин «анализ тестирования» как процесс анализа базиса тестирования и определения целей тестирования. Анализ тестирования – это активность, которая определяет в виде тестовых условий, что должно быть протестировано. Тестовые условия можно определить при помощи анализа базиса тестирования, целей тестирования и рисков продукта. Их можно рассматривать как подробные метрики и цели для достижения успеха (например, как часть критериев выхода). Должна прослеживаться их связь с базисом тестирования и определенными стратегическими целями, включающими цели тестирования и другие критерии успеха проекта или участников проекта. Тестовые условия должны также быть прослеживаемыми перед проектированием тестов и в других тестовых рабочих продуктах, если эти рабочие продукты уже созданы.

Анализ тестирования для конкретного уровня можно выполнить, как только для этого уровня тестирования будет подготовлен базис тестирования. Методы формального тестирования и другие общие аналитические методы (например, аналитические стратегии, основанные на риске и на требованиях) можно использовать для определения тестовых условий. Тестовые условия могут (или не могут) определять значения или переменные в зависимости от уровня тестирования, информации, доступной во время выполнения анализа и выбранной степени детализации (например, степень детализации документации).

Существует некоторое число факторов, которые надо учесть, чтобы определить уровень детализации тестовых условий. Они включают:

- уровень тестирования;
- уровень детализации и качество базиса тестирования;
- сложность системы/ПО;
- продуктовые и проектные риски;
- связь между базисом тестирования (что должно быть протестировано и как это должно быть протестировано);
- использование жизненного цикла разработки ПО;
- использование инструмента по управлению тестированием;
- уровень, на котором проектирование тестов и другие рабочие продукты должны быть определены и документированы;
- знания и навыки для анализа тестирования;
- уровень зрелости процесса тестирования и организации (заметим, что высокая зрелость может потребовать высокой степени детализации или позволить низкую степень детализации);
- доступность консультации других участников проекта.

Определение тестовых условий в развернутом виде может привести к росту числа этих условий. Например, у вас может быть одно общее тестовое условие «Контроль тестирования» приложения для электронной коммерции. Однако в подробных тестовых условиях оно может быть разделено на тестовые условия со сложной структурой, с условиями для каждого поддерживаемого метода платежа, возможной страны назначения и так далее.

Определение тестовых условий на детальном уровне имеет следующие преимущества:

- Обеспечивает большую гибкость других работ по тестированию ПО (например, сценариев тестирования) относительно базиса и целей тестирования, обеспечивая таким образом более подробный контроль со стороны руководителя тестирования.
- Вносит вклад в предотвращение дефектов, как указывалось ранее в «Базовом уровне», которые могут появиться на ранней стадии проекта для высокоуровневого тестирования,

- как только будет готов базис тестирования, и, возможно, до того, как станут доступны системная архитектура и детальный дизайн.
- Связывает тестирование рабочих продуктов с участниками проекта в терминах, которые они могут понимать. Часто сценарии тестирования и другие рабочие продукты тестирования ничего не означают для участников проекта со стороны бизнеса. Простые метрики, такие как количество выполняемых сценариев тестирования, ничего не дают для понимания полноты покрытия требований всем участникам проекта.
- Помогает влиять и управлять проведением не только активностей по тестированию, но также других активностей по разработке ПО.
- Позволяет проектировать, реализовывать и выполнять тесты наряду с результатами рабочих продуктов, которые будут оптимизированы более эффективным покрытием из детальных измерений и целей.
- Обеспечивает основу более понятной горизонтальной трассируемости для уровня тестирования.

Недостатки определения тестовых условий на детальном уровне:

- Временные затраты.
- Поддержка может стать затруднительной в изменяющемся окружении.
- Должна быть определена степень формализации и реализации внутри команды.

Спецификация детальных тестовых условий может быть особенно эффективной в следующих ситуациях:

- Упрощенные методы документирования при проектировании тестов. Чек-листы используются для корректировки жизненного цикла разработки, стоимости и/или временных ограничений и других факторов.
- Малое количество или отсутствие формальных требований. Другие рабочие продукты разработки доступны в качестве базиса тестирования.
- Проект является крупномасштабным, сложным или высоко рискованным и требует соответствующего уровня мониторинга и контроля, который не может быть достигнут простым соотношением сценариев тестирования с разработкой программного продукта.

Тестовые условия могут быть определены с меньшей степенью детализации, когда базис тестирования легко и прямо связан с проектированием тестов программного продукта:

- Тестирование на уровне отдельных компонентов.
- Менее сложные проекты, где существует простая иерархическая связь между тем, что должно быть протестировано и тем, как это должно быть протестировано.
- Приемочное тестирование, в котором пользовательские тестовые сценарии могут быть использованы в помощь для определения тестов.

1.4 Проектирование тестов

Проектирование тестов – это активность, которая определяет, что и как должно быть протестировано. Оно включает определение сценариев тестирования при помощи пошаговой разработки определенных тестовых условий или базиса тестирования, использующих методы тестирования, определенных в стратегии тестирования и/или плане тестирования.

В зависимости от подходов, использованных при мониторинге тестирования, контроле тестирования и трассируемости, сценарии тестирования должны быть связаны напрямую (или не напрямую через

тестовые условия) с базисом тестирования и определенными целями. Эти задачи включают стратегические цели, цели тестирования и другие критерии успешности проекта или их участников.

Когда определены тестовые условия и достаточно доступной информации для разработки высокоуровневых и низкоуровневых сценариев тестирования, можно выполнить проектирование тестов для данного уровня тестирования в соответствии с используемым подходом к проектированию тестов. Более вероятно, что для высоких уровней тестирования проектирование тестов – это отдельная активность, выполняемая перед анализом тестирования. Вероятно, что для низких уровней тестирования анализ и проектирование будут выполняться как единый процесс.

Также бывает, что некоторые задачи, которые обычно выполняются при реализации тестов, могут быть интегрированы в процесс проектирования тестов, когда используется итеративный подход создания тестовых требований для их выполнения (например, создание тестовых данных). Фактически этот подход может оптимизировать покрытие тестовых условий, создавая в процессе низкоуровневые или высокоуровневые сценарии тестирования.

1.5 Реализация тестов

Реализация тестов – это активность, во время которой тест-аналитик организует и приоритизирует тесты. В формально документированном контексте реализация тестов – это процесс, во время которого разработка тестов выполняется в виде конкретных сценариев тестирования, процедур тестирования и тестовых данных. Некоторые организации следуют стандарту IEEE 829 [IEEE829], определяющему входные данные и связанные результаты в спецификациях сценариев тестирования и шаги теста в спецификациях тестирования. Чаще всего входные данные теста, ожидаемые результаты и шаги тестового сценария документируются одновременно. Реализация тестов также включает создание хранимых тестовых данных (например, в плоских файлах или в таблицах базы данных).

Реализация тестов также включает окончательную проверку готовности команды тестировщиков к выполнению тестов. Проверки могут включать обеспечение предоставления необходимого тестового окружения, тестовых данных и кода (возможно, запускающего тестовое окружение и/или приёмочные тесты кода). Все сценарии тестирования записаны, рецензированы и готовы к выполнению. Это может также включать проверку уровня тестирования на предмет соответствия явным и неявным критериям входа (см. Раздел 1.7). Реализация тестов может также включать разработку подробного описания тестового окружения и тестовых данных.

Уровень детализации и связанная сложность работы, выполняемая во время реализации тестов, может влиять на детализацию при тестировании ПО (например, сценарии тестирования и тестовые условия). В некоторых случаях, особенно когда тесты необходимо сохранять для долгосрочного повторного использования при регрессионном тестировании, тесты должны содержать подробное описание шагов, необходимых для исполнения тестов. Если применяются определенные правила, то в тестах должны предоставляться подтверждение соблюдения применяемых стандартов.

Порядок во время реализации тестов, в котором ручные и автоматизированные тесты могут быть запущены, должен быть включен в график тестирования. Руководитель тестирования должен внимательно проверять ограничения, включая риски и приоритеты, что может потребовать запускать тесты в определенном порядке или на определенном оборудовании. Зависимость от тестового окружения или тестовых данных должна быть известна и проверена.

У ранней реализации тестов может быть несколько недостатков. Например, в случае гибкой (Agile) разработки, код может сильно измениться от итерации к итерации, делая работу по реализации тестов ненужной. Даже без жизненного цикла изменений гибкой (Agile) разработки, любой

Итерационный или инкрементальный жизненный цикл может привести к серьезным изменениям между итерациями, делая подготовленные ранее тесты неактуальными. То же самое справедливо по отношению к плохо управляемым последовательным жизненным циклам, где требования часто изменяются, особенно на завершающих этапах проекта. Перед осуществлением большого объема работ по реализации тестов полезно понимать жизненный цикл разработки ПО и изменимость функций ПО, которые будут доступны для тестирования.

У ранней реализации тестов может быть несколько преимуществ. Например, конкретные тесты дают рабочие примеры того, как программа должна вести себя, если они были написаны в соответствии с базисом тестирования. Эксперты в области бизнеса, вероятно, найдут для себя верификацию конкретных тестов более легкой, чем верификацию абстрактных бизнес-правил, благодаря чему смогут в дальнейшем идентифицировать слабые места в требованиях к ПО. Такая верификация тестов может обеспечивать наглядную иллюстрацию требуемого поведения системы для разработчиков и проектировщиков.

1.6 Выполнение тестов

Выполнение тестов начинается после того, как объект тестирования предоставлен и соблюдены критерии начала выполнения тестов. Тесты должны быть разработаны или, по крайней мере, определены до начала тестирования. Инструменты должны быть в наличии, особенно для управления тестированием, отслеживания дефектов и (если применимо) автоматизации тестирования. Отслеживание результатов тестирования, включая отслеживание метрик, должно выполняться и быть понятным для всех членов команды. Стандарты протоколирования тестирования и заведения дефектов должны быть общедоступны. Выполнение всех этих пунктов до начала выполнения тестов позволит повысить эффективность процесса тестирования.

Тесты должны быть выполнены в соответствии со сценариями тестирования, хотя руководитель тестирования должен рассмотреть возможность расширения объемов тестирования, так как тестировщик может выполнять дополнительно интересные сценарии тестирования и действия, в зависимости от поведения системы при тестировании. При соблюдении стратегии тестирования, которая, по меньшей мере, частично, является реактивной, должно быть зарезервировано некоторое количество времени для проведения тестирования, использующего методы, основанные на опыте и дефектах. Конечно, любой сбой, обнаруженный при импровизированном тестировании, должен быть зафиксирован в сценарии тестирования, который необходимо выполнить для воспроизведения сбоя системы. Автоматизированные тесты следуют выполнять согласно инструкциям без отклонений.

Главная роль руководителя тестирования при выполнении тестов – это отслеживание прогресса относительно плана тестирования и, если необходимо, быть инициатором и осуществлять контроль успешности выполнения тестирования с точки зрения целей, соответствия стратегии и решения основных задач. Для выполнения этого руководитель тестирования может использовать трассируемость результатов тестирования относительно тестовых условий, базиса тестирования и в конечном счете целей тестирования, и именно из достижения целей тестирования формируется результат тестирования. Этот процесс детально описан в Разделе 2.6.

1.7 Определение критериев выхода и отчетности

Документация и отчетность по мониторингу и контролю прогресса по тестированию подробно рассматривается в Разделе 2.6.

С точки зрения процесса тестирования очень важно, чтобы эффективные процессы предоставляли информацию, необходимую для определения критериев выхода и составления отчетности.

Определение требований к информации и методам сбора - часть планирования тестирования, мониторинга и контроля. Во время тестового анализа, проектирования, реализации и выполнения тестов руководитель тестирования должен убедиться в том, что члены команды тестировщиков, ответственные за эти этапы, предоставляют требуемую информацию точно и своевременно для облегчения выполнения оценки и составления отчетности.

Частота и степень детализации, требуемая для разработки отчетности, зависит от проекта и организации. Во время планирования тестирования об этом необходимо договориться, предварительно проведя консультации с соответствующими заинтересованными сторонами, принимающими участие в проекте.

1.8 Активности по завершению тестирования

Как только тестирование полностью выполнено, ключевые результаты необходимо передать соответствующему заинтересованному специалисту или сохранить для последующей передачи. Все вместе это – активности по завершению тестирования. Они разделяются на четыре главные группы:

- 1) Проверка завершения тестирования – подтверждение, что все работы по тестированию действительно завершены. Например, все запланированные тесты могут быть или выполнены, или по желанию частично не использованы, все открытые дефекты должны быть или исправлены и протестированы, или перенесены в срок будущего релиза, или принимаются в качестве ограничений тестирования.
- 2) Передача артефактов тестирования – предоставление ценных рабочих продуктов всем, кому они необходимы. Например, найденные дефекты должны быть переданы тем, кто будет в дальнейшем использовать и поддерживать систему. Тесты и тестовое окружение нужно предоставить лицам, ответственным за тестирование в период поддержки. Наборы регрессионных тестов (автоматизированных или ручных) должны быть задокументированы и переданы команде поддержки.
- 3) Накопленный опыт – участие в ретроспективных встречах, где важные уроки (начиная с начала проекта по тестированию и в течение всего жизненного цикла) должны быть зафиксированы. На этих встречах происходит разработка планов для того, чтобы наиболее успешные практики можно было бы повторно использовать, плохих практик можно было бы избежать или, если проблемы невозможно решить, учесть их в рамках проектных планов. Рассматриваемые области включают следующие:
 - a) Было ли у пользователя представление о процессе анализа риска качества достаточно широким? Например, из-за позднего обнаружения непредвиденного кластера дефектов команда смогла понять, что в активностях по анализу рисков качества для будущих проектов должна принимать участие более широкая группа представителей пользователей.
 - b) Были ли оценки точными? Например, они могли быть в значительной степени неточными. Следовательно, будущие активности по оценке должны учитывать предыдущий опыт (например, тестирование было неэффективным или оценка была фактически ниже, чем должна была быть).
 - c) Каковы тенденции, результаты отказов, эффективность анализа дефектов? Например, если запрос на изменение пришел поздно и повлиял на качество анализа и разработки, то как это повлияло на тенденции, которые показывают неэффективные методы работы (например, пропуск уровня тестирования, на котором можно было бы выявить дефекты раньше), что более эффективно с точки зрения затрат на тестирование. Проверить, связаны ли тенденции работы с дефектами с такими областями, как новые технологии, изменения персонала или недостаток квалификации.
 - d) Существуют ли потенциальные возможности улучшения процесса?

- е) Были ли в плане неожиданные изменения, которые нужно учитывать при будущем планировании?
- 4) Формализация рабочих продуктов, протоколов, отчетов и других документов в системе управления конфигурацией. Например, план тестирования и план проекта могут храниться в архиве планирования с возможностью привязки к системе и версии, для которой они используются.

Эти задачи являются важными, но их часто не замечают, поэтому они должны быть четко указаны в плане тестирования. Это характерно для одной или несколько задач, как правило, из-за перехода специалиста на другой проект, увольнения члена команды, загруженности графика работы или повышенной загрузки команды. По таким проектам, выполняемым в рамках контракта (например, заказная разработка), контракт должен определять задачи, требуемые для выполнения.

2. Управление тестированием – 750 минут

Ключевые слова

Уровневый план тестирования; главный план тестирования; риск продукта; риск проекта; риск качества; риск; анализ риска; оценка риска; определение риска; уровень риска; управление риском; смягчение риска; тестирование, основанное на рисках; подход к тестированию; тестовое условие; контроль тестирования; директор по тестированию; оценка затрат на тестирование; ведущий специалист по тестированию; уровень тестирования; управление тестированием; мониторинг тестирования; план тестирования; политика тестирования; стратегия тестирования; широкополосный оракул.

Цели изучения Управления тестированием

2.2 Управление тестированием в контексте

ТМ-2.2.1 (К4) Провести анализ заинтересованных лиц, обстоятельств и потребностей программного проекта или программы, включая модель жизненного цикла разработки программного обеспечения, и определить оптимальные задачи тестирования.

ТМ-2.2.2 (К2) Понять, как активности и рабочие продукты жизненного цикла разработки программного обеспечения влияют на тестирование, и как тестирование влияет на активности и рабочие продукты жизненного цикла разработки программного обеспечения.

ТМ-2.2.3 (К2) Объяснить способы управления проблемами в управлении тестированием, связанными с тестированием на основе опыта и нефункциональным тестированием.

2.3 Тестирование, основанное на рисках и другие подходы для приоритизации тестов

ТМ-2.3.1 (К2) Объяснить различные способы того, как тестирование, основанное на рисках, реагирует на риски.

ТМ-2.3.2 (К2) Объяснить на примерах различные методы анализа риска продукта

ТМ-2.3.3 (К4) Проанализировать, выявить и оценить риск качества продукта, суммируя риски и оцениваемый уровень риска, основанный на мнениях ключевых заинтересованных лиц проекта.

ТМ-2.3.4 (К2) Описать, как выявленные риски качества продукта могут быть смягчены и управляемы в соответствии с оцениваемым уровнем их риска на протяжении жизненного цикла и процесса тестирования.

ТМ-2.3.5 (К2) Привести примеры выбора различных вариантов тестирования, определения приоритетов тестирования и распределения усилий.

2.4 Тестовая документация и другие рабочие продукты

ТМ-2.4.1 (К4) Проанализировать предоставленные примеры политик и стратегий тестирования, и создать главный план тестирования, уровневый план тестирования, и другие завершённые и соответствующие этим документам рабочие продукты тестирования.

ТМ-2.4.2 (К4) Для предоставленного проекта проанализировать его риски и выбрать соответствующие варианты управления рисками (т.е. смягчение последствий, корректировка, передача и/или принятие).

TM-2.4.3 (K2) Описать на примерах, как стратегия тестирования влияет на активности тестирования.

TM-2.4.4 (K3) Определить нормативы и шаблоны документации рабочих продуктов тестирования, которые будут соответствовать организации, жизненному циклу и потребностям проекта, адаптируя, где это применимо, имеющиеся шаблоны из органов стандартизации.

2.5 Оценка трудозатрат на тестирование

TM-2.5.1 (K3) Для приведенного проекта создать оценку всех видов деятельности процесса тестирования с использованием всех применимых методов оценки.

TM-2.5.2 (K2) Понять и привести примеры обстоятельств, которые могут повлиять на оценку затрат на тестирование.

2.6 Определение и использование метрик тестирования

TM-2.6.1 (K2) Описать и сравнить типичные метрики, связанные с тестированием.

TM-2.6.2 (K2) Сравнить различные аспекты мониторинга хода тестирования.

TM-2.6.3 (K4) Проанализировать и составить отчет о результатах тестирования с точки зрения остаточного риска, статуса дефектов, статуса тестового покрытия и доверия с целью обеспечения представления и рекомендаций, которые позволяют заинтересованным лицам проекта принимать решение о выпуске.

2.7 Ценность тестирования для бизнеса

TM-2.7.1 (K2) Привести примеры для каждой из четырех категорий, определяющих стоимость качества.

TM-2.7.2 (K3) Оценить стоимость тестирования, основываясь на стоимости качества совместно с другими количественными и качественными факторами, и сообщить расчетную стоимость заинтересованным лицам тестирования.

2.8 Распределенное, внешнее стороннее и локальное стороннее тестирование

TM-2.8.1 (K2) Понять факторы командообразования, необходимые для успешного использования распределенной, внешней сторонней и локальной сторонней командами тестирования.

2.9 Управление применением отраслевых стандартов

TM-2.9.1 (K2) Резюмировать источники и использование стандартов для тестирования программного обеспечения.

2.1 Введение

На Продвинутом уровне для профессионалов по тестированию появляется карьерная специализация. Эта глава посвящена области знаний, необходимых профессионалам по тестированию, стремящимся занять позиции ведущего специалиста по тестированию, руководителя тестирования, и директора по тестированию. В этой программе мы называем таких профессионалов общим термином - руководитель тестирования, понимая, что в различных организациях используются разные определения названий и уровней ответственности людей на таких позициях.

2.2 Управление тестированием в контексте

Главной обязанностью менеджера является обеспечение и использование ресурсов (люди, программное обеспечение, оборудование, инфраструктура и т.д.) для осуществления наиболее эффективного процесса. Для менеджеров программного обеспечения и ИТ часто процессы являются частью проекта или программы, предназначенных для поставки программного обеспечения или системы для внутреннего или внешнего использования. Для руководителя тестирования эти процессы связаны с тестированием, в частности основные активности процесса тестирования описаны в Главе 1 программы Базового уровня. Поскольку процесс тестирования дополнительно вносит свой вклад в общий успех проекта или программы (путем предотвращения более серьезных типов отказа), руководитель тестирования соответственно должен планировать и контролировать процессы тестирования. Другими словами, руководитель тестирования должен надлежащим образом организовать процессы тестирования, в том числе связанные активности и рабочие продукты, в соответствии с потребностями и обстоятельствами других заинтересованных лиц, их деятельность (например, жизненный цикл разработки программного обеспечения, в котором тестирование происходит), а также их рабочие продукты (например, спецификации требований).

2.2.1 Понятие заинтересованных лиц тестирования

Заинтересованные лица тестирования — это люди, которые имеют интерес к активностям тестирования, рабочим продуктам тестирования или качеству конечной системы или поставки. Заинтересованное лицо может быть явно или неявно вовлечено в активности тестирования, прямое или косвенное получение рабочих продуктов тестирования, прямой или косвенный эффект поставки, полученной в рамках проекта или программы.

Поскольку заинтересованные лица тестирования изменяются в зависимости от проекта, продукта, организации и других особенностей, они могут включать в себя следующие роли:

- Разработчики, ведущие специалисты разработки и менеджеры разработки. Эти заинтересованные лица обеспечивают реализацию тестируемого программного обеспечения, получают результаты тестирования и часто должны принимать меры на основе этих результатов (например, исправление найденных дефектов).
- Архитекторы баз данных, системные архитекторы и дизайнеры. Эти заинтересованные лица проектируют программное обеспечение, получают результаты тестирования, и часто должны принимать меры на основе этих результатов.
- Маркетинг и бизнес-аналитики. Эти заинтересованные лица определяют функции и уровень качества, присущий реализуемой функциональности. Они также часто участвуют в определении необходимого тестового покрытия, анализе результатов тестирования и принятии решений, основанных на результатах тестирования.
- Руководители высшего звена, менеджеры продукта и спонсоры проекта. Эти заинтересованные лица часто участвуют в определении необходимого тестового покрытия, анализе результатов тестирования и принятии решений, основанных на результатах тестирования.
- Менеджеры проекта. Эти заинтересованные лица несут ответственность за управление своими проектами, способствуя успеху, требующему баланса между показателями качества, сроками, функциями и бюджетом. Они часто закупают ресурсы, необходимые для активностей тестирования, и сотрудничают с руководителем тестирования при планировании и управлении.
- Сотрудники технической поддержки, поддержки клиентов, и сотрудники горячей линии. Эти заинтересованные лица поддерживают пользователей и клиентов, которые получают пользу от функций и качества поставляемого программного обеспечения.

- Прямые и косвенные пользователи. Эти заинтересованные лица напрямую используют программное обеспечение (то есть, они являются конечными пользователями), или получают продукцию или услуги, производимые или поддерживаемые программным обеспечением.

Дополнительные сведения о заинтересованных лицах можно получить в Главе 2 работы [Goucher09].

Этот список заинтересованных лиц не является исчерпывающим. Руководители тестирования должны определить для их проекта или программы конкретных заинтересованных лиц тестирования. Руководитель тестирования также должен понять точный характер отношений заинтересованных лиц к тестированию и как команда тестирования удовлетворяет потребности заинтересованных лиц. В дополнение к определению заинтересованных лиц тестирования, как описано выше, руководитель тестирования должен определить другие активности и рабочие продукты жизненного цикла разработки программного обеспечения, которые влияют на тестирование и/или зависят от тестирования. Без этого, процесс тестирования может не достичь оптимальной эффективности и результативности (см. Раздел 2.2.3).

2.2.2 Дополнительные активности жизненного цикла разработки программного обеспечения и рабочие продукты

Поскольку тестирование программного обеспечения — это оценка качества одного или нескольких рабочих продуктов, произведенных за пределами активностей тестирования, оно обычно существует в контексте более широкого набора активностей жизненного цикла разработки программного обеспечения. Руководитель тестирования должен планировать и направлять активности по тестированию с пониманием того, как эти другие виды активностей и их рабочие продукты влияют на тестирование, как обсуждалось в программе Базового уровня, и как тестирование влияет на эти другие виды активности и их рабочие продукты.

Например, в организациях, использующих методы гибкой (Agile) разработки, разработчики часто выполняют разработку через тестирование, создают автоматизированные модульные тесты и непрерывно интегрируют код (вместе с тестами для этого кода) в систему управления конфигурацией. Руководитель тестирования должен работать с менеджером разработки, чтобы иметь возможность убедиться, что тестировщики интегрировались и идут в одном направлении с этими активностями. Тестировщики могут рассмотреть модульные тесты, чтобы внести предложения по увеличению покрытия и эффективности этих тестов и получить более глубокое понимание программного обеспечения и его реализации. Тестировщики могут оценить способы интеграции своих собственных автоматизированных тестов, особенно функциональных регрессионных тестов, в систему управления конфигурацией. [Crispin09]

Несмотря на то, что определенные взаимоотношения между активностями тестирования, другими заинтересованными лицами тестирования, рабочими активностями жизненного цикла разработки программного обеспечения и рабочих продуктов изменяются в зависимости от проекта, выбранного жизненного цикла разработки программного обеспечения и ряда других особенностей, тестирование тесно взаимосвязано со следующими активностями:

- Разработка и управление требованиями. Руководитель тестирования должен рассмотреть требования во время определения объема и оценки трудозатрат тестирования, а также осознавать изменения требований и осуществлять управляющие воздействия на тестирование для соответствия этим изменениям. Технические тест-аналитики и тест-аналитики должны участвовать в рецензировании требований.
- Управление проектом. Руководитель тестирования, работая с тест-аналитиками и техническими тест-аналитиками, должен предоставить график и потребность в ресурсах

руководителю проекта. Руководитель тестирования должен работать с руководителем проекта для понимания изменений в плане проекта и осуществления управляющих воздействий на тестирование для соответствия этим изменениям.

- Управление конфигурацией, управление выпусками и управление изменениями. Руководитель тестирования, работая с командой тестирования, должен установить процессы и механизмы поставки объектов тестирования, а также отразить их в плане тестирования. Руководитель тестирования может попросить тест-аналитиков и технических тест-аналитиков создать тесты проверки сборки и обеспечить контроль версий во время выполнения тестирования.
- Разработка программного обеспечения и сопровождение. Руководитель тестирования должен работать с менеджерами разработки для координации поставки объектов тестирования, включая содержимое и дату каждого тестового выпуска, а также участвовать в управлении дефектами (см. Главу 4).
- Техническая поддержка. Руководитель тестирования должен работать с менеджерами технической поддержки, чтобы обеспечить надлежащую доставку результатов тестирования при завершении тестирования, чтобы те, кто участвует в поддержке продукта после выпуска, были осведомлены о известных отказах и временных решениях. Кроме того, руководитель тестирования должен работать с менеджером технической поддержки для анализа сбоев во время промышленной эксплуатации продукта с целью внедрения улучшений процесса тестирования.
- Производство технической документации. Руководитель тестирования должен работать с менеджером технической документации для обеспечения своевременной поставки документации для тестирования, а также управлением дефектов, обнаруженных в этих документах.

В дополнение к идентификации заинтересованных лиц тестирования, как описано выше, руководитель тестирования должен определить другие активности и рабочие продукты жизненного цикла разработки программного обеспечения, которые влияют на тестирование и/или зависят от тестирования. Если это не сделано, процесс тестирования не приведет к достижению оптимальной эффективности и результативности.

2.2.3 Согласование активностей тестирования и других активностей жизненного цикла

Тестирование должно быть составной частью проекта, независимо от используемых моделей разработки программного обеспечения. Это могут быть:

- Последовательные модели, такие как модель водопада, V-модель и W-модель. В последовательной модели все рабочие продукты и активности для конкретной фазы (например, требования, проектирование, реализация, модульное тестирование, интеграционное тестирование, системное тестирование, и приемочное тестирование) будут завершены до начала следующей фазы. Планирование тестирования, тест анализ, проектирование тестов и реализация тестовых процедур пересекается с планированием проекта, анализом бизнес- и системных требований, проектированием программного обеспечения и баз данных и программированием, с перекрытием в зависимости от рассматриваемого уровня тестирования. Выполнение тестирования протекает последовательно в соответствии с уровнями тестирования, обсуждаемыми в программе Базового уровня и в настоящей программе.
- Итеративные или инкрементальные модели, такие как Rapid Application Development (RAD) и Rational Unified Process (RUP). В итеративной или инкрементальной модели группируются функции, которые должны быть реализованы (например, в соответствии с приоритетами бизнеса или рисками) на различных фазах проекта, в том числе их рабочие продукты и активности, а затем это происходит для каждой группы функций. Фазы могут быть последовательными или пересекаться, и итерации сами по себе могут быть

последовательными или пересекаться. Инициация проекта, верхнеуровневое планирование тестирования и тест анализ происходит параллельно с планированием проекта и анализом бизнес- и системных требований. Детальное планирование тестирования, тест анализ, проектирование тестов, и выполнение тестирования происходит в начале каждой итерации перекрывающимся образом. Выполнение тестирования часто включает в себя перекрывающиеся уровни тестирования. Каждый уровень тестирования начинается как можно раньше и может продолжаться после того, как стартовали последующие, более высокие уровни тестирования.

- Гибкие (Agile) разработки, такие как SCRUM и Extreme Programming (XP). Это итеративный жизненный цикл, где итерации очень короткие (часто от двух до четырех недель). Рабочие продукты и активности для каждой итерации завершаются до начала следующей итерации (т.е. итерации являются последовательными). Тестирование проводится аналогично итерационным моделям, но с более высокой степенью перекрытия различных активностей с активностями разработки, в том числе значительное перекрытие выполнения тестирования (на различных уровнях) с активностями разработки. Все активности в итерации, в том числе активности тестирования, должны быть завершены до начала следующей итерации. В проекте с гибкой методологией роль руководителя тестирования обычно изменяется от прямой управленческой роли до технической роли полномочий/консультации.
- Спиральные. В спиральной модели на ранних стадиях проекта используются прототипы, чтобы подтвердить целесообразность и поэкспериментировать с дизайном и реализацией решений, используя уровень бизнес-приоритета и технического риска для выбора порядка, в котором проводятся прототипирование. Эти прототипы проходят тестирование, чтобы определить, какие аспекты технических проблем остаются нерешенными. После того, как основные технические проблемы будут решены, проект переходит в соответствии с любой последовательной или итеративной моделью.

Чтобы должным образом настроить активности тестирования в рамках жизненного цикла, руководитель тестирования должен иметь детальное понимание моделей жизненного цикла, используемых в организации. Например, в V-модели фундаментальный процесс тестирования ISTQB применительно к уровню системного тестирования может настраиваться следующим образом:

- Активности по планированию системного тестирования происходит одновременно с планированием проекта, и контроль тестирования продолжается до полного выполнения системного тестирования и завершения тестирования.
- Системный тест анализ и проектирование тестов происходят одновременно с описанием требований, с проектированием системной и архитектурной (высокоуровневой) спецификаций и проектированием спецификации компонентов (низкоуровневой).
- Активности по реализации системных тестов могут начаться во время системного проектирования, хотя большая часть этих мероприятий, как правило, происходит одновременно с кодированием и компонентным тестированием, работа по реализации системных тестов часто растягивается и заканчивается за несколько дней до начала выполнения системного тестирования.
- Активности по выполнению системного тестирования начинаются, когда все критерии начала тестирования достигнуты (или временно откладываются), что обычно означает, что, по крайней мере, компонентное тестирование и часто также тестирование интеграции компонентов закончены. Выполнение системного тестирования продолжается до тех пор, пока критерии завершения системного тестирования не будут достигнуты.

- Оценка критериев выхода системного тестирования и отчетов о результатах системного тестирования происходит в процессе выполнения системного тестирования, как правило, с большей частотой и срочностью по мере того как подходят сроки проекта.
- Активности по завершению системного тестирования происходят после того, как критерии выхода системного тестирования достигнуты, и выполнение системного тестирования признано завершенным, хотя они иногда могут быть отложены до завершения приемочного тестирования, и все активности по проекту закончены.

В итеративной или инкрементальной модели жизненного цикла должны быть выполнены одни и те же задачи, но сроки и масштабы могут различаться. Например, вместо того, чтобы реализовать все тестовое окружение в начале проекта, может быть более эффективным реализовать только ту часть, которая необходима для текущей итерации. При использовании любой из итерационных или инкрементальных моделей жизненного цикла объем расширения фундаментального процесса тестирования определяется глубиной планирования.

Дополнительно к этапам планирования, которые происходят для каждого проекта, выполнение тестов и отчетность также могут зависеть от жизненного цикла, используемого командой. Например, в рамках итерационного жизненного цикла может быть эффективным представить полные отчеты и проводить сеансы пост-итерационных рецензирования до начала следующей итерации. При обработке каждой итерации как мини-проекта команде дается возможность исправить и скорректировать планы на основе опыта предыдущей итерации. Поскольку итерация может быть короткой, а время ограниченным, может быть целесообразным сократить время и усилия, затрачиваемые на создание этой отчетности и оценок, но задачи должны быть выполнены для того, чтобы отследить общий прогресс тестирования и выявить любые проблемные области как можно быстрее. Проблемы процесса, с которыми сталкивались в одной итерации, могут легко повлиять и даже повториться в следующей итерации, если корректирующие меры не будут приняты.

Общая информация о том, как согласовать тестирование с другими видами активностей в течение жизненного цикла, может быть зафиксирована в стратегии тестирования (см. Раздел 2.4.2). Руководитель тестирования должен выполнять согласование конкретного проекта для каждого уровня тестирования и для любой выбранной комбинации жизненного цикла разработки программного обеспечения и процесса тестирования, во время планирования тестирования и / или планирования проекта.

В зависимости от потребностей организации, проекта и продукта, могут потребоваться дополнительные уровни тестирования помимо тех, которые определены в программе Базового уровня, например:

- Интеграционное тестирование программно-аппаратного комплекса
- Интеграционное системное тестирование
- Функциональное интеграционное тестирование
- Интеграционное тестирование продуктов клиента

Каждый уровень тестирования должны иметь следующие четко определенные элементы:

- Достижимые цели тестирования.
- Объем тестирования и элементы тестирования.
- Базис тестирования, вместе со средствами измерения покрытия на основе базиса (т.е. трассируемость).
- Критерии входа и выхода.
- Тестовые поставки, включая отчетность по результатам тестирования.

- Применяемые методы тестирования, вместе со способами обеспечения соответствующей степени покрытия с использованием этих методов.
- Измерения и метрики, имеющие отношение к целям тестирования, критериям входа и выхода и отчетности по результатам тестирования (включая измерения покрытия).
- Инструменты тестирования, которые будут применяться для выполнения конкретных задач тестирования (там, где это применимо).
- Ресурсы (например, тестовое окружение).
- Ответственные лица и группы, внутри и за ее пределами команды тестирования.
- Соблюдение организационных, нормативных или иных стандартов (там, где это применимо).

Как уже обсуждалось недавно в этом разделе, лучшая практика должна определить эти элементы согласованно через все уровни тестирования, чтобы избежать расточительных и опасных провалов на различных уровнях похожих тестов.

2.2.4 Управление нефункциональным тестированием

Недостаточность планирования нефункциональных тестов может привести к обнаружению серьезных, иногда катастрофических, проблем качества в системе после выпуска. Однако, многие виды нефункциональных тестов являются дорогостоящими, поэтому руководитель тестирования должен выбрать на основе рисков и ограничений, какие нефункциональные тесты выполнять. Кроме того, существует много различных типов нефункциональных тестов, некоторые из них могут не подходить для применения в конкретном случае.

Поскольку руководитель тестирования может не иметь достаточного опыта, чтобы учесть все особенности планирования, он должен делегировать часть обязанностей по планированию тестирования техническим тест-аналитикам (а в некоторых случаях тест-аналитикам), назначая им активности по нефункциональному тестированию. Руководители тестирования должны попросить аналитиков рассмотреть следующие общие факторы:

- требования заинтересованных лиц;
- необходимые инструменты;
- тестовое окружение;
- организационные факторы;
- безопасность.

Более подробную информацию можно найти в Главе 4 программы Продвинутого уровня для технического тест-аналитика. [ISTQB ATA SIL].

Другое важное соображение для руководителей тестирования состоит в том, как интегрировать нефункциональные тесты в жизненный цикл разработки программного обеспечения. Распространенной ошибкой является ожидание старта нефункциональных тестов, пока все функциональные тесты не будут завершены, которые могут привести к обнаружению критических нефункциональных дефектов в конце. Вместо этого, нефункциональные тесты должны быть приоритетными и упорядочены в соответствии с рисками. Часто существуют способы смягчить нефункциональные риски на ранних стадиях тестирования или даже во время разработки. Например, анализ практичности прототипов пользовательского интерфейса во время системного проектирования может быть весьма эффективным при определении дефектов практичности, которые создали бы значительные проблемы со сроками, если бы были обнаружены к концу системного тестирования.

В итерационных моделях жизненного цикла, из-за темпов изменений и итераций может стать трудным сфокусироваться на некоторых нефункциональных тестах, которые требуют создания сложных тестовых сред. Активности проектирования тестов и реализация, которые занимают

больше времени, чем время одной итерации, должны быть организованы в виде отдельных рабочих активностей вне итераций.

2.2.5 Управление тестированием, основанным на опыте

В то время как тестирование, основанное на опыте, предоставляет преимущества, эффективно находя дефекты, которые другие методы тестирования могут пропустить, и служит в качестве проверки полноты этих методов, могут возникать проблемы управления тестированием. Руководитель тестирования должен знать о проблемах, а также о преимуществах методов, основанных на опыте, в частности об исследовательском тестировании. Трудно определить покрытие, достигнутое в ходе такого тестирования, поскольку в его рамках выполняется минимальная подготовка тестов. Воспроизводимость результатов тестирования требует особого внимания со стороны руководства, особенно, когда участвуют несколько тестировщиков.

Один из способов управления тестированием, основанным на опыте, особенно исследовательским тестированием, состоит в том, чтобы разбить работу по тестированию на небольшие периоды времени, от 30 до 120 минут, иногда называемые тестовыми сессиями. Ограничение времени и нацеленность на работу в сессии обеспечивает уровень контроля и планирования. Каждая сессия предусматривает порядок тестирования, о котором руководитель тестирования сообщает тестировщику в письменной форме или устно. Порядок тестирования предоставляет тестовые условия, которые должны быть охвачены в сессии тестирования, что дополнительно помогает удерживать в центре внимания и предотвратить перекрытие, если несколько человек одновременно проводят исследовательское тестирование.

Другой метод управления тестированием на основе накопленного опыта - интеграция такого самостоятельного и спонтанного тестирования в более традиционные заранее спроектированные сессии тестирования. Например, тестировщикам могут разрешить (и выделить время) исследовать за пределами явных шагов, входов и ожидаемых результатов в рамках заранее определенных тестов. Тестировщикам также могут быть назначены такие самостоятельные сеансы тестирования, как часть их ежедневного тестирования до, во время или после запуска заранее определенных тестов. Если такие сеансы тестирования выявят дефекты или интересные области для дальнейшего тестирования, заранее определенные тесты могут быть пополнены.

В начале исследовательской сессии тестировщик проверяет и выполняет необходимую настройку задач для тестов. Во время сеанса тестировщик узнает о тестируемом приложении, разрабатывает и выполняет тесты в соответствии с методикой, применяя то, что узнал о приложении, расследует какие-либо дефекты, и фиксирует результаты проверки в журнале. (Если требуется повторяемость тестов, тестировщики также должны регистрировать входные данные тестов, действия и события.) После сессии может произойти анализ, который задает направление последующих сессий.

2.3 Тестирование, основанное на рисках, и другие подходы приоритизации тестов и распределения трудозатрат

Универсальной задачей управления тестированием является правильный выбор, распределение и приоритизация тестов. То есть, из практически бесконечного числа тестовых условий и сочетаний условий, которые могут быть покрыты, команда тестирования должна выбрать конечный набор условий, определить соответствующие трудозатраты, чтобы распределить покрытие каждого условия тестовым сценарием, и упорядочить полученные тестовые сценарии в приоритетном порядке, который оптимизирует эффективность и результативность планируемой работы тестирования. Наряду с другими факторами руководителем тестирования, могут быть использованы выявление и анализ риска, чтобы помочь решить эту проблему, хотя многие взаимодействующие ограничения и переменные могут потребовать компромиссного решения.

2.3.1 Тестирование, основанное на рисках

Риск — это возможность негативного или нежелательного результата или события. Риски существуют всякий раз, когда может произойти некоторая проблема, которая уменьшила бы восприятие качества продукта или успеха проекта клиентом, пользователем, участником или заинтересованной стороной. Там, где основной эффект потенциальной проблемы влияет на качество продукта, потенциальные проблемы называются рисками качества, рисками продукта или рисками качества продукта. Там, где основной эффект потенциальной проблемы влияет на успешное завершение проекта, потенциальные проблемы называются рисками проекта или рисками планирования.

В тестировании, основанном на рисках, риски качества выявляют и оценивают в ходе анализа с заинтересованными лицами рисков качества продукции. Затем команда тестирования разрабатывает, внедряет и выполняет тесты для снижения рисков качества. Качество включает в себя совокупность функций, поведения, характеристик и атрибутов, которые влияют на удовлетворение клиента, пользователя и всех заинтересованных лиц. Таким образом, риск качества - это потенциальная ситуация, когда проблемы качества могут существовать в продукте. Примеры рисков качества для системы включают в себя: неверные вычисления в отчетах (функциональный риск, связанный с точностью), медленную реакцию на пользовательский ввод (нефункциональный риск, связанный с эффективностью и временем отклика), а также трудности в понимании экранов и полей (нефункциональный риск, связанный с практичностью и понятностью). Когда тесты обнаруживают дефекты, тестирование смягчает риск качества, обеспечив осведомленность о дефектах и возможности для борьбы с ними перед выпуском. Если тесты не находят дефекты, тестирование смягчает риск качества путем гарантии, чтобы при выполнении тестовых условий система работает правильно.

Тестирование, основанное на рисках, использует риски качества продукта для выбора тестовых условий, для распределения трудозатрат на тестирование этих условий, а также для определения приоритетов полученных тестовых сценариев. Существует различные методы тестирования, основанного на рисках, с существенными изменениями в типе и уровне собранной документации, и уровня применения формальности. Явно или неявно тестирование, основанное на рисках, имеет цель с помощью тестирования уменьшить общий уровень риска качества, и, в частности, снижение этого уровня риска до приемлемого.

Тестирование, основанное на рисках, состоит из четырех основных видов деятельности:

- Определение рисков
- Оценка рисков
- Смягчение рисков
- Управление рисками

Эти виды деятельности пересекаются. В следующих подразделах будет обсуждаться каждый из этих видов деятельности.

Для наибольшей эффективности, определение и оценка рисков должна включать представителей всех групп заинтересованных лиц продукта и проекта, хотя иногда в проектной реальности в результате некоторые заинтересованные лица выступают в качестве заместителей для других заинтересованных лиц. Например, в разработке свободно распространяемого программного обеспечения небольшая выборка потенциальных клиентов может быть приглашена помочь выявить потенциальные дефекты, которые могут повлиять на использование программного обеспечения в наибольшей степени; в этом случае выборка потенциальных клиентов служит в качестве замены всей возможной клиентской базы. Из-за их особого опыта с рисками качества продукта и отказов, тестировщики должны активно участвовать в процессе определения и оценки риска.

2.3.1.1 Определение рисков

Заинтересованные лица могут определить риски с помощью одного или нескольких из следующих методов:

- экспертные интервью;
- независимые оценки;
- использование шаблона рисков;
- ретроспектива проекта;
- группы специалистов для обсуждения рисков;
- мозговой штурм;
- чек-листы;
- использование прошлого опыта.

Привлечение более широкой выборки заинтересованных лиц в процесс определения рисков, скорее всего, определит большинство значимых рисков качества продукции.

Определение риска часто приводит к побочным продуктам, т.е. определению проблем, которые не являются риском качества продукта. Примеры включают общие вопросы/ проблемы продукта или проекта, проблемы в ссылочных документах, таких как требования и спецификации дизайна. Риски проекта также часто определяются как побочный продукт определения рисков качества, но не являются основным направлением тестирования, основанного на рисках. Тем не менее, управление рисками проекта имеет важное значение для всего тестирования, а не только для тестирования, основанного на риске, и обсуждается далее в Разделе 2.4.

2.3.1.2 Оценка рисков

Как только определение риска произошло, может начаться оценка риска путем исследования выявленных рисков. В частности, оценка риска включает классификацию каждого риска и определения вероятности и воздействия, связанные с каждым риском. Оценка риска может также включать оценку или назначения других свойств каждого риска (например, владельца риска).

Классификация риска означает присваивание каждому риску соответствующего типа, такого как производительность, надежность, функциональность и так далее. Некоторые организации используют стандарт ISO 9126 [ISO9126] (который заменяется стандартом ISO 25000 [ISO25000]) характеристик качества для классификации, но многие организации используют другие схемы классификации. Тот же чек-лист, используемый для определения риска, часто используется для классификации рисков. Существуют общие чек-листы рисков качества, и многие организации настраивают эти чек-листы. При использовании чек-листов в качестве основы для определения рисков классификация риска часто возникает во время определения.

Определение уровня риска, как правило, включает в себя оценку каждого элемента риска, вероятность возникновения и влияние при наступлении. Вероятность возникновения - это вероятность того, что потенциальная проблема существует в тестируемой системе. Другими словами, вероятность — это оценка уровня технического риска. Факторы, влияющие на вероятность для рисков продукта и проекта, включают:

- сложность технологии и команд;
- личные проблемы и проблемы подготовки среди бизнес-аналитиков, дизайнеров и программистов;
- конфликт внутри команды;
- контрактные проблемы с поставщиками;
- географически распределенная команда;
- конфликт используемых и новых подходов;

- инструменты и технологии;
- слабое управленческое или техническое руководство;
- время, ресурсы, бюджет и давление менеджмента;
- отсутствие мероприятий по обеспечению качества;
- высокие темпы изменений;
- высокий темп выявления ранних дефектов;
- проблемы взаимодействия и интеграции.

Воздействие при возникновении - серьезность эффектов для пользователей, потребителей или других заинтересованных лиц. Факторы, влияющие на воздействие рисков проекта и продукта, включают:

- частота использования затронутой функции;
- критичность функции для достижения бизнес-целей;
- ущерб репутации;
- потеря бизнеса;
- потенциальные финансовые, экологические и социальные потери или помехи;
- гражданские или уголовные правовые санкции;
- потеря лицензии;
- отсутствие разумных обходных путей;
- видимость отказа, приводящего к отрицательной рекламе;
- безопасность.

Уровень риска может быть оценен как количественно, так и качественно. Если вероятность и влияние могут быть определены количественно, можно перемножить эти два значения, чтобы вычислить количественный приоритет риска. Тем не менее, как правило, уровень риска может быть установлен только качественно. То есть, можно говорить о вероятности очень высокой, высокой, средней, низкой или очень низкой, но не может быть выражена вероятность в процентах с любой реальной точностью. Аналогичным образом, можно говорить о влиянии - очень высокое, высокое, среднее, низкое или очень низкое, но нельзя выразить влияние в финансовых терминах полностью или точно. Эту качественную оценку уровня риска не следует рассматривать как ухудшение количественных подходов; на самом деле, когда количественные оценки уровней риска используются неуместно, результаты вводят в заблуждение заинтересованных лиц о том, насколько на самом деле риски понятны и управляемы. Качественные оценки уровней риска часто объединяются, путем умножения или сложения, чтобы создать совокупную оценку риска. Это совокупная оценка риска может быть выражена в качестве приоритета риска, который следует интерпретировать как относительный рейтинг.

Кроме того, если анализ риска не основывается на обширных и статистически достоверных данных о рисках, он будет основываться на субъективных представлениях о вероятности и воздействия заинтересованных лиц. Проектные менеджеры, программисты, пользователи, бизнес-аналитики, архитекторы и тестировщики, как правило, имеют различные представления и, таким образом, возможно, различные мнения относительно уровня риска для каждого элемента риска. Процесс анализа риска должен включать в себя способ достижения согласия или, в худшем случае, установление согласованного уровня риска (например, руководство диктует или берет среднее, медиану или моду для элемента риска). Кроме того, уровни риска должны быть проверены на хорошем распределении на всем диапазоне, чтобы убедиться, что показатели риска обеспечивают целенаправленные указания в отношении последовательности тестирования, определения приоритетов и распределения трудозатрат. В противном случае, уровни риска не могут быть использованы в качестве руководства для осуществления деятельности по снижению рисков.

2.3.1.3 Смягчение рисков

Тестирование, основанное на риске, начинается с анализа рисков качества (выявления и оценки рисков качества продукта). Этот анализ является основой главного плана и других планов тестирования. Как указано в плане(ах), тесты разрабатываются, реализовываются и выполняются для того, чтобы покрыть риски. Трудозатраты, связанные с разработкой и выполнением тестирования, пропорциональны уровню риска, что означает, что более тщательные методы тестирования (такие, как парное тестирование) используются для более высоких рисков, в то время как менее тщательные методы тестирования (например, разбиение эквивалентности или ограниченное по времени исследовательское тестирование) используются для более низких рисков. Кроме того, приоритет разработки и выполнения тестирования основан на уровне риска. Некоторые стандарты, связанные с безопасностью (например, FAA DO-178B / ED 12B, IEC 61508), предписывают методы тестирования и степень покрытия на основе уровня риска. Кроме того, уровень риска должен влиять на принятие решений, таких как использование рецензирования проектных работ (включая тесты), уровень независимости, уровень опыта тестировщика, степень подтверждающего тестирования (повторного тестирования) и выполнение регрессионного тестирования.

В ходе проекта, группа тестирования должна постоянно помнить о дополнительной информации, которая меняет набор рисков качества и / или уровень риска, связанный с известными рисками качества. Периодическая корректировка анализа риска качества приводит к корректировке тестов, которые должны проводиться, по крайней мере, на основных этапах проекта. Корректировки включают определение новых рисков, повторную оценку уровня существующих рисков, а также оценки эффективности деятельности по снижению рисков. Рассмотрим один пример - если сеанс определения и оценки риска произошел на основе спецификации требований на этапе требований, то, как только спецификация проектирования будет завершена, должна пройти повторная оценка рисков. Рассмотрим другой пример - если во время тестирования компонента находятся значительно больше дефектов, чем предполагалось, можно сделать вывод, что вероятность возникновения дефектов в этой области была выше, чем предполагалось, и, таким образом, следует увеличить вероятность и общий уровень риска. Это может привести к увеличению тестирования для этого компонента.

Риски качества продукта также могут быть смягчены до начала выполнения тестов. Например, если проблемы с требованиями находятся в процессе определений рисков, проектная группа может осуществить тщательное рецензирование спецификации требований в качестве смягчающего действия. Это может снизить уровень риска, который может означать, что необходимо меньше тестов для снижения оставшихся рисков качества.

2.3.1.3 Управление рисками в жизненном цикле

В идеале, управление рисками происходит на протяжении всего жизненного цикла. Если организация имеет документированную политику и/или стратегию тестирования, то в них должен быть описан общий процесс управления рисками проекта и продукта в тестировании, и показано, как управление рисками интегрировано и влияет все этапы тестирования.

В зрелой организации, где команда проекта осознает риск, управление рисками происходит на многих уровнях, а не только для тестирования. Важные риски не только рассматривались ранее в конкретных уровнях тестирования, но на более ранних уровнях тестирования. (Например, если производительность определяется как ключевая область риска качества, то тестирование производительности начинается не только на ранних этапах системного тестирования - тесты производительности будут запускаться во время модульного и интеграционного тестирования.) Зрелые организации выявляют не только риски, но и источники риска и последствия, которые должны произойти в результате тех рисков. Для тех дефектов, которые встречаются, используется анализ причин, чтобы более глубоко понять источники риска и реализовать усовершенствования

процесса, которые предотвращают дефекты в первую очередь. Смягчение существует на протяжении всего жизненного цикла разработки программного обеспечения. Анализ рисков — это полностью информированный, принимающий во внимание соответствующую трудовую деятельность, анализ поведения системы, оценка рисков на основе затрат, анализа рисков продукта, анализа рисков конечного пользователя и анализа рисков ответственности. Анализ рисков выходит за рамки тестирования, с участием команды тестирования и влиянием на анализ рисков в рамках всей программы.

Большинство методов тестирования, основанного на рисках, также включают в себя методы использования уровня риска для последовательности и приоритизации тестирования, тем самым обеспечивая раннее покрытие наиболее важных областей и открытие наиболее важных дефектов во время выполнения тестирования. В одних случаях все тесты наивысшего риска выполняются перед любыми тестами более низких рисков, и тесты выполняются в строгом порядке риска (часто называемым «в глубину»); в других случаях используется выборочный метод для отбора выборки тестов по всем выявленным рискам объектов на основе веса риска, в то же время обеспечивая покрытие каждого риска объекта, по крайней мере, один раз (часто называемый "в ширину").

Независимо от того происходит тестирование, основанное на рисках, в глубину или в ширину, вполне возможно, что время, отведенное для тестирования, истечет до того, как будут запущены все тесты. Тестирование, основанное на рисках, позволяет тестировщикам предоставить отчет руководству с точки зрения остаточного уровня риска в этой точке, и позволяет руководству решить, следует ли расширить тестирование или перевести оставшийся риск на пользователей, клиентов, справочную службу / техническую поддержку, а также / или обслуживающий персонал.

Во время выполнения тестирования самые сложные методы тестирования, основанного на рисках, не должны быть самыми формальными или тяжеловесными — это позволяет участникам проекта, менеджерам проекта и продукту, руководителям, топ-менеджерам и заинтересованным лицам проекта контролировать и управлять жизненным циклом разработки программного обеспечения, включая принятие решения о выпуске, на основе остаточного уровня риска. Для этого требуется предоставить руководителю тестирования отчет о тестировании с точки зрения риска, так, чтобы его могли понять все заинтересованные лица тестирования.

2.3.2 Методы тестирования, основанного на рисках

Есть много методов тестирования, основанного на рисках. Некоторые из этих методов являются весьма неформальными. Примеры включают в себя подходы, где тестировщик анализирует риски качества в ходе исследовательского тестирования [Whittaker09]. Это может помочь руководить тестированием, но может и привести к чрезмерному акценту на вероятность дефектов, а не их влияние, и не содержать кросс-функциональные входные данные заинтересованных лиц. Кроме того, такие подходы носят субъективный характер и зависят от навыков, опыта и предпочтений конкретного тестировщика. По существу, эти подходы редко достигают всех преимуществ тестирования, основанного на рисках.

В попытке получить преимущества тестирования, основанного на рисках, при минимизации затрат, много практиков используют легкие подходы, основанные на риске. Это позволяет сочетать оперативность и гибкость неформальных подходов с мощностью и согласованностью более формальных подходов. Примеры легких подходов включают в себя: Прагматический анализ и управление рисками (PRAM) [Black09], Систематическое тестирование программного обеспечения (SST) [Craig02], и Управление рисками продукта (PRisMa) [vanVeenendaal12]. Дополнительно к обычным атрибутам тестирования, основанного на рисках, эти методы, как правило, имеют следующие атрибуты:

- Развитый с течением времени отраслевой опыт работы вместе с тестированием, основанным на рисках, особенно в тех отраслях промышленности, где вопросы эффективности являются важными.
- Активное участие кросс-функциональной команды заинтересованных сторон, представляющих как бизнес, так и техническую точку зрения, в ходе первоначального определения и оценки рисков.
- Оптимизация введения на самых ранних этапах проекта, когда варианты смягчения риска качества максимальны и когда основные и побочные результаты работы анализа риска качества могут помочь повлиять на спецификацию и реализацию продукта таким образом, чтобы свести к минимуму риск.
- Использование сгенерированного вывода (матрицы риска или таблицы анализа риска) в качестве основы плана тестирования и тестовых условий и, следовательно, для всех последующих действий управления тестированием и анализом.
- Поддержка отчетов о результатах испытаний с точки зрения остаточного риска для всех уровней заинтересованных сторон тестирования.

Некоторые из этих методов (например, SST) требуют спецификаций требований в качестве входных данных для анализа риска и не могут быть использованы, за исключением случаев, когда спецификации требований предусмотрены. Другие методы (например, PRisMa и PRAM) поощряют использование смешанной стратегии, основанной на рисках и на требованиях, используя требования и / или другие спецификации в качестве входных данных для анализа риска, но могут функционировать полностью на основе входных данных заинтересованных лиц. Применение требований в качестве входных данных помогает обеспечить хорошее покрытие требований, но руководитель тестирования должен удостовериться, что важные риски, не предложенные требованиями, особенно в нефункциональных областях, не будут пропущены. Когда хорошие, приоритезированные требования предоставляются в качестве входных данных, то, как правило, видна сильная корреляция между уровнями риска и приоритетами требований.

Многие из этих методов также поощряют использование процесса и оценки рисков как способ достижения согласия между заинтересованными лицами относительно подхода тестирования. Это мощное преимущество, но оно требует от заинтересованных лиц уделить время для участия или в групповых сессиях "мозгового штурма" или в интервью один на один. Недостаточное участие заинтересованных лиц приводит к пробелам в анализе риска. Конечно, заинтересованные лица не всегда соглашались с уровнями риска, поэтому лицо, проводящее анализ рисков качества, должно работать творчески и позитивно с заинтересованными лицами для достижения наилучшей степени согласия. Все навыки подготовленного координатора совещаний ведущего рецензирования применимы к лицу, ведущему анализ рисков качества.

Подобно формальным методам, легкие методы позволяют использовать вес вероятности возникновения и факторов влияния, чтобы подчеркнуть бизнес или технические риски (например, в зависимости от уровня тестирования). В отличие от формальных методов, легкие методы:

- используют только два фактора - вероятность и влияние;
- используют простые, качественные оценки и масштаб.

Легкий характер этих подходов обеспечивает гибкость, применимость в ряде областей, и доступность между командами всех уровней опыта и квалификации (даже с не техническими людьми и новичками).

При формальном, тяжеловесном завершении определении веса, руководитель тестирования имеет альтернативы:

- Анализ опасных факторов, который расширяет процесс анализа, пытаюсь определить опасности, которые лежат в основе каждого риска.
- Стоимость воздействия, где процесс оценки риска включает в себя определение, для каждого элемента риска качества трех факторов:
 - 1) вероятности (выраженной в процентах) отказа, связанного с элементом риска;
 - 2) стоимости потери (выраженной в качестве финансовой величины), связанной с типичным отказом, связанным с элементом риска, который может случиться в промышленной эксплуатации;
 - 3) стоимости тестирования таких отказов.
- Анализ типов отказа и эффекта (FMEA) и его варианты [Stamatis03], где риски качества, их возможные причины, а также их возможный эффект определены и затем установлены оценки критичности, приоритета и выявления.
- Развертывание функции качества (QFD), которая представляет собой метод управления рисками качества с результатами тестирования, в частности, связанные с рисками качества, которые возникают от неправильного или недостаточного понимания требований клиентов или пользователей.
- Анализ дерева недочетов (FTA), где различные фактические наблюдаемые отказы (во время тестирования или промышленной эксплуатации), или потенциальные отказы (риски качества), подвергаются анализу первопричин, начиная с дефектов, которые могут вызвать собой, затем ошибок или дефектов, которые могут вызвать те дефекты, продолжая до тех пор, пока различные первопричины не определены.

Конкретные методы, которые должны быть использованы для тестирования, основанного на рисках, а также степень формальности каждого метода, в зависимости от проекта, процесса и соображений продукта. Например, неформальный подход, такой как, исследовательский метод Уиттакера, может применяться к патчу или быстрому исправлению. В гибких (Agile) проектах анализ риска качества полностью интегрирован в ранний период каждого спринта, и документирование рисков сочетается с отслеживанием пользовательских историй. Системы систем требуют анализа риска для каждой системы, а также в целом для системы систем. Для проектов с критически важными аспектами безопасности и предназначения требуется более высокие уровни формальности и документации.

Входные данные, процессы и выходные данные для тестирования, основанного на рисках, будут, как правило, определяется выбранным методом. Общие входные данные включают идеи заинтересованных лиц, спецификации и исторические данные. Общие процессы включают в себя определение, оценку и контроль рисков. Общие выходные данные включают в себя перечень рисков качества (с соответствующей степенью риска и рекомендованным распределением трудозатрат на тестирование), дефектов, обнаруженных в исходных документах, таких как спецификации, вопросы или проблемы, связанные с элементами риска, а также риски проекта, влияющие на тестирование или на проект в целом.

Как правило, наиболее важным фактором успеха тестирования, основанного на рисках, является вовлечение правильной группы заинтересованных лиц в определение и оценку рисков. У всех заинтересованных лиц есть свое собственное понимание того, что представляет собой качество продукта, а также свой собственный набор приоритетов и опасений по поводу качества. Однако, заинтересованные лица, как правило, делятся на две основные категории: бизнес-участники и техническая сторона.

Бизнес-участники включают, в частности, клиентов, пользователей, обслуживающий персонал, службу технической помощи и персонал службы технической поддержки. Эти заинтересованные лица понимают клиента и пользователя, и, таким образом, могут определить риски и оценить влияние с точки зрения бизнеса.

Техническая сторона включает, в частности, разработчиков, архитекторов, администраторов баз данных и сетевых администраторов. Эти заинтересованные лица понимают основные способы, при которых программное обеспечение может выйти из строя, и, таким образом, могут определить риски и оценить вероятность с технической точки зрения.

Некоторые заинтересованные лица имеют и точку зрения бизнеса, и техническую точку зрения. Например, эксперты предметной области, которые находятся в ролях тестировщиков или бизнес-аналитиков, часто имеют более обширное представление о рисках, соответствующее их техническому и бизнес опыту.

Процесс определения элементов риска порождает значительный список рисков. Нет необходимости заинтересованным лицам спорить об элементе риска, до тех пор, пока один из заинтересованных лиц воспринимает что-то как риск для качества системы — это и есть элемент риска. Тем не менее, важно, чтобы заинтересованные лица достигли согласия в отношении оценки уровня риска. Например, в облегченных подходах, которые используют вероятность и влияние как оценку факторов, часть процесса должна включать в себя нахождение общей схемы ранжирования вероятности и влияния. Все заинтересованные лица, в том числе группы тестирования, должны использовать эту шкалу и должны быть в состоянии прийти к единой оценке вероятности и воздействия для каждого элемента риска качества.

Если тестирование, основанное на рисках, будет использоваться в долгосрочной перспективе, то руководитель тестирования должен предложить тестирование, основанное на рисках, заинтересованным лицам. Кросс-функциональная группа должна видеть ценность анализа риска для обеспечения непрерывного использования метода. Это требует, чтобы руководитель тестирования понимал потребности заинтересованных лиц, ожидания и наличие времени для участия в этом процессе.

Хорошее взаимодействие заинтересованных лиц с процессом анализа рисков качества является важным преимуществом руководителя тестирования. Это преимущество заключается в том, что по указанным проектам со слабыми или недостающими требованиями, заинтересованные лица могут по-прежнему определять риски, когда руководствуются подходящими чек-листами. Это преимущество можно увидеть, когда после внедрения тестирования, основанного на рисках, эффективность обнаружения дефектов команды тестирования улучшается. Это происходит потому, что используется более полный базис тестирования - в данном случае список элементов риска качества.

Во время завершения тестирования, основанного на рисках, команда тестирования должны оценить степень, с которой они достигли поставленных целей. Во многих случаях это связано с ответами на все или некоторые из следующих четырех вопросов с помощью метрик и совещаний с командой:

- Обнаружила ли команда тестирования больший процент важных дефектов, чем менее важных дефектов?
- Нашла ли команда тестирования большинство важных дефектов в начале периода выполнения тестирования?
- Была ли команда тестирования в состоянии объяснить заинтересованным лицам результаты тестирования с точки зрения риска?
- Тесты, не выполненные командой тестирования (если таковые имеются), имеют меньший уровень риска, чем выполненные?

В большинстве случаев, успешные результаты тестирования, основанного на рисках, дают утвердительный ответ на все четыре вопроса. В долгосрочной перспективе руководитель тестирования должен ставить перед собой цели улучшения процесса для этих показателей, наряду

со стремлением повысить эффективность процесса анализа рисков качества. Конечно, и другие показатели, и критерии успеха могут быть применены к тестированию, основанному на рисках, а руководителю тестирования следует внимательно рассмотреть взаимосвязь между этими метриками, стратегическими целями группы тестирования, и поведением, которое будет результатом управления на основе конкретного набора показателей и критериев успеха.

2.3.3 Другие методы для выбора тестов

В то время как многие руководители тестирования используют тестирование, основанное на рисках, в качестве одного из элементов своей стратегии тестирования, многие из них также используют и другие методы.

Одним из наиболее известных альтернативных методов разработки и приоритизации тестовых условий является тестирование на основе требований. Тестирование на основе требований может использовать несколько методов, таких как рецензирование неоднозначности, анализ тестовых условий, и отображение причинно-следственных связей. Рецензирование неоднозначности определяет и устраняет неоднозначности в требованиях (которые служат в качестве базиса тестирования), часто с использованием чек-листа общих дефектов требований (см. [Wiegerts03]).

Как описано в [Craig02], анализ тестовых условий предполагает внимательное чтение спецификации требований для определения тестовых условий для покрытия. Если этим требованиям присвоен приоритет, это может быть использовано для распределения трудозатрат и приоритизации тестовых случаев. При отсутствии присвоенного приоритета трудно определить соответствующие трудозатраты и порядок тестирования без сочетаний тестирования на основе требований с тестированием, основанном на рисках.

Отображение причинно-следственных связей обсуждается в модуле Повышенного уровня для тест-аналитика (Advanced Test Analyst) в контексте покрытия комбинаций тестовых условий в рамках тестового анализа. Однако оно имеет более широкое применение в том, что может уменьшить чрезвычайно большую проблему при тестировании до контролируемого количества тестовых сценариев и тем не менее обеспечивая 100% покрытие функционала тестового базиса. Отображение причинно-следственных связей также выявляет пробелы в тестовом базисе во время проектирования тестовых сценариев, которое может выявить дефекты на ранней стадии жизненного цикла разработки программного обеспечения, когда проектирование тестов стартует на этапе проекта требований. Одним из главных препятствий для выбора отображения причинно-следственных связей является сложность создания этих графов. Есть инструменты, чтобы поддержать этот метод, который сделать вручную сложно.

Во время тестового анализа и планирования, команда тестирования определяет профили использования и пытается покрыть их тестами. Профиль использования представляет собой оценку, на основе доступной информации реалистичного использования программного обеспечения. Это означает, что, как и с тестированием, основанным на рисках, профили использования могут не полностью смоделировать возможную действительность. Однако, если информации достаточно и входные данные заинтересованных лиц доступны, то модель будет адекватной (см. [Musa04]).

Некоторые руководители тестирования применяют также методические подходы, такие как чек-листы, чтобы определить, что проверить, в каком количестве и в каком порядке. Для продуктов, которые стабильны, может быть достаточно наличие чек-листов с основными функциональными и нефункциональными областями тестирования, в сочетании с репозиторием существующих тестовых сценариев. Чек-лист содержит эвристики для распределения затрат и последовательности тестирования, как правило, в зависимости от типа и количества изменений,

которые произошли. Такие подходы, становятся менее действенными, когда используется тестирование для более незначительных изменений.

Общим препятствием для тестирования на основе требований является то, что часто спецификации требования неоднозначны, не тестопригодны, неполные или вообще отсутствуют. Не все организации мотивированы, чтобы решить эти проблемы, поэтому тестировщикам, которые сталкиваются с такими ситуациями, необходимо выбрать другую стратегию тестирования.

Другим методом, который иногда применяется, чтобы увеличить использование существующих требований, является создание пользовательских или функциональных профилей, подход на основе модели, который использует сочетание сценариев использования, пользователей (иногда называемых действующими лицами), входные и выходные данные, так чтобы точно описать реальное использование системы. Это позволяет тестировать не только функциональность, но и практичность, возможность взаимодействия, надежность, безопасность и производительность.

И, наконец, еще один общий метод состоит в использовании реактивного подхода. В реактивном подходе очень мало тестового анализа, проектирование или реализация задач возникают перед выполнением тестирования. Команда тестирования ориентирована на реакцию продукта, который реально предоставлен. Обнаруженные кластеры ошибок становятся центром дальнейшего тестирования. Приоритизация и распределение полностью динамическая. Реактивный подход может работать в качестве дополнения к другим подходам, но при использовании исключительно реактивного подхода, как правило, пропускаются основные области применения, которые являются важными, но не страдают от большого количества ошибок.

2.3.4 Приоритизация тестов и распределение усилий в процессе тестирования

Какой бы метод или сочетание методов не использовались, руководитель тестирования должен включить этот метод в процессы проекта и тестирования. Например, в последовательных жизненных циклах (например, V-модели), команда тестирования выбирает тесты, распределяет трудозатраты на тестирование и первоначально устанавливает приоритеты тестам во время фазы требований, с периодическими корректировками. В то время как в итеративной или гибкой методологии жизненные циклы требуют подхода от итерации к итерации, а планирование и контроль тестирования должны учитывать, в какой степени риски, требования и / или профили использования будут развиваться и соответствующим образом реагировать.

Во время тестового анализа, проектирования и реализации должны быть применены распределение и приоритизация, определяемые при планировании тестирования. Это общая разбивка в процессе тестирования для тщательного анализа и/или моделирования встречается, только для той информации, которая не будет использоваться для развития процесса тестирования. Такая разбивка обычно происходит в процессе проектирования и реализации.

При выполнении тестирования должна быть выполнена приоритизация, установленная при планировании тестирования, несмотря на то что важно обновлять приоритизацию периодически на основе информации, полученной после того, как план был изначально написан. При оценке и отчетности результатов тестирования и статуса критериев выхода, руководитель тестирования должен также оценивать и сообщать с точки зрения рисков, требований, профилей использования, чек-листов, а также других руководств, факторы, используемые для выбора и определения приоритетности тестов. При необходимости, сортировка тестов должна происходить на основании схемы приоритизации тестов.

Как часть представления результатов и оценки критериев выхода, руководитель тестирования может измерить степень завершенности тестирования. Это должно включать трассировку тестовых сценариев и обнаруженных дефектов к соответствующему базису тестирования. Например, при тестировании, основанном на рисках, по мере того как тесты запускаются и обнаруживают дефекты,

тестировщики могут рассматривать оставшийся, остаточный уровень риска. Это поддерживает использование тестирования, основанного на рисках, в определении правильного момента для релиза. В отчетности по тестированию следует учитывать покрытые и оставшиеся риски, а также преимущества, достигнутые и еще не достигнутые. Для примера отчетности по результатам тестирования на основе покрытия рисков (см. [Black03]).

Наконец, во время завершения тестирования руководитель тестирования должен оценить показатели и критерии успеха, которые имеют отношение к потребностям и ожиданиям заинтересованных лиц тестирования, включая потребности и ожидания клиентов и пользователей с точки зрения качества. Только когда тестирование удовлетворяет этим потребностям и ожиданиям команда тестирования может претендовать на то, что действительно была эффективной.

2.4 Тестовая документация и другие рабочие продукты

Документацию часто разрабатывают в рамках мероприятий по управлению тестированием. Конкретные виды документов по управлению тестированием и их рамки, как правило, весьма различны. Ниже представлены наиболее распространенные виды документов по управлению тестированием, найденные в организациях и на проектах:

- Политика тестирования – описывает назначение и корпоративные цели тестирования.
- Стратегия тестирования – описывает корпоративные методы тестирования, не зависящие от проекта.
- Главный план тестирования (или план тестирования проекта) – описывает применение стратегии тестирования в конкретном проекте.
- Уровневый план тестирования – описывает конкретные активности, которые необходимо выполнить в рамках конкретного уровня тестирования.

Физическая организация этих типов документов может отличаться в зависимости от специфики. В некоторых организациях и на некоторых проектах они могут быть объединены в один документ; в других – они могут находиться в отдельных документах; в некоторых их содержание может проявиться в качестве неписаных, интуитивно понятных или традиционных методологий тестирования. Большие и формальные организации и проекты имеют тенденцию реализовывать эти типы документов как формализованные рабочие продукты, в то время как небольшие и менее формальные организации и проекты имеют тенденцию формализовывать рабочие продукты в меньшем количестве. Эта программа обучения описывает каждый из этих типов документов отдельно, хотя на практике контекст организации и проекта определяет правильное использование каждого типа.

2.4.1 Политика тестирования

Политика тестирования описывает, почему организация занимается тестированием. Она определяет общие цели тестирования, которые организация хочет достигнуть. Эта политика может быть разработана старшим руководителем тестирования организации в сотрудничестве со старшими руководителями тестирования из групп заинтересованных сторон.

В некоторых случаях политика тестирования будет дополнять или будет компонентом более широкой политики качества. Эта политика качества описывает общую ценность и цели, связанные с управлением качеством.

Если формализованная политика тестирования существует, она может быть коротким высокоуровневым документом, который:

- Обобщает ценности, которую организация получает от тестирования.

- Определяет цели тестирования, такие как укрепление уверенности в ПО, обнаружение дефектов в ПО и снижение уровня рисков качества (см. Раздел 2.3.1).
- Описывает, как оценивается эффективность и действенность тестирования при выполнении этих целей.
- Обрисовывает в общих чертах стандартный процесс тестирования, используя в качестве основы базовый процесс тестирования ISTQB.
- Определяет, как организация может улучшить свой процесс тестирования (см. Главу 5).

Политика тестирования должна обращаться к активностям по тестированию для новых разработок, а также для поддержки. Она может также ссылаться на внутренние или внешние стандарты тестирования рабочих продуктов и корпоративную терминологию.

2.4.2. Стратегия тестирования

Стратегия тестирования описывает корпоративную методологию тестирования. Она включает способы, которые используют при тестировании для управления рисками продукта и проекта, разделяющие тестирование на уровни и высокоуровневые активности, связанные с тестированием. (Одна и та же организация может использовать различные стратегии для различных ситуаций, таких как различные жизненные циклы ПО, различные уровни риска или различные нормативные требования.) Стратегия тестирования и описанные в ней процессы и активности должны быть связаны с политикой тестирования. Это должно обеспечивать универсальные критерии входа и выхода для организации одной или нескольких программ.

Как упоминалось выше, стратегии тестирования описывают общие методологии тестирования, которые, как правило, включают:

- Стратегии, основанные на требованиях, такие как тестирование на основе рисков, где команда тестировщиков анализирует базис тестирования для определения покрываемых тестовых условий. Например, при тестировании на основе требований анализ тестирования выделяет тестовые условия из требований, после этого тесты проектируются и реализуются для покрытия этих условий. Результаты тестирования сообщаются в терминах статуса требований (например, требование протестировано и тесты пройдены, требование протестировано и тесты не пройдены, требование еще не полностью протестировано, тестирование требования заблокировано и т.д.).
- Стратегии, основанные на модели, такие как эксплуатационное профилирование, где команда тестировщиков разрабатывает модель (основанную на актуальных или предполагаемых ситуациях) в окружении, в котором система работает, входные данные и условия, которым система подвергается, и то, как система должна вести себя. Например, при основанном на моделях тестировании производительности быстро развивающегося приложения для мобильных устройств можно разрабатывать модели входящего и исходящего сетевого трафика, активных и неактивных пользователей и получать обработку нагрузки, основанной на текущем использовании и развитии проекта в течение времени. Кроме того, модели могли бы быть разработаны с учетом текущей производительности аппаратного обеспечения, ПО, объема данных, сети и инфраструктуры. Модели также могут быть разработаны для идеального, ожидаемого и минимального уровня пропускной способности, времени ответа, распределения ресурсов.
- Методические стратегии, основанные на характеристиках качества, где команда тестировщиков руководствуется predetermined набором тестовых условий, таких как стандарт качества (например, ISO 25000 [ISO2500], который сменил ISO 9126 [ISO9126]), перечень или набор обобщенных логических тестовых условий, которые могут относиться к определенной области, приложению или типу тестирования (например, тестирование безопасности), и использует этот набор тестовых условий от одной итерации к другой или от одного релиза к другому. Например, в эксплуатационном тестировании простого и

стабильного веб-сайта по электронной торговле тестировщики могут использовать список, который определяет ключевые функции, атрибуты и связи для каждой страницы. Тестировщики будут проверять соответствующие элементы этого списка каждый раз при внесении изменений в сайт.

- Стратегии, совместимые с процессом или стандартом, такие как стандарт управления по санитарному надзору за качеством пищевых продуктов и медикаментов США, где команда тестировщиков следует набору процессов, которые определены комитетом по стандартам или иным экспертным советом, где процессы обращаются к документации, прошедшей идентификацию и использующую базис тестирования, тестовые предсказатели и организацию команды по тестированию. Например, в проектах, следующих технологии менеджмента Scrum Agile, в каждой итерации тестировщики анализируют пользовательские истории, которые описывают определенные функции, оценивают объем тестирования каждой функции как часть процесса планирования итерации, определяют тестовые условия для каждой пользовательской истории (их часто называют критериями приемки), выполняют тесты, которые покрывают эти условия, и отчитываются по статусу каждой пользовательской истории (не протестирована, не пройдена или пройдена) во время прохождения тестов.
- Реактивные стратегии, такие как использование атак на основе ошибок, в которой команда тестировщиков ожидает проектирование и реализацию тестов до получения ПО, реагируя на поведение тестируемой системы. Например, при исследовательском тестировании приложения на основе меню можно разработать концепции тестирования, соответствующие функциям, пунктам меню, интерфейсам. Каждому тестировщику дают концепцию тестирования, которую они затем используют для структурирования сессий по исследовательскому тестированию. Тестировщики регулярно отчитываются о результатах сессий тестирования перед руководителем тестирования, который может пересмотреть концепции на основании полученных данных.
- Консультативные стратегии, такие как тестирование, управляемое пользователями, в котором команда тестировщиков основывается на входных данных от одной или нескольких ключевых заинтересованных сторон для определения покрытия тестовых условий. Например, при тестировании совместимости веб-приложения, выполняемого сторонними организациями, компания может предоставлять стороннему поставщику услуг по тестированию список приоритетов версий браузера, антивирусного ПО, операционных систем, типов соединения и другие конфигурационные параметры. Поставщик услуг по тестированию может затем использовать такие методы, как попарное тестирование (для опций высокого приоритета) и классы эквивалентности (для опций низкого приоритета) для создания тестов.
- Стратегии анти-регрессионного тестирования, такие как широкая автоматизация, где команда тестировщиков использует различные методы для управления риском регрессии, особенно функциональной и/или нефункциональной регрессии, автоматизацией регрессионного тестирования на одном или нескольких уровнях. Например, в регрессионном тестировании веб-приложения тестировщики могут использовать инструмент автоматизации тестирования, основанный на графическом интерфейсе, для автоматизации стандартных и исключительных сценариев использования приложения. Эти тесты затем выполняются каждый раз при модификации приложения.

Различные стратегии могут комбинироваться. Определенные стратегии должны соответствовать потребностям и возможностям организации, и организации могут адаптировать стратегии для соответствия особенностям операций и проектов.

Стратегии тестирования могут описывать выполняемые уровни тестирования. В таких сценариях это может давать управление критериями входа и выхода для каждого уровня и взаимоотношением между уровнями (например, разделение целей тестового покрытия).

Стратегия тестирования может также описывать следующее:

- процедуры интеграции;
- методы разработки спецификации тестов;
- независимость тестирования (которое может отличаться для разных уровней);
- обязательные и необязательные стандарты;
- тестовое окружение;
- автоматизацию тестирования;
- инструменты тестирования;
- повторное использование ПО и тестов рабочих продуктов;
- подтверждающее и регрессионное тестирование;
- контроль тестирования и отчетность;
- тестовые измерения и метрики;
- управление дефектами;
- подход к управлению конфигурацией тестового обеспечения;
- роли и ответственности.

Различные стратегии тестирования могут быть нужны для краткосрочных и длительных проектов. Различные стратегии тестирования подходят для различных организаций и проектов. Например, в случае, когда затрагиваемая безопасность или надежность приложения критична, очень интенсивная стратегия может быть более подходящей, чем другие сценарии. Кроме того, стратегия тестирования отличается в зависимости от различных моделей разработки.

2.4.3. Главный план тестирования

Главный план тестирования охватывает все работы по тестированию, выполняемые в конкретном проекте, включающие отдельные уровни, которые будут выполнены, связи между уровнями, и соответствие уровней тестирования и активностей по разработке. Главный план тестирования должен решать, как тестировщики могут реализовать стратегию тестирования в этом проекте (т.е. подход к тестированию). Главный план тестирования должен соответствовать политике и стратегии тестирования, и, в конкретных областях, где их нет, он должен объяснять те отклонения и исключения, включая любое потенциальное влияние, возникающее вследствие отклонений. Например, если в некоей корпоративной стратегии тестирования заложено проведение одного полного регрессионного тестирования системы до выпуска релиза, но в текущем проекте не будет регрессионного тестирования, то план тестирования должен объяснить, почему оно не было запланировано и что должно быть сделано для снижения ожидаемых рисков в отличие от обычной стратегии. План тестирования должен также включать многие другие эффекты, которые ожидаются от этого различия. Например, игнорирование регрессионного тестирования может потребовать планирования релиза обновленной эксплуатационной версии через месяц после первоначального выпуска проекта. Главный план тестирования должен дополнять план проекта или руководство по работе с продуктом, в котором могут быть описаны трудозатраты по тестированию, которое является частью большего проекта или эксплуатации.

Хотя особенность содержания и структуры главного плана тестирования различаются в зависимости от организации, стандартов документации, формальной стороны проекта, типичные разделы главного плана тестирования содержат:

- Тестируемые и нетестируемые элементы.
- Тестируемые и нетестируемые характеристики качества.
- Программу тестирования и бюджет (которые должны быть увязаны с бюджетом проекта или эксплуатации).
- Циклы выполнения тестов и их связь с планом выпуска ПО.

- Взаимоотношение и результаты тестирования других специалистов или отделов.
- Определение, какие элементы тестирования входят и не входят в состав каждого описанного уровня.
- Конкретные критерии входа, продолжения (приостановки/возобновления) и выхода для каждого уровня и взаимоотношений между уровнями.
- Риски проекта по тестированию.
- Общее управление тестированием.
- Ответственность за выполнение каждого уровня тестирования.
- Входные и выходные данные для каждого уровня тестирования.

На меньших проектах или операциях, где формализован только один уровень тестирования, главный план тестирования и тестовый план этого формализованного уровня тестирования часто объединяются в одном документе. Например, если тесты системы формализованы только для одного уровня с обычным компонентным и интеграционным тестированием, выполняемым разработчиками, и обычным приемочным тестированием, выполняемым клиентами, как часть процесса бета-тестирования, тогда план тестирования системы может включать элементы, упоминаемые в этом разделе.

Кроме того, тестирование, как правило, зависит от других действий в рамках проекта. Если эти мероприятия недостаточно документированы, особенно в части влияния и связей с тестированием, то те темы, которые связаны с активностями, могут быть охвачены в главном плане тестирования (или в соответствующем плане тестирования уровня). Например, если процесс управления конфигурацией не документирован, то план тестирования должен определить, как объекты тестирования будут переданы команде тестировщиков.

2.4.4. Уровневый план тестирования

Уровневые планы тестирования описывают определенные активности, выполняемые в рамках каждого уровня тестирования, или, в некоторых случаях, типа тестирования. Уровневый план тестирования распространяется, если необходимо, на главный план тестирования для конкретного уровня или типа тестирования, которые документированы. Они содержат график, задачи и вехи, не обязательно включенные в главный план тестирования. Кроме того, степень различия стандартов и применяемых шаблонов к спецификации тестов на различных уровнях могут быть покрыты в уровневом плане тестирования.

На менее формальных проектах или операциях индивидуальный план тестирования является единственным зафиксированным документом по управлению тестированием. В таких ситуациях некоторые из информационных элементов, упомянутых выше в этом разделе, могут быть включены в этот план тестирования.

Для гибких (Agile) проектов планы тестирования спринта или итерации могут занять место уровневого плана тестирования.

2.4.5. Управление рисками проекта

Важная часть, присущая планированию, включает работу с рисками проекта. Риски проекта можно идентифицировать с использованием процессов, похожих на описанные в Разделе 2.3. Когда риски проекта определены, они должны быть переданы менеджеру проекта для принятия мер. Снижение таких рисков не всегда находится в компетенции организации по тестированию. Однако руководитель тестирования может и должен минимизировать такие риски проекта, как:

- тестовое окружение и готовность оборудования;
- готовность и уровень подготовки тестировщиков;
- недостаток стандартов, регламентов и методов при тестировании.

Подходы к управлению рисками проекта включают предыдущую подготовку тестового обеспечения, предварительное тестирование тестового окружения, предварительное тестирование ранних версий продукта, применение более строгих критериев входа к тестированию, обеспечение тестируемости требований, участие в рецензировании ранних рабочих продуктов проекта, участие в управлении изменениями и мониторинг хода выполнения проекта и качества.

Когда риск проекта был выявлен и проанализирован, существуют четыре главные функции управления этим риском:

1. Снизить риск при помощи предупредительных мер для того, чтобы снизить вероятность и/или влияние.
2. Сделать планы непредвиденных обстоятельств снижения влияния, если риск становится реальным.
3. Передать риск другому участнику для контроля.
4. Игнорировать или принять риск.

Выбор лучшего решения зависит от выгод и преимуществ созданного решения, а также от стоимости, и возможно, от дополнительных рисков, связанных с решением. Когда план непредвиденных обстоятельств определен для проектного риска, лучшие практики определяют триггер (который может решить, когда и как должен быть вызван план непредвиденных обстоятельств), и владельца (кто должен выполнять план непредвиденных обстоятельств).

2.4.6 Другие рабочие продукты тестирования

Тестирование включает создание нескольких рабочих продуктов, таких как отчеты о дефектах, спецификации сценария тестирования, протоколы тестирования. Большая часть этих рабочих продуктов создается тест-аналитиком и техническим тест-аналитиком. В программе обучения «Тест-Аналитик» рассматриваются разработка и документирование этих рабочих продуктов. Руководитель тестирования должен следить за последовательностью и качеством этих рабочих продуктов во время следующих активностей:

- Создание и мониторинг метрик, которые отображают качество этих рабочих продуктов, таких как процент отклоненных отчетов о дефектах.
- Взаимодействие с тест-аналитиками и техническими тест-аналитиками для выбора и настройки соответствующих шаблонов для этих рабочих продуктов.
- Взаимодействие с тест-аналитиками и техническими тест-аналитиками по созданию стандартов для этих рабочих продуктов, таких как степень необходимой детализации тестов, протоколов и отчетов.
- Рецензирование рабочих продуктов тестирования, используя соответствующие методы, соответствующих участников и заинтересованных сторон.

Выбранный жизненный цикл ПО, применимые стандарты и регламенты могут повлиять на степень, тип и специфичность документации по тестированию. Риски качества продукта и риски проекта связаны с особенностями разрабатываемой системой, среди других соображений.

Существуют различные источники шаблонов рабочих продуктов тестирования, такие как IEEE 829 [IEEE829]. Для руководителя тестирования важно помнить, что документы IEEE 829 проектируются для использования в любой отрасли. В связи с этим шаблоны содержат высокую степень детализации, что можно или нельзя применить в определенной организации. Создание стандартных шаблонов для использования в определенной организации – это лучший способ адаптации документов IEEE 829. Логичное применение шаблона снижает требования к подготовке и помогает объединить процессы в организации.

Тестирование также включает создание отчетов по результатам тестирования, которые, как правило, создает руководитель тестирования и которые описаны позже в этой главе.

2.5 Оценка затрат на тестирование

Оценка как управленческая активность – это создание приблизительных целей по затратам и датам завершения, связанным с активностями, включенными в конкретные операции или проекты. Лучшие оценки:

- Представляют коллективный опыт специалистов-практиков и имеют поддержку вовлеченных участников.
- Предусматривают конкретный, подробный список затрат, ресурсов, задач и вовлеченных участников.
- Представляют для каждой оцениваемой активности наиболее вероятную стоимость, трудозатраты и продолжительность.

Известно, что оценка стоимости разработки ПО и систем сталкивается с трудностями, как техническими, так и политическими, хотя передовые подходы управления проектами для оценки хорошо известны. Оценка затрат на тестирование является применением этих передовых подходов к активностям по тестированию, связанным с проектом или операцией.

Оценка затрат на тестирование должна охватывать все активности, участвующие в тестировании, описанные в Главе 1. Оценки стоимости, трудозатрат и особенно продолжительности выполнения тестов представляют наибольший интерес для менеджмента, так как выполнение тестов обычно относится к важнейшим этапам проекта. Однако оценка затрат на тестирование имеет тенденцию быть сложной и ненадежной в том случае, когда общее качество ПО является низким или неопределенным. Кроме того, знакомство и опыт работы с системой могут повлиять на качество оценки. Распространенным подходом является оценка количества сценариев тестирования, требуемых при тестировании, но этот подход применим при условии, что количество ошибок в тестируемом ПО мало. Предположения, сделанные при оценке, должны быть задокументированы как часть оценки.

Оценка затрат на тестирование должна связывать все факторы, которые могут повлиять на стоимость, трудозатраты и продолжительность активностей по тестированию. Эти факторы включают (но не ограничиваются) следующие:

- требуемый уровень качества системы;
- размер тестируемой системы;
- предыстория тестирования на предшествующих проектах по тестированию, которые могут быть дополнены промышленными или контрольными данными из других организаций;
- факторы процесса, включающие стратегию тестирования, развитие или поддержку жизненного цикла и зрелости процесса, и точность оценки проекта;
- материальные факторы, включающие автоматизацию тестирования и инструменты, тестовая среда(-ы), данные для тестирования, среды разработки, проектная документация (т.е. требования, проекты) и повторно применяемые рабочие продукты тестирования;
- человеческие факторы, включающие менеджеров и технических руководителей, исполнительный и старший менеджмент, обязательства и ожидания, навыки, опыт и отношения в проектной команде, стабильность проектной команды, взаимосвязь проектной команды, тесты и окружение поддержки для исправления дефектов, доступность опытных исполнителей и консультантов и знание предмета;

- сложность процесса, технология, организация, число участников тестирования, структура и расположение субподрядных команд;
- резкое увеличение объемов, подготовка;
- усвоение или разработка новых инструментов, технологии, процессов, техник, собственного оборудования или большого количества программ для тестирования;
- требования для высокой степени детализации спецификации тестов, особенно соответствующие неизвестным стандартам в документации;
- суммарный расчет входящих компонентов, особенно для интеграционного тестирования и разработки тестов;
- нестабильные тестовые данные (например, данные, которые зависят от времени).

При этом качество ПО, предоставленного для тестирования, является главным фактором, который руководители тестирования должны учитывать при оценке. Например, если разработчики принимают лучшие практики, такие как автоматизированное модульное тестирование и непрерывную интеграцию, то примерно 50% ошибок могут быть найдены и устранены непосредственно перед предоставлением кода команде тестировщиков (подробнее об эффективности устранения дефектов при этих подходах (см. [Jones11]). Некоторые определяют, что гибкие (Agile) методологии, включающие разработку через тестирование, приводят к достижению высоких уровней качества в продуктах, предоставляемых для тестирования.

Оценка затрат может быть как восходящей, так и нисходящей. Методы, которые могут быть использованы при оценке затрат на тестирование, отдельно или в сочетании, включают следующее:

- интуиция, догадки и прошлый опыт;
- структурный план проекта (СПП);
- встречи команды для оценки (например, Wide Band Delphi);
- стандарты и правила компании;
- доля общих трудозатрат проекта или штатное расписание;
- организационные история и метрики, включающие модели на основе метрик, которые оценивают число ошибок, число тестовых циклов, количество сценариев тестирования, средние трудозатраты каждого теста и количество регрессионных циклов;
- средние по отрасли и прогнозирующие модели оценки трудозатрат, использующие функциональные точки, строки кода, оцененные трудозатраты разработчиков или другие параметры проекта (например, см. ([Jones07]).

В большинстве случаев, после того как подготовлена оценка затрат, нужно представить ее руководству вместе с обоснованием (см. Раздел 2.7). Далее следует согласование, часто заканчивающееся переработкой оценки. В идеале финальная оценка затрат на тестирование представляет собой окончательный баланс организационных и проектных целей в областях качества, графика, бюджета или функций.

Важно помнить, что оценка основана на информации, доступной на момент ее подготовки. В начале проекта информация может быть сильно ограничена. Кроме того, информация, доступная в начале проекта, может измениться с течением времени. Для того, чтобы оставаться точной, оценка должна быть обновлена для отражения новой и измененной информации.

2.6 Определение и использование метрик тестирования

Менеджерское клише гласит, что то, что поддается измерению, можно выполнить. Кроме того, то, что нельзя измерить, нельзя выполнить, так как то, что нельзя измерить, легко игнорировать.

Следовательно, важно, чтобы правильный набор метрик был установлен для любой деятельности, включая тестирование.

Метрики тестирования могут быть классифицированы как принадлежащие к одной или нескольким следующим категориям:

- Метрики проекта, которые измеряют прогресс к установленным в проекте критериям выхода, таким как процент выполненных, пройденных и не пройденных сценариев тестирования.
- Метрики продукта, которые измеряют некоторый атрибут продукта, такой как степень тестирования или плотность дефектов.
- Метрики процесса, которые измеряют возможности тестирования или процесса разработки, такие как процент дефектов, обнаруженных при тестировании.
- Человеческие метрики, которые измеряют возможности отдельных сотрудников или групп, таких как реализация сценариев тестирования в рамках предоставленного плана-графика.

Любая предоставляемая метрика может относиться к двум, трем или даже четырем категориям. Например, диаграмма тенденции, демонстрирующая частоту поступления дефектов в день, может быть связана с критериями выхода (ноль новых дефектов, найденных за неделю), качеством продукта (тестирование больше не может выявить в нем дефектов) и возможностью процесса тестирования (тестирование обнаружило большое количество дефектов на ранней стадии во время выполнения тестов).

Человеческие метрики требуют особого внимания. Менеджеры иногда используют метрики процесса в качестве человеческих метрик, что приводит к катастрофическим результатам, когда поведение людей направлено, прежде всего, на достижение высоких показателей метрик. Соответствующая мотивация и оценка тестировщиков затрагивается в Главе 7 этой программы обучения и в программе обучения «Эксперт в управлении тестированием») [ISTQB ETL SYL].

На Продвинутом уровне мы касаемся главным образом использования метрик для измерения прогресса в тестировании, т.е. проектных метрик. Некоторые из проектных метрик, используемые в измерении прогресса тестирования, связаны с продуктом и процессом. В программе обучения «Эксперт в управлении тестированием» [ISTQB ITP SYL] содержится больше информации об управлении использованием продуктовых и процессных метрик. В программе обучения «Эксперт улучшения процесса тестирования» содержится больше информации по использованию процессных метрик.

Использование метрик дает возможность тестировщикам отчитываться по результатам в непротиворечивой форме и включать связь с фиксацией прогресса в течение долгого времени. Руководители тестирования часто требуют представлять метрики на совещаниях, которые могут посещать заинтересованные стороны разного уровня, начиная от технического персонала до высшего руководства. Так как метрики часто используются при определении общего успеха проекта, требуется особая осторожность при определении того, что отслеживать, как часто готовить отчет и какой метод использовать при предоставлении информации. При этом руководитель тестирования должен учитывать следующее:

- Определение метрик. Должен быть определен ограниченный набор полезных метрик. Метрики должны быть определены в соответствии с конкретными целями проекта, процесса и/или продукта. Метрики должны быть сбалансированы, так как отдельная метрика может создать неверное впечатление о статусе или тенденции. Так как метрики были определены, их адаптация должна быть согласована всеми заинтересованными сторонами, чтобы избежать беспорядка при обсуждении измерений. При этом часто существует тенденция

определять слишком много метрик вместо того, чтобы сосредоточиться на наиболее подходящих.

- Отслеживание метрик. Подготовка отчета и объединение метрик должно быть автоматизировано настолько, насколько возможно для сокращения времени на сбор и выполнение измерения. Варианты измерения конкретной метрики в течение длительного времени могут отразить информацию, кроме интерпретации, согласованной во время этапа определения метрик. Руководитель тестирования должен быть готов внимательно анализировать возможное отклонение измерения от ожидаемого, и причины этого расхождения.
- Отчетность по метрикам. Цель - обеспечить каждого непосредственным пониманием информации для целей управления. Презентации могут демонстрировать текущее состояние метрики в определенное время или показывать динамику метрики в течение длительного времени, так чтобы можно было оценить тенденции.
- Точность метрик. Руководитель тестирования может только проверять информацию, которая была предоставлена. Сделанные для метрик измерения могут не отражать настоящий статус проекта или могут отражать слишком позитивные или негативные тенденции. Перед любой датой презентации руководитель тестирования должен прорецензировать их на предмет точности и в отношении данных, которые они вероятно передают.

Существует пять основных измерений, которые контролируют прогресс выполнения тестов:

- Риски продукта (качества)
- Дефекты
- Тесты
- Покрытие
- Уверенность

Риски продукта, дефекты, тесты и покрытие могут быть зачастую измерены и отображены конкретными способами в течение проекта или активности. Если эти измерения связаны с определенными критериями выхода, указанными в плане тестирования, то они могут обеспечивать собой объективный стандарт для решения этого вопроса. Уверенность – это измеряемое путем опроса или использующее покрытие как дополнительную метрику; впрочем, в отчетах степень достоверности отражается субъективно.

Метрики, связанные с рисками продукта, включают:

- процент рисков, полностью покрытых при прохождении тестов;
- процент рисков, для которых некоторые или все тесты закончились неудачно;
- процент рисков, которые еще полностью не протестированы;
- процент покрытых рисков, отсортированных по категории риска;
- процент рисков, определенных после начала анализа качества рисков.

Метрики, связанные с дефектами, включают:

- общее число найденных дефектов в сопоставлении с числом исправленных дефектов;
- среднее время наработки на отказ или частота поступления отказа;
- количество сбоев или процент дефектов, распределенных по следующим категориям:
 - конкретные группы тестов или компоненты;
 - корневая причина;
 - источник дефектов (например, спецификация требований, новая функция, регрессия и т.п.);

- тесты релизов;
- Начало, выполнение или завершение фазы; приоритет/серьезность ошибки;
- отвергнутые или дублированные отчеты о дефектах.
- тенденции задержки от сообщения о дефекте до его закрытия;
- число исправленных дефектов, которые порождают новые дефекты (иногда называемые дочерними дефектами).

Метрики, связанные с тестами, включают:

- общее количество запланированных, реализованных, запущенных, пройденных, потерпевших неудачу, заблокированных и пропущенных тестов;
- статус подтверждающих и регрессионных тестов, включающий тенденции и общее количество не успешных регрессионных и подтверждающих тестов;
- плановое и фактическое время, запланированное на тестирование в день;
- доступность тестового окружения (процент времени, запланированного на тестирование, когда тестовое окружение используется командой тестировщиков).

Метрики, связанные с покрытием тестирования, включают:

- покрытие требований и дизайна;
- покрытие риска;
- покрытие окружения/конфигурации;
- покрытие кода.

Метрики должны быть также связаны с активностями базового процесса тестирования (описанными в программе обучения «Базовый курс» и в этой программе обучения). При этом метрики могут быть использованы на протяжении всего процесса тестирования для мониторинга процесса тестирования, а также для прогресса достижения целей проекта.

Метрики для мониторинга планирования тестов и контроля активностей могут включать:

- риск, требования и другой элемент покрытия базиса тестирования;
- обнаружение дефекта;
- запланированные и фактические часы работ по разработке тестового окружения и выполнения сценариев тестирования.

Метрики для мониторинга активностей анализа тестирования могут включать:

- количество определенных тестовых условий;
- количество дефектов, найденных при анализе тестов (например, при определении рисков или других тестовых условий, использующих базис тестирования);

Метрики для мониторинга активностей по проектированию тестов могут включать:

- процент условий тестирования, покрытых сценариями тестирования;
- количество дефектов, найденных при проектировании тестов (например, при разработке тестов на основе базиса тестирования).

Метрики для мониторинга активностей по реализации тестов могут включать:

- процент конфигурации тестовой среды;
- процент загруженных записей тестовых данных;

- процент автоматизированных сценариев тестирования.

Метрики для мониторинга активностей по выполнению тестов могут включать:

- процент выполненных, пройденных и не пройденных запланированных сценариев тестирования;
- процент тестовых условий, покрытых пройденными (и/или) выполненными сценариями тестирования;
- запланированное и фактическое число найденных / исправленных дефектов;
- планируемое и фактическое покрытие.

Метрики для контроля прогресса при тестировании и действиями по завершению будут включать сопоставление контрольных точек, критериев запуска и критериев выхода (определенных и согласованных при планировании тестов), которые могут включать следующее:

- количество запланированных и выполненных тестовых условий, сценариев тестирования или спецификаций тестирования, распределенных по тому, пройдены они или нет;
- общее количество дефекты с разделением по серьезности, приоритету, текущему состоянию, затронутым компонентам или другой группировке (см. Главу 4);
- количество требуемых, принятых, выполненных и протестированных изменений;
- планируемая стоимость в соответствии с фактической;
- планируемая длительность в соответствии с фактической;
- планируемые даты этапов тестирования в соответствии с фактическими;
- планируемые даты этапов проекта, связанные с тестированием (например, запрет на внесение изменений в код) в соответствии с фактическими;
- статус рисков продукта (качества), зачастую с разбивкой по митигируемым и не митигируемым, основным областям риска, новым рискам, выявленным при анализе тестирования и т.д.;
- процент снижения объема работ по тестированию, стоимости или времени в связи с блокирующими событиями или запланированными изменениями;
- статус подтверждающих и регрессионных тестов.

Метрики для мониторинга при завершении тестирования могут включать:

- процент реализованных, пройденных, не пройденных, заблокированных и пропущенных сценариев тестирования при выполнении тестов;
- процент сценарии тестирования, зарегистрированных в репозитории повторно используемых сценариев тестирования;
- процент автоматизированных сценариев тестирования, или планируемых к автоматизации в соответствии с актуальными автоматизированными сценариями тестирования;
- процент сценариев тестирования, интегрированных в регрессионные тесты;
- процент отчетов о дефектах, исправленных/не исправленных;
- процент рабочих продуктов, которые определены и заархивированы.

Кроме того, стандартные методы управления проектом, такие, как структурная декомпозиция работ проекта, часто используются для мониторинга процесса тестирования. В гибких (Agile) командах тестирование является частью успеха историй использования в построении срочных графиков. Если используются бережливые методы управления, процесс тестирования, основанный на последовательности историй, зачастую контролируется при наличии карт пользовательских историй, перемещающихся через колонки в доске Канбан.

Важно, чтобы руководитель тестирования понимал, как интерпретировать и использовать покрытие метрик для понимания и отслеживания статуса тестирования. Для более высоких уровней тестирования, таких как системное тестирование, системное интеграционное тестирование и приемочное тестирование, важнейшей основой тестов являются обычно рабочие продукты, такие, как спецификации требований, спецификация дизайна, сценарии использования, пользовательские истории, риски продукта, поддерживаемые окружения и поддерживаемые конфигурации. Структурные метрики покрытия кода применяются чаще для низких уровней тестирования, таких как модульное тестирование (например, покрытие условий и ветвей) и компонентно-интеграционное тестирование (например, покрытие интерфейса). В то время как руководитель тестирования может использовать метрики покрытия для измерения степени, до которой эти тесты изучают структуру тестируемой системы, отчетность результатов высокоуровневых тестов, как правило, не должна включать метрики покрытия кода. Кроме того, руководитель тестирования должен понимать что, даже если модульное тестирование и компонентно-интеграционное тестирование на 100% достигнут целей структурного покрытия, дефекты и риски качества остаются для более высоких уровней тестирования.

Для заданного набора метрик измерения могут быть представлены устно в повествовательной форме, в цифровой форме, в виде таблиц и графиков. Измерения могут быть использованы в разнообразных целях, включая:

- Анализ, который выявляет, какие тенденции и причины могут быть обнаружены в результатах тестов.
- Отчет, который сообщает найденные данные тестирования участникам или заинтересованным сторонам проекта.
- Контроль, который изменяет направление тестирования или проекта в целом, и отслеживает результаты корректировки направления.

Соответствующие способы сбора, анализа и подготовки отчетности по этим измеримым тестам зависят от конкретных информационных запросов, целей и способностей специалистов, которые будут использовать измерения. Кроме того, конкретное содержание отчетов по тестированию может отличаться, в зависимости от аудитории.

В целях контроля тестирования важно, чтобы метрики во время процесса тестирования (как только планирование тестирования выполнено) обеспечивали руководителя тестирования необходимой информацией для руководства усилиями, прикладываемыми к тестированию для успешного завершения миссии тестирования, стратегий и целей. Более того, планирование должно учитывать эту необходимую информацию, и мониторинг должен включать сбор всех необходимых метрик рабочих продуктов. Объем требуемой информации и усилия, затраченные на сбор, зависят от различных факторов проекта, включающих размер, сложность и риск.

Контроль тестирования должен соответствовать как информации, созданной при тестировании, так и изменяющимся условиям, в которых существуют или пытаются существовать проект. Например, если при динамическом тестировании обнаруживаются кластеры дефектов в областях, считавшихся маловероятными для появления большого количества дефектов, или, если период выполнения тестов сократится из-за задержки в начале тестирования, анализ рисков и план должны быть пересмотрены. Это может привести к изменению приоритетов тестов и перераспределению оставшихся усилий для выполнения тестов.

Если отклонение от плана тестирования будет обнаружено при помощи отчета о тестировании, то должен быть выполнен контроль тестирования. Контроль тестирования направлен на перенаправление проекта и/или тестирования в более благоприятное направление. Когда

использование результатов тестирования приводит к влиянию или измерению для достижения контроля проекта, должны приниматься во внимание следующие функции:

- Пересмотр анализа рисков качества, приоритетов тестирования и/или планов тестирования.
- Добавление ресурсов или, иначе, увеличение проекта или объема работ по тестированию.
- Перенос даты релиза.
- Ослабление или усиление критериев выхода тестирования.
- Изменение содержания проекта (функционального и/или нефункционального).

Реализация таких функций требует консенсуса между заинтересованными сторонами проекта и менеджерами проекта или операционными менеджерами.

Информация, представленная в отчете о тестировании, должна во многом определяться необходимой информацией для целевой аудитории, например, менеджмента проекта или бизнес-менеджмента. Для менеджера проекта интересно иметь детальную информацию по дефектам, для бизнес-менеджера статус продуктовых рисков может быть ключевой информацией при предоставлении отчетности.

2.7 Бизнес-ценность тестирования

Руководитель тестирования должен так работать, оптимизируя тестирование, чтобы добиваться хорошей ценности для бизнеса. Тестирование не предоставляет чрезмерно хорошую ценность для бизнеса, потому что тестирование накладывает серьезные задержки и затраты. Тестирование не предоставляет слишком маленькую ценность для бизнеса, потому что слишком много дефектов могут быть переданы пользователям. Оптимальный вариант находится между этими двумя значениями. Руководитель тестирования должен помочь заинтересованным лицам в тестировании понять этот оптимальный вариант и ценность проведения тестирования.

Пока большинство организаций, оценивают ценность тестирования, небольшое количество менеджеров, в том числе руководители тестирования, могут определить количественно, описать или явно аргументировать эту ценность. Кроме того, много руководителей тестирования, тест-лидов и тестировщиков фокусируются на тактических деталях тестирования (определенных аспектах для задач или уровня тестирования) при игнорировании больших стратегических (высокоуровневых) проблем, связанных с тестированием, о котором заботятся другие участники проекта, в особенности менеджеры.

Тестирование предоставляет ценность для организации, проекта и/или процессов как количественно, так и качественно, когда:

- Количественные значения включают нахождение дефектов, которые предотвращаются или устраняются до выпуска релиза, дефектов, которые известны до выпуска релиза (не устраненные, но задокументированные, возможно с обходными решениями), снижение риска в соответствии с запуском тестов и предоставление информации о проекте, процессе и статусе продукта.
- Качественные значения включают улучшение репутации для качества, более плавные и предсказуемые релизы, увеличение доверия, защиту от юридической ответственности и снижение риска потери целых задач или уменьшения срока службы.

Руководители тестирования должны понимать, какие ценности применимы для их организации, проекта и/или процессов, и быть в состоянии взаимодействовать по вопросам тестирования с точки зрения этих ценностей.

Известный метод измерения количественной ценности и эффективности тестирования называют стоимостью качества (или, иногда, стоимостью низкого качества). Стоимость качества включает классификацию проекта и операционные затраты по четырем категориям, связанным с затратами на дефекты продукта:

- 1) Затраты на предотвращение (например, обучение разработчиков написанию очень удобного в сопровождении или безопасного кода).
- 2) Затраты на обнаружение (например, написание тестовых сценариев, конфигурация тестовых сред и рецензирование требований).
- 3) Затраты на внутренние ошибки (например, исправление дефектов, обнаруженных во время тестирования или рецензирования, перед поставкой пользователям).
- 4) Затраты на внешние ошибки (например, стоимость технической поддержки, связанной с дефектным программным обеспечением, предоставленным клиентам).

Часть бюджета тестирования является затратами на обнаружение (т.е. деньги, которые были бы потрачены даже если тестировщики не найдут дефекты, например деньги, потраченные на разработку тестов), в то время как остаток является затратами на внутренние ошибки (т.е. актуальные затраты, связанные с найденными дефектами). Общие затраты на обнаружение и внутренние ошибки обычно значительно ниже затрат на внешние ошибки, которые делают превосходную ценность тестирования. Путем определения затрат в этих четырех категориях руководители тестирования могут создать убедительный бизнес-сценарий для тестирования.

2.8 Распределенное, аутсорсинговое и внутреннее тестирование.

Часто некоторые или даже все усилия по тестированию осуществляются людьми, находящимися в разных локациях (местах), работающими в разных компаниях или выделенными в отдельный проект.

Если тестирование проводится в разных местах, то такое тестирование называется распределённым. Если тестирование проводится в одном или разных местах людьми, не работающими в той же компании, что и разработчики, то такое тестирование называется аутсорсинговым. Если тестирование проводится людьми, работающими с командой разработчиков в одной организации, но в разных командах, то такое тестирование называется внутренним.

При любом подходе к тестированию необходимы хорошие каналы коммуникации и хорошо определенные цели и ожидания от выполнения задач. Команда разработчиков должна меньше использовать неформальные каналы коммуникации, такие, как разговоры в коридорах и общение в неформальной обстановке. Важно определить способы коммуникации, включая такие, как эскалация проблем, типы информации для передачи и методы коммуникации. Каждый член команды должен иметь четкое представление о своей роли и обязанностях, а также о роли и ответственности других членов команды, чтобы избежать непонимания и нереалистичные ожидания. Проблемы коммуникации и ожиданий особенно важны для разных местоположений, временных зон, культурных и языковых различий.

Для всех задач тестирования необходимо определить и методологии тестирования. Это особенно важно, когда тестирование распределено между группами или осуществляется внешней командой. Если две тестовые команды пользуются разной методологией, или тестовая команда использует методологию, не согласующуюся с методологией команды разработчиков, это может привести к очень серьезным проблемам, особенно во время выполнения тестов. Например, если разработчики используют гибкую (agile) методологию разработки программного обеспечения, а тестировщики пользуются методами на основе жизненного цикла программного обеспечения, то попытки обеспечить своевременность и качество тестирования приведут к серьезным разногласиям.

Для распределенного тестирования очень важно четко разделить и обосновать разделение работ по разным командам. Без подобной инструкции может получиться так, что самая компетентная группа не занимается частью работ, соответствующей ее квалификации. К тому же если команда не знает, за что именно она ответственна, она может не делать ту работу, которую от нее ожидают. Ожидания от каждой команды должны быть строго оговорены. Без правильного управления распределение работ по тестированию может иметь пробелы (что увеличивает риск при поставке продукта), или перекрытия (что уменьшает эффективность).

И наконец, для подобных работ по тестированию критически важна уверенность проектной команды в том, что все выполняют свои роли должным образом, несмотря на организационные, культурные, языковые и географические особенности.

2.9 Управление на основе промышленных стандартов

В программах как Базового, так и Продвинутого уровня есть ссылки на различные стандарты. Эти стандарты описывают циклы разработки программного обеспечения, процессы тестирования, характеристики качества, способы рецензирования и управления дефектами. Руководитель тестирования должен быть знаком с этими стандартами, политикой их применения использования этих стандартов в компании и, независимо от того, требуются они или нет, понимать необходимость и полезность их применения.

Стандарты создаются на различных уровнях, таких, как:

- Международные или с международными целями.
- Национальные, в том числе национальные интерпретации международных стандартов.
- Специфические для сфер деятельности, в том числе разработка или адаптация международных и национальных стандартов под особенности конкретной сферы деятельности.

Среди международных ассоциаций по стандартизации выделяют ISO и IEEE. ISO (также называемая IOS) - Международная Организация по Стандартизации, состоит из членов, представляющих в своих странах национальные интересы, относящиеся к стандартизируемой области. Эта ассоциация выпустила несколько стандартов, используемых для тестирования ПО, такие как ISO 9126 (замененный впоследствии на ISO 25000), ISO 12207 [ISO12207] и ISO 15504 [ISO15504].

IEEE – Институт инженеров по электротехнике и радиоэлектронике – профессиональная организация, базирующаяся в США, но имеющая представителей в более чем ста странах по всему миру. Эта организация выпустила несколько стандартов, которые применяются в тестировании ПО, такие как IEEE 829 [IEEE829] и IEEE 1028 [IEEE1028].

Многие страны имеют свои собственные национальные стандарты. Некоторые из этих стандартов используются в области тестирования ПО. Например, в Великобритании - стандарт BS 7925-2 [BS7925-2], который содержит информацию о многих методах проектирования тестов, описанных в программах Продвинутого тест-аналитика и Продвинутого технического тест-аналитика.

Некоторые стандарты являются специфичными в определенных областях, и некоторые из них содержат описание тестирования ПО, понятие качества и разработки ПО. Например, в авиационной области стандарт Федерального агентства по Авиации (США) DO-178B (и его Европейский эквивалент ED 12B) касается использования ПО в гражданской авиации. Этот стандарт устанавливает определенные уровни критериев структурного покрытия, основанных на уровнях критичности тестируемого ПО.

Другой пример подобного стандарта - из области медицины – стандарт '21 CFR Часть 820' [FDA21] Администрации США по контролю продуктов питания и лекарств. Этот стандарт рекомендует определённые структурные и функциональные техники тестирования. В нем также есть рекомендации стратегий и принципов тестирования, согласующиеся с программой ISTQB.

В некоторых случаях на тестирование оказывают воздействие стандарты и широко распространённые методы, которые не относятся непосредственно к тестированию, но влияют на процесс разработки ПО, частью которого является тестирование. Один из примеров – система совершенствования процессов разработки ПО CMMI®. Она включает две ключевые области – верификацию и валидацию, на которые часто ссылаются как на уровни тестирования (такие, как системное и приемочное тестирование соответственно). Она также влияет на стратегию тестирования, которая подразумевает наличие тестирования, основанного на требованиях.

Еще три важных примера: PMBOK от PMI, PRINCE2® и ITIL®. PMI и PRINCE2 обычно используются как методологии управления проектами в Северной Америке и Европе соответственно. ITIL – это система для проверки того, что ИТ группа предоставляет значимые сервисы для своей организации. Терминология и функции, определенные в этих системах, отличаются от определений в ISTQB. Работая в компаниях, использующих PMBOK, PRINCE2 и/или ITIL, руководитель тестирования должен знать выбранные процессы, их реализацию и терминологию настолько, чтобы работать эффективно в этом контексте.

Какие бы стандарты и методологии ни использовались, важно помнить, что они разработаны коллективами профессионалов. Стандарты отражают как коллективный опыт и знания их создателей, так и их слабости. Руководители тестирования должны знать стандарты, применяемые в их области работы, независимо от того, рекомендуются ли формальные (международные, национальные или специфичные для области) или внутренние стандарты.

В случае использования нескольких стандартов необходимо понимать, что некоторые стандарты не соответствуют друг другу или даже предоставляют конфликтующие определения. Руководитель тестирования должен обосновать необходимость использования нескольких стандартов в определённом контексте, в котором осуществляется тестирование. Информация, взятая из стандартов, может быть полезной или мешать работе проекта. Однако стандарты могут предоставить ссылки на проверенные практики и помочь определить базис для организации процесса тестирования.

В некоторых случаях соответствие стандартам является обязательным и это влияет на тестирование. Руководитель тестирования должен знать эти требования, следуя стандартам, и быть уверенным, что поддерживается адекватное соответствие.

3. Рецензирование – 180 минут

Ключевые слова

Аудит; неформальное рецензирование; инспекция; управленческое рецензирование; координатор; рецензирование; план рецензирования; рецензент; технический анализ; разбор.

Цели изучения Рецензирования

3.2 Управленческие рецензирования и аудиты

ТМ3.2.1 (К2) Понять основные особенности управленческих рецензирования и аудитов.

3.3 Управление рецензированием

ТМ-3.3.1 (К4) Проанализировать проект для выбора подходящего типа рецензирования и определить план проведения рецензирования для надлежащего выполнения, отслеживания и отчетности.

ТМ-3.3.2 (К2) Понимание необходимых особенностей, навыков и времени для участия в рецензировании.

3.4 Метрики рецензирования

ТМ-3.4.1 (К3) Определить метрики процесса и продукта, используемые при рецензировании.

3.5 Управление формальным рецензированием

ТМ-3.5.1 (К2) Объяснить, используя примеры, особенности формального рецензирования

3.1 Введение

В программе Базового уровня ISTQB рецензирования рассмотрены как активности по статическому тестированию продуктов. Аудиты и управленческие рецензирования сосредотачиваются прежде всего на процессе разработки программного обеспечения, но не на его рабочих продуктах.

Поскольку рецензирования — это вид статического тестирования, руководители тестирования могут быть ответственными за их полный успех применительно к объектам тестирования. Однако, в более широком контексте проектов разработки программного обеспечения, эта ответственность должна быть предметом организационной политики. Учитывая возможное применение формальных рецензирования во многих дисциплинах, как до, так и в рамках проектов программного обеспечения, ответственность может возлагаться на руководителя тестирования, менеджера по качеству или специально подготовленного координатора рецензирования. В настоящей программе ответственная сторона (кто бы это ни был) называется руководителем рецензирования.

Как определено в программе Базового уровня ISTQB, руководитель рецензирования должен убедиться, что существует окружение, которое подходит для успешной реализации рецензирования. Кроме того, руководитель рецензирования должен продумать план измерений, чтобы иметь возможность убедиться в эффективности рецензирования.

Поскольку тестировщики имеют глубокое понимание функционала и требуемых характеристик систем программного обеспечения, важно вовлечение тестировщиков в процесс рецензирования.

Участники рецензирования должны быть подготовлены и понимать соответствующие роли в любом процессе рецензирования. Все участники должны стремиться к успеху правильно организованного рецензирования.

Если все сделано правильно, рецензирование является максимальным и наиболее экономически эффективным вкладом в обеспечении качества. Таким образом, чрезвычайно важно, чтобы руководители рецензирования могли реализовать эффективные рецензирования в своих проектах и продемонстрировать преимущества данных рецензирования.

Возможные рецензирования в рамках проекта:

- Рецензирования контрактов инициируются в начале проекта и при достижении основных вех проекта.
- Рецензирования требований инициируются, когда требования доступны для рецензирования, идеально покрывающих функциональные и не функциональные требования.
- Рецензирования дизайна верхнего уровня инициируется, когда общее архитектурный дизайн доступен для рецензирования.
- Рецензирования детального проектирования инициируются, когда детальный проект доступен для рецензирования.
- Рецензирования кода инициируются, когда созданы конкретные модули, включая наряду с самим кодом модульные тесты и их результаты.
- Рецензирования рабочих продуктов тестирования, включая план(ы) тестирования, тестовые условия, результаты анализа рисков качества, тесты, тестовые данные, тестовые окружения и результаты тестов.
- Рецензирования входа тестов (готовности тестов) и выхода тестов для каждого уровня тестирования, которые соответственно проверяют критерии начала выполнения тестов и критерии завершения тестирования.
- Рецензирования приемки, которые используются для получения клиентом и заинтересованными лицами одобрения системы.

Помимо применения нескольких типов рецензирования продукта, руководителю рецензирования важно помнить, что поскольку рецензирования могут найти дефекты в статических документах, они должны быть дополнены другими формами статического тестирования (например, статическим анализом) и динамическим тестированием кода. Используя комбинацию этих методов, мы улучшаем тестовое покрытие и локализуем больше дефектов.

Различные методы имеют различные точки приложения. Например, рецензирование требований может устранить проблему на уровне требований прежде, чем проблема будет реализована в коде. Статический анализ может помочь обеспечить соблюдение стандартов программирования и проверить проблемы, которые могут быть слишком трудоемкими для команды, чтобы найти их анализом рабочих продуктов. Инспекции могут привести не только к обнаружению и устранению дефектов, но также могут показать авторам, как избегать дефектов в их рабочих продуктах.

В программе базового уровня ISTQB представлены следующие типы рецензирования:

- неформальное рецензирование;
- разбор;
- технический анализ;
- инспекции.

В дополнение к этим, руководитель тестирования может так же быть вовлечен в:

- управленческие рецензирования;
- аудиты.

3.2 Управленческие рецензирования и аудиты

Управленческие рецензирования используют для контроля прогресса, оценки статуса и принятия решения относительно будущих действий. Эти рецензирования поддерживают решения о будущем проекта, такие как потребность в ресурсах, осуществление корректирующих действий или изменение масштаба проекта.

Ключевые особенности управленческих рецензирования:

- Проводятся для менеджеров, имеющих непосредственное отношение к проекту или системе.
- Проводятся для заинтересованных лиц или лиц, принимающих решение, т.е. для топ-менеджера или директора.
- Проверяют соответствие плану или отклонение от плана.
- Проверяют адекватность процедур управления.
- Оценивают риски проекта.
- Оценивают влияния мер и способы измерения этих влияний.
- Создают список действий, проблем, которые будут устранены и решений, которые будут приняты.

Управленческие рецензирования процессов, такие как ретроспектива проекта (т.е. уроки и выводы) являются неотъемлемой частью совершенствования процессов.

Руководители тестирования должны участвовать, и могут инициировать управленческие рецензирования хода тестирования.

Аудиты обычно выполняются для демонстрации соответствия конкретному набору критериев, типично применимому стандарту, нормативным ограничениям или контрактным обязательствам. Также аудиты предназначены для обеспечения независимой оценки соблюдения процесса, нормативов, стандартов и т.д. Основные характеристики аудитов:

- Проводятся и координируются ведущим аудитором.
- Подтверждаются соответствия, собранные посредством интервью участников процессов и исследования документов.
- Документированы результаты, включая замечания, рекомендации, корректирующие действия и оценку успеха или не успеха.

3.3 Управление рецензированием

Рецензирования должны быть запланированы так, чтобы уложиться в естественный интервал или успеть к вехе проекта программного обеспечения. Как правило, рецензирования должны быть проведены после определения требований и дизайна, и проводиться, начиная с бизнес-целей вплоть до самого низкого-уровневого дизайна. Рецензирования должны перед достижением основных вех проекта, обычно при проверке деятельности до, во время и после выполнения тестов, а также других значимых этапов проекта.

Стратегия рецензирования должна быть согласована с политикой тестирования и в целом со стратегией тестирования.

Прежде чем формулировать общий план рецензирования на уровне проекта, руководитель рецензирования (им может быть и руководитель тестирования) должен учитывать:

- Что должно быть подвергнуто рецензированию (продукты и процессы).
- Кто будет вовлечен в конкретные рецензирования.
- Какие значимые факторы риска покрыты.

Ранее на стадии планирования проекта руководитель рецензирования должен определить, что должно быть подвергнуто рецензированию и выбрать подходящий тип (неформальное рецензирование, разбор, технический анализ или инспекция, или смесь двух или трех типов) и степень формализма. В этот момент можно рекомендовать провести обучение рецензированию. Здесь же может быть выделен бюджет (время и ресурсы) для рецензирования. Определение бюджета должно включать оценку рисков и вычисление возврата инвестиций.

Возврат инвестиций рецензирования — это разность между стоимостью проведения рецензирования и объемом расходов на исправление таких же дефектов на более поздней стадии (или пропущенных вообще), если рецензирования не было проведено.

Определение оптимального времени для выполнения рецензирования зависит от:

- Доступности объектов рецензирования в достаточно завершённом виде.
- Доступности нужного персонала для рецензирования.
- Времени, когда финальная версия объекта должна быть доступна.
- Времени, требуемого для рецензирования конкретного объекта.

Соответствующие метрики оценки рецензирования должны быть определены руководителем рецензирования во время планирования тестирования. Если используются инспекции, то краткие инспекции должны быть проведены по запросу автора, как только фрагмент документа (т.е. отдельные требования или разделы) закончен.

Цели процесса рецензирования должны быть определены во время планирования тестирования. Они включают проведение эффективных и результативных рецензирования и достижение согласия по решению относительно обратной связи рецензирования.

Проектное рецензирование часто проводится для всей системы, а также может быть необходимо для подсистем и даже отдельных компонентов программного обеспечения. Количество, тип, организация рецензирования и вовлеченный персонал зависят от размера проекта, его сложности и рисков продукта.

Для результативности участники рецензирования должны иметь соответствующий уровень знаний, как технический, так и процедурный. Тщательность и внимательность к деталям являются некоторыми из навыков рецензентов, необходимых для эффективного рецензирования. Ясность и правильная расстановка приоритетов — необходимые условия хороших комментариев рецензирования. Потребность в процедурных знаниях может означать, что необходимо некоторое обучение для уверенности, что рецензенты понимают свои роли и ответственности в процессе рецензирования.

При планировании рецензирования должны учитывать риски, связанные с техническими, организационными факторами и проблемами людей, которые выполняют рецензирования. Наличие

у рецензентов технических знаний в достаточной мере имеет решающее значение для успешного рецензирования. Все команды в рамках проекта должны быть вовлечены в планирование рецензирования, которые должны убедиться, что каждая команда стремится к успеху процесса рецензирования. Планирование должно убедиться, что каждая организация выделит рецензентам достаточное время для подготовки и участия в рецензировании в соответствующие моменты графика проекта. Также должно быть запланировано время для необходимого технического или процессного обучения рецензентов. Должны быть определены дублиеры рецензентов для случаев, когда ключевые рецензенты недоступны в связи с изменением личных или бизнес-планов.

Во время фактического выполнения формального рецензирования руководитель рецензирования должен убедиться, что:

- Соответствующие измерения доступны участникам рецензирования, что позволяет проводить оценку эффективности рецензирования.
- Созданы и поддерживаются чек-листы для улучшения будущих рецензирования.
- Для проблем, найденных во время рецензирования, критичность и приоритетов определена в рамках процесса управления дефектами (см. Глава 4).

После каждого рецензирования руководитель рецензирования должен:

- Собрать метрики рецензирования и убедиться, что выявленные проблемы полностью решены для достижения конкретных целей тестирования для рецензирования.
- Использовать метрики рецензирования в качестве входных данных при определении возврата инвестиций рецензирования (ROI).
- Предоставить информацию обратной связи соответствующим заинтересованным лицам.
- Обеспечить обратную связь участникам рецензирования.

Чтобы оценить эффективность рецензирования, руководитель тестирования может сравнивать фактические результаты, полученные в последующем тестировании (т.е. после рецензирования), с результатами из отчетов рецензирования. В случае, когда в проверенном рабочем продукте, утвержденном на основе результатов рецензирования, позже найдены дефекты, руководитель рецензирования должен рассмотреть действия, из-за которых в процессе рецензирования эти дефекты не были обнаружены. Вероятные причины - проблемы с процессом рецензирования (т.е. плохие критерии входа\выхода), неправильный состав команды рецензирования, неподходящие инструменты рецензирования (чек-листы и т.д.), недостаточный опыт и подготовка рецензентов, слишком мало времени на подготовку и проведение рецензирования.

Паттерн пропущенных дефектов (особенно серьезных), повторяющийся на нескольких проектах, показывает, что с проведением рецензирования есть серьезные проблемы. В таких ситуациях руководителю рецензирования необходимо пересмотреть процесс и принять соответствующие меры. Возможно также, по разным причинам, что рецензирование со временем потеряло эффективность. Уменьшение эффективности обнаружения дефектов рецензированием будет раскрыто во время ретроспективы проекта. И снова руководитель рецензирования должен исследовать и зафиксировать причины. В любом случае метрики рецензирования не должны использоваться, чтобы наказать или поощрить конкретных рецензентов или авторов, но должны сосредоточиться на самом процессе рецензирования.

3.4 Метрики рецензирования

Руководитель рецензирования (им, как говорилось выше, может быть руководитель тестирования) должен убедиться, что метрики пригодны для:

- Оценки качества рассмотренного объекта.
- Оценки стоимости проведения рецензирования.
- Оценки последующей выгоды от проведения рецензирования.

Руководитель рецензирования может использовать измерения для определения возврата инвестиций и эффективности рецензирования. Эти метрики могут быть использованы также для отчета и деятельности по совершенствованию процесса.

Для каждого рассмотренного рабочего продукта следующие метрики могут быть измерены и использованы для оценки продукта:

- размер рабочего продукта (страниц, строк кода и др.);
- время для подготовки (перед рецензированием);
- время проведения рецензирования;
- время исправления найденных дефектов;
- продолжительность процесса рецензирования;
- количество найденных дефектов и их серьезность;
- определение групп дефектов в рамках рабочего продукта (т.е. области, которые имеют более высокую плотность дефектов);
- тип рецензирования (неформальное рецензирование, разбор, технический анализ или инспекция);
- средняя плотность дефектов (например, количество дефектов на страницу или тысячу строк кода);
- предполагаемые остаточные дефекты (или остаточная плотность дефектов).

Для каждого рецензирования следующие метрики могут быть изменены и использованы для оценки процесса:

- эффективность обнаружения дефектов (с учетом дефектов, найденных позже);
- трудозатраты и время на совершенствование процесса рецензирования;
- процент покрытия запланированных рабочих продуктов;
- тип найденных дефектов и их серьезность;
- опрос участников об эффективности и результативности процесса рецензирования;
- стоимость метрик качества для дефектов рецензирования по сравнению с дефектами динамического тестирования и производственных дефектов;
- зависимость эффективности рецензирования (сравнение эффективности обнаружения дефектов по типам рецензирования);
- количество рецензентов;
- найденных дефектов за единицу времени (час);
- ожидаемое сэкономленное время проекта;
- средние трудозатраты на дефект (т.е. полное время обнаружения и фиксации дефектов, деленное на число дефектов).

Кроме того, упомянутые выше метрики для оценки продукта также полезны при оценке процесса.

3.5 Управление формальным рецензированием

В программе Базового уровня ISTQB описываются различные фазы формального рецензирования: планирование, старт, индивидуальная подготовка, встреча экспертов, повторная обработка и отслеживание. Чтобы правильно осуществить формальное рецензирование, руководитель

рецензирования должен убедиться, что все шаги в процессе рецензирования были выполнены полностью и правильно.

Формальное рецензирование имеет такие характеристики, как:

- Определение входных и выходных критериев.
- Чек-листы, которые будут использоваться рецензентами.
- Конечные результаты, такие как отчеты, оценочные листы и итоговые листы рецензирования.
- Метрики для отчета эффективности, результативности и прогресса рецензирования.

До начала формального рецензирования, выполнение предусловий рецензирования (как определено в процедуре или в списке входных критериев) должно быть подтверждено руководителем рецензирования.

Если предварительные условия для формального рецензирования не выполнены, то руководитель рецензирования может для принятия окончательного решения предложить одно из следующих действий:

- Переопределение рецензирования с новыми задачами.
- Корректирующие действия, необходимые для продолжения рецензирования.
- Отсрочка рецензирования.

Как часть управления формальным рецензированием, эти рецензирования контролируют в контексте полной (высокоуровневой) программы и связаны с действиями по обеспечению качества проекта. Контроль формальных обзоров включает информацию об обратной связи, используя метрики процесса и продукта.

4. Управление дефектами – 150 минут

Ключевые слова

Аномалия; дефект; группа управления дефектами; отказ; ложноотрицательный результат; ложноположительный результат; фазовая локализация; приоритет; основная причина; серьезность.

Цели обучения для управления дефектами

4.2 Жизненный цикл дефекта и жизненный цикл разработки программного обеспечения

ТМ-4.2.1 (К3) Разработать процесс управления дефектами для организации тестирования, включая жизненный цикл отчета о дефекте, который может использоваться для мониторинга и контроля за дефектами на проекте на протяжении всего жизненного цикла тестирования.

ТМ-4.2.2 (К2) Разъяснить описание процесса и требования к участникам, необходимые для эффективного управления дефектами.

4.3 Информация об отчетах об ошибках

ТМ-4.3.1 (К3) Определить данные и классификацию, которые должны быть собраны в процессе управления дефектами.

4.4 Оценка возможностей процесса с помощью информации о дефектах

ТМ-4.4.1 (К2) Объяснить, каким образом статистика дефектов может быть использована для оценки зрелости процессов разработки и тестирования программного обеспечения.

4.1 Введение

Процесс управления дефектами в организации и инструмент, используемый для управления дефектами, имеют решающее значение не только для команды тестирования, но для всех команд, участвующих в разработке программного обеспечения. Информация, собранная с помощью эффективного управления дефектами, позволяет руководителю тестирования и другим участникам проекта получить представление о состоянии проекта на протяжении всего жизненного цикла разработки, а также путем сбора и анализа данных может помочь определить местонахождение областей потенциального улучшения процессов тестирования и разработки.

В дополнение к пониманию общего жизненного цикла дефекта и того, как он используется для контроля и управления процессами разработки и тестирования программного обеспечения, руководитель тестирования также должен понимать, какую информацию важно собирать, и должен быть сторонником правильного использования как процесса, так и выбранного инструмента управления дефектами.

4.2 Жизненный цикл дефекта и жизненный цикл разработки программного обеспечения

Как было объяснено в программе подготовки Базового уровня, дефекты вносятся, когда человек делает ошибку в процессе создания рабочего продукта. Этим рабочим продуктом может быть спецификация требований, пользовательская история, технический документ, тестовый сценарий, программный код, или любой другой рабочий продукт, полученный в ходе разработки программного обеспечения или процесса технического обслуживания.

Дефекты могут быть найдены в любой точке жизненного цикла разработки программного обеспечения и в любом связанном программном обеспечении.

Поэтому каждая фаза жизненного цикла разработки программного обеспечения должна включать в себя мероприятия по выявлению и устранению потенциальных дефектов. Например, статические методы тестирования (например, рецензирование и статический анализ) могут быть использованы для проектных спецификаций, спецификации требований, а также кода перед доставкой этих рабочих продуктов для последующей деятельности. Чем раньше каждый дефект обнаружен и устранен, тем меньше общая стоимость качества системы; стоимость качества для определенного уровня дефектов сводится к минимуму, когда каждый дефект устраняется в той же фазе, в которой он был обнаружен (например, когда процесс разработки добивается идеальной фазовой локализации). Кроме того, как описано в программе подготовки Базового уровня, статическое тестирование непосредственно обнаруживает дефекты, а не находит сбои, и, таким образом, стоимость устранения дефекта ниже, так как не требуется отладка, чтобы изолировать дефект.

В ходе динамического тестирования, например, модульного, интеграционного и системного тестирования, наличие дефекта обнаруживается, когда он вызывает сбой, что приводит к расхождению между фактическими и ожидаемыми результатами тестов (например, аномалия). В некоторых случаях, ложноотрицательный результат появляется, когда тестировщик не наблюдает аномалию. Если тестировщик действительно наблюдает аномалию, то произошла ситуация, которая требует дальнейшего изучения. Это исследование начинается с заполнения отчета о дефекте.

В разработке на основе тестов автоматизированные модульные тесты используются в качестве исполняемых проектных спецификаций. По мере разработки кода он сразу же выполняется с помощью этих тестов. До тех пор, пока разработка модуля не завершена, некоторые или все тесты не будут проходить. Поэтому провал таких тестов не является дефектом, и, как правило, не отслеживается.

4.2.1 Жизненный цикл дефекта и его состояния

Большинство организаций используют инструмент для управления отчетами о дефектах на протяжении всего жизненного цикла. Отчет о дефекте, как правило, проходит через последовательность состояний, и это происходит в течение всего жизненного цикла дефекта. В большинстве этих состояний один участник несет ответственность за выполнение задачи, которая после завершения повлияет на перевод дефекта в следующее состояние (и назначение на следующего ответственного). В терминальных состояниях, например, когда отчет о дефекте закрыт (как правило, это означает, что рассматриваемый дефект исправлен и исправление проверено с помощью тестов подтверждения), отменен (как правило, это означает, что отчет о дефекте является неверным), не воспроизводится (как правило, это означает, что аномалии больше нельзя наблюдать), или отложен (как правило, это означает, что аномалия связана с реальным дефектом, но этот дефект не будет исправлен в ходе проекта), отчет не имеет ответственного, потому что никаких дальнейших действий не требуется.

Для дефектов, обнаруженных тестировщиками во время тестирования, есть три состояния, где действие находится на стороне команды тестирования:

- Начальное состояние:
 - В этом состоянии один или несколько тестировщиков собирают необходимую информацию для ответственного, который будет воспроизводить аномалию, чтобы разрешить дефект (см. Раздел 4.3 для получения более подробной информации, которая должна быть включена в отчет о дефектах).
 - Это также называется "открытым" или "новым" состоянием.

- Возвращенное состояние:
 - В этом состоянии получатель отчета отклонил отчет или просит тестировщика предоставить дополнительную информацию. Это состояние может указывать на дефицит в начальном процессе сбора информации или самого тестирования, и руководители тестирования должны отслеживать чрезмерное количество возвращений. Тестировщик(и) должны предоставить дополнительную информацию или подтвердить, что отчет действительно должен быть отклонен.
 - Состояние может также называться как "отклонено" или "на выяснении".
- Состояние «на подтверждении»:
 - В этом состоянии тестировщик будет проходить тест на подтверждение (часто выполняя шаги, чтобы воспроизвести ошибку из самого отчета о дефекте), чтобы определить, действительно ли исправление решило эту проблему. Если тест подтверждения указывает на то, что дефект исправлен, тестировщик должен закрыть отчет. Если тест подтверждения указывает на то, что дефект не исправлен, тестировщик должен повторно открыть отчет, назначая его предыдущему владельцу, который затем может завершить работу, необходимую для исправления дефекта.
 - Состояние может также называться как "исправленный" или "на проверке".

4.2.2 Управление отчетами о неверных дефектах или дубликатах

В некоторых случаях аномалия возникает не как симптом дефекта, а скорее из-за проблемы в тестовой среде, с тестовыми данными, каким-то другим элементом тестового обеспечения или из-за собственного непонимания тестировщика. Если тестировщик создает отчет о дефекте, и впоследствии при тестировании выясняется, что он не относится к дефекту в рабочем продукте, то это есть ложноположительный результат. Такие отчеты, как правило, отменяют или закрывают как неверные отчеты о дефектах. Кроме того, в некоторых случаях дефект может иметь различные симптомы, которые могут показаться тестировщику(ам) как совершенно не связанные. Если созданы два или более отчетов о дефектах, которые впоследствии оказываются относящимися к одной и той же основной причине, один из отчетов, как правило, сохраняется, а остальные закрываются в качестве дублирующих отчетов о дефектах.

В то время как отчеты о неверных дефектах и дубликатах говорят об определенном уровне неэффективности, какое-то количество таких отчетов неизбежно и должно быть принято как данность руководителем тестирования. Когда менеджеры пытаются устранить все отчеты о неверных дефектах и дубликатах, количество ложноотрицательных дефектов обычно увеличивается, поскольку у тестировщиков отбивают охоту заполнять отчеты о дефектах. Это снижает эффективность обнаружения дефектов, которая связана с одной из ключевых целей организации тестирования в большинстве случаев.

4.2.3 Кросс-функциональное управление дефектами

В то время как организация тестирования и руководитель тестирования, как правило, отвечают за общий процесс управления дефектами и инструмент управления дефектами, кросс-функциональная команда, как правило, отвечает за управление заведенными дефектами для данного проекта. Вместе с руководителем тестирования участники комитета по управлению дефектами (или группы управления дефектами), как правило, состоят из представителей групп разработки, управления проектом, управления продуктом и других заинтересованных лиц, которые имеют отношение к программному обеспечению в стадии разработки.

Когда аномалии обнаружены и занесены в инструмент управления дефектами, комитет по управлению дефектами должен встретиться, чтобы определить, является ли каждый отчет о дефекте действительным, и должен ли он быть исправлен или отложен. Это решение требует

рассмотрения комитетом по управлению дефектами, чтобы понять преимущества, риски и затраты, связанные с исправлением или не исправлением дефекта. Если дефект должен быть исправлен, команда должна установить приоритет исправления дефекта относительно других задач проекта. Руководитель тестирования и команда по тестированию могут консультироваться относительно важности дефекта и должны предоставить имеющуюся в наличии объективную информацию.

Инструмент отслеживания дефектов не должен использоваться в качестве замены личного общения и заседания комитета по управлению дефектами не должны быть использованы вместо эффективного использования инструмента отслеживания дефектов. Коммуникация, адекватная поддержка инструмента, четко определенный жизненный цикл дефекта, и вовлеченный комитет по управлению дефектами являются необходимыми для эффективного и рационального управления дефектами.

4.3 Информация, содержащаяся в отчете о дефекте

Когда дефект обнаружен (как часть статического тестирования) или наблюдается сбой (как часть динамического тестирования), информация должна быть собрана и включена в отчет о дефекте.

Эта информация должна удовлетворять трем целям:

- 1) Управление отчетом на протяжении всего жизненного цикла дефекта.
- 2) Оценка состояния проекта, особенно с точки зрения качества продукции, а также оценка прогресса тестирования.
- 3) Оценка возможностей процесса (как описано в Разделе 4.4 ниже).

Информация, необходимая для управления отчетами о дефектах и состоянием проекта может различаться в зависимости от того, когда дефект обнаружен в жизненном цикле, как правило, с меньшим количеством информации, необходимой ранее (например, рецензирование требований и модульное тестирование). Тем не менее, основная собранная информация должна быть постоянной на всем протяжении жизненного цикла, а в идеале на всех проектах для осмысленного сравнения данных о дефектах на протяжении всего проекта и по всем проектам.

Сбор информации о дефектах может помочь в мониторинге прогресса тестирования, контроля и оценки критериев завершения тестирования.

Например, информация о дефектах должна поддерживать анализ плотности дефектов, анализ тенденций дефектов (обнаруженных и исправленных), среднее время от обнаружения до исправления дефекта, и интенсивность отказов (например, анализ наработки на отказ).

Собранная информация о дефектах может включать в себя следующее:

- имя сотрудника, нашедшего дефект;
- роль сотрудника, нашедшего дефект (например, конечный пользователь, бизнес-аналитик, разработчик, служба технической поддержки);
- тип тестирования (например, тестирование практической, тестирование производительности, регрессионное тестирование);
- краткое описание проблемы (как правило, заголовок описания дефекта);
- детальное описание проблемы;
- действия по воспроизведению сбоя (для дефекта), наряду с фактическими и ожидаемыми результатами (подчеркивая различия), включая скриншоты, дампы базы данных, и логи, где это возможно;

- фаза жизненного цикла дефекта - внедрение, обнаружение и устранение, включая уровень тестирования, если это возможно;
- рабочий продукт, в котором был найден дефект;
- серьезность воздействия на систему и/или на заинтересованных лиц продукта (как правило, определяется техническим поведением системы);
- приоритет решения этой проблемы (как правило, определяется влиянием сбоя на бизнес);
- подсистема или компонент, в котором находится дефект (для кластерного анализа дефектов);
- проектная активность, во время которой проблема была обнаружена;
- метод идентификации, который выявил проблему (например, рецензирование, статический анализ, динамическое тестирование, использование продукта);
- тип дефекта (обычно соответствует используемой классификации дефектов);
- характеристика качества, пострадавшая от дефекта;
- тестовая среда, в которой наблюдался дефект (для динамического тестирования);
- проект и продукт, в котором эта проблема существует;
- текущий владелец; т.е. лицо, назначенное работать над проблемой, предполагая, что отчет не в конечном состоянии;
- текущее состояние отчета (как правило, управляется с помощью инструмента отслеживания дефектов как часть жизненного цикла);
- конкретные рабочие продукты (например, тестовые элементы и их номера версий), в которых наблюдалась проблема, наряду с конкретными рабочими продуктами, в которых эта проблема была решена в конечном счете;
- воздействие на проект и интересы заинтересованных лиц;
- выводы, рекомендации и разрешения на предпринятые или на не предпринятые действия для решения этой проблемы;
- риски, затраты, возможности и выгоды, связанные с исправлением или нет данного дефекта;
- даты, в которые дефект менял различные состояния в течение жизненного цикла, владельцы отчета по каждому состоянию, а также меры, принятые членами проектной команды, чтобы изолировать, исправить и проверить исправление дефекта;
- описание того, как дефект был исправлен в конечном счете и рекомендации для проверки исправления (если дефект был исправлен за счет изменения в программном обеспечении);
- другие ссылки, такие как тест, который обнаружил дефект, и риск, требование или другой элемент базиса тестирования, связанный с дефектом (для динамического тестирования).

Различные стандарты и документы, такие как ISO 9126 [ISO9126] (был заменен на ISO 25000), IEEE 829 [IEEE829], IEEE 1044 [IEEE1044], и Ортогональная Классификация Дефектов, используются, чтобы помочь руководителю тестирования определить, какую информацию собрать для отчетности по дефектам.

Любая конкретная информация, которая определена как необходимая для отчетов по дефектам, должна быть полной, краткой, точной, объективной, актуальной и своевременной. Даже когда ручное вмешательство и личные коммуникации преодолевают проблемы с данными отчета с точки зрения решения отдельного дефекта, проблемы с данными отчета могут создавать непреодолимые препятствия для правильной оценки состояния проекта, прогресса тестирования, и возможностей процесса.

4.4 Оценка возможности процесса с помощью отчетов о дефектах

Как уже говорилось в Главе 2, отчеты о дефектах могут быть полезны для мониторинга состояния проекта и составления отчетов. Хотя процесс влияния метрик, в первую очередь рассматривается

в программе подготовки руководителей тестирования экспертного уровня, руководители тестирования должны знать о том, что отчеты означают с точки зрения оценки возможностей процессов разработки и тестирования программного обеспечения.

В дополнение к мониторингу хода тестирования, упомянутому в Главе 2 и в Разделе 4.3, информация о дефектах должна поддерживать инициативы по совершенствованию процессов.

Примеры включают в себя:

- Использование фазы внедрения, обнаружения и удаления информации, на основе перехода фаз, для оценки фазовой локализации и предложить пути повышения эффективности обнаружения дефектов в каждой фазе.
- Использование фазы входной информации для анализа фаз методом Парето, в которых представлено наибольшее количество дефектов, чтобы делать направленные улучшения для уменьшения общего числа дефектов.
- Использование информации о корневых причинах дефектов для определения основных причин появления дефектов, чтобы усовершенствовать процессы, которые уменьшают общее число дефектов.
- Использование информации о фазах внедрения, обнаружения и устранения для подсчета стоимости анализа качества, чтобы свести к минимуму затраты, связанные с дефектами.
- Использование информации о дефектных компонентах для проведения кластерного анализа дефектов, чтобы лучше понять технические риски (для тестирования на основе рисков), а также для того, чтобы выполнить реинжиниринг проблемных компонентов.

Использование показателей для оценки эффективности и рентабельности процесса тестирования обсуждается в программе подготовки руководителей тестирования экспертного уровня.

В некоторых случаях команды могут принять решение не отслеживать дефекты, обнаруженные в течение некоторого этапа или всего жизненного цикла разработки программного обеспечения. В то время как это часто делается во имя эффективности и ради сокращения накладных расходов процесса, в действительности, это значительно снижает прозрачность процесса тестирования и разработки программного обеспечения. Это делает предложенные выше улучшения трудно осуществимыми из-за отсутствия достоверных данных.

5. Улучшение процесса тестирования – 135 минут

Ключевые слова

Интегрированная модель зрелости процессов программного обеспечения (СММl); Критические процессы тестирования (СТР); Процесс систематического тестирования и оценки (STEP); Интегрированная модель зрелости тестирования (ТММi); Модель непрерывного улучшения процессов тестирования (ТPI Next).

Цели обучения для улучшения процесса тестирования

5.2 Улучшение процесса тестирования

ТМ-5.2.1 (К2) Объяснить, используя примеры, почему важно улучшать процесс тестирования.

5.3 Улучшение процесса тестирования

ТМ-5.3.1 (К3) Определить план улучшения процесса тестирования с помощью модели IDEAL (Модель организации совершенствования).

5.4 Улучшение процесса тестирования с помощью ТММi

ТМ-5.4.1 (К2) Резюмировать опыт, ограничения и цели модели улучшения процессов тестирования ТММi.

5.5 Улучшение процесса тестирования с помощью ТPI Next

ТМ-5.5.1 (К2) Резюмировать опыт, ограничения и цели модели улучшения процессов тестирования ТPI Next.

5.6 Улучшение процесса тестирования с СТР

ТМ-5.6.1 (К2) Резюмировать опыт, ограничения и цели модели улучшения процессов тестирования СТР.

5.7 Улучшение процесса тестирования с STEP

ТМ-5.7.1 ТМ-5.6.1 (К2) Резюмировать опыт, ограничения и цели модели улучшения процессов тестирования STEP.

5.1 Введение

После того, как процесс тестирования в организации в целом установлен, он должен постоянно совершенствоваться. В этой главе сначала рассматриваются общие вопросы совершенствования процессов, а затем приводится введение в некоторые конкретные модели, которые могут быть использованы для улучшения процесса тестирования. Руководители тестирования должны понимать, что они будут движущей силой изменений и улучшений процесса тестирования и поэтому они должны быть знакомы с промышленными методами, обсуждаемыми в этой главе. Более подробную информацию о процессах улучшения тестирования можно найти в учебном плане экспертного уровня.

5.2 Процесс улучшения тестирования

Подобно тому, как организации используют тестирование с целью улучшения программного обеспечения, методы совершенствования процессов могут быть выбраны и использованы для улучшения процесса разработки программного обеспечения и конечного результата. Улучшение процесса также может быть применено к процессам тестирования. Доступны различные способы и методы улучшения тестирования программного обеспечения и систем, содержащих программное обеспечение. Эти методы направлены на совершенствование процесса, а, следовательно, и результатов, путем предоставления руководящих принципов и областей для улучшения.

Тестирование часто приходится на большую часть от общей стоимости проекта. Тем не менее, лишь ограниченное внимание уделяется процессу тестирования в различных моделях улучшения процесса разработки ПО, таких как CMMI® (см. ниже).

Были разработаны модели улучшения тестирования, такие как TMMi®, STEP, STP и TPI Next®, которые повышают внимание к тестированию в большинстве моделей улучшения процесса разработки ПО. При правильном использовании, эти модели могут предложить набор кросс-организационных метрик, которые могут быть использованы для эталонных сравнений.

Модели, представленные в этой программе, не являются рекомендацией к использованию, но представлены здесь, чтобы обеспечить репрезентативное представление о том, как работают модели и что они включают в себя.

5.2.1 Введение в улучшение процесса

Совершенствования процесса имеют отношение к процессу разработки программного обеспечения, так же, как и к процессу тестирования. Извлечение уроков из собственных ошибок позволяет улучшить процесс разработки и тестирования программного обеспечения. Цикл улучшения Деминга: Планируй, Выполни, Проверь, Действуй (Plan – Do – Check – Act, PDCA) был использован в течение многих десятилетий, и по-прежнему актуален, когда тестировщикам необходимо улучшить процесс сегодня.

Одной предпосылкой улучшения процесса является убеждение, что качество системы во многом зависит от качества процесса, используемого для разработки программного обеспечения. Повышение качества в индустрии программного обеспечения снижает потребность в ресурсах для поддержания программного обеспечения и, таким образом, обеспечивает больше времени для создания новых и более эффективных решений в будущем. Модели процессов обеспечивают старт улучшений, путем измерения возможностей организации с помощью модели. Модели также обеспечивают базу для совершенствования процессов в организации, основанные на результатах оценки.

Оценка процесса ведет к определению возможностей процесса, которая способствует улучшению процесса. Это может вызвать последующую оценку процесса для измерения эффекта от улучшения.

5.2.2 Типы улучшений процесса

Использование моделей оценки является распространенным методом, который обеспечивает стандартизированный подход к совершенствованию процессов тестирования с использованием проверенных и надежных методов.

Модели совершенствования процессов подразделяются на два типа:

1. Эталонная модель процесса обеспечивает измерение зрелости как часть оценки возможности организации по сравнению с моделью, чтобы оценить организацию и понять пути улучшения процесса.
2. Эталонная модель содержимого, которая обеспечивает бизнес-ориентированную оценку возможностей организации для улучшения, в некоторых случаях, сравнительного анализа со средними значениями по отрасли с использованием объективных измерений. Эта оценка может быть использована для создания пути улучшения процесса.

Улучшение процесса тестирования также может быть достигнуто без моделей, а с использованием, например, аналитических подходов и ретроспективных встреч.

5.3 Улучшение процесса тестирования

ИТ-индустрия может работать с моделями улучшения процесса тестирования, чтобы достичь более высокого уровня зрелости и профессионализма. Промышленные стандарты помогают разрабатывать кросс-организационные метрики и меры, которые могут быть использованы для сравнения. Из необходимости улучшения процессов в промышленности появилось несколько рекомендованных наборов процессов. К ним относятся STEP, CMMI, TPI Next и CTP. Ступенчатые модели, такие как TMMi и CMMI, обеспечивают стандарты для сравнения между различными компаниями и организациями. Непрерывные модели, такие как CTP, STEP и TPI Next, позволяют организации решать свои наиболее приоритетные вопросы с большей свободой в порядке реализации. Каждая из них обсуждается далее в этом разделе.

Все эти модели позволяют определить организации, где она находится с точки зрения текущих процессов тестирования.

После того, как оценка выполнена, TMMi и TPI Next предлагают дорожную карту для улучшения процесса тестирования. В качестве альтернативы STEP и CTP обеспечивают организацию средствами определения, где может быть получен наибольший возврат инвестиций от улучшения процесса, и оставляет возможность организации выбрать соответствующую дорожную карту.

После того, как было решено, что процессы тестирования должны быть пересмотрены и улучшены, этапы внедрения улучшения процессов, которые будут приняты для этой деятельности, могут быть определены как в модели IDEALSM [IDEAL96]:

- Инициирование процесса улучшения
- Диагностирование текущей ситуации
- Создание плана улучшения процессов тестирования
- Действия для внедрения улучшений
- Извлечение уроков из программы улучшения

Инициирование процесса улучшения

До того, как началась деятельность по улучшению процесса, цели, задачи, масштаб и охват улучшений процесса должны быть согласованы заинтересованными сторонами. Также в это время происходит выбор модели улучшения процесса. Модель может быть выбрана из общедоступных (например, CTP, STEP, TMMi и TPI Next) или разработана своими силами. Кроме того, критерии успеха должны быть определены, и способ, с помощью которого они будут измеряться во время деятельности по улучшению.

Диагностирование текущей ситуации

Осуществляется согласованный подход к оценке и создается отчет об оценке тестирования, который содержит оценку текущей практики тестирования и список возможных улучшений процесса.

Создание плана улучшения процессов тестирования

Перечень возможных улучшений процесса является приоритетным. Приоритетность может зависеть от возврата от инвестиций, рисков, согласованности со стратегией организации, и/или измеримых количественных или качественных преимуществ. Установив порядок приоритетов, план доставки улучшений разработан.

Действия для внедрения улучшений

Внедряется план улучшения процесса тестирования. Это может включать в себя любое необходимое обучение или наставничество, пилотирование процессов и в конечном счете их полное развертывание.

Извлечение уроков из программы улучшения

После полного внедрения улучшений процесса, важно проверить, какие выгоды (из тех, которые были установлены ранее в дополнение к потенциально неожиданным выгодам) были получены. Важно также проверить, какие из критериев успеха для деятельности по улучшению процессов были выполнены.

В зависимости от используемой модели процесса, это этап процесса, в котором начинается мониторинг следующего уровня зрелости и принимается решение либо начать процесс улучшения снова, или прекратить деятельность.

5.4 Улучшение процесса тестирования с TMMi

TMMi состоит из пяти уровней зрелости и призвана дополнить CMMI. Каждый из уровней зрелости содержит определенные области процесса, которые должны быть завершены на 85% путем достижения конкретных и общих целей, прежде чем организация может перейти к следующему уровню.

Уровнями зрелости TMMi являются:

Уровень 1: Начальный

Начальный уровень представляет собой состояние, в котором нет официально задокументированного или структурированного процесса тестирования. Тесты обычно разрабатываются в каждом отдельном случае после кодирования и тестирование рассматривается как отладка. Под целью тестирования понимают доказательство того, что программное обеспечение работает.

Уровень 2: Управляемый

Второй уровень достигается, когда процесс тестирования отделен от отладки. Это может быть достигнуто путем установки политики и целей тестирования, вводя шаги, найденные в фундаментальном процессе тестирования (например, планирование тестирования), а также внедрение основных методов и техник тестирования.

Уровень 3: Стандартизованный

Третий уровень достигается, когда процесс тестирования интегрируется в жизненный цикл разработки программного обеспечения, и задокументирован в формальных стандартах, процедурах и методах. Проводится рецензирование и выделен отдел тестирования, который контролируется и управляется.

Уровень 4: Измеряемый

Четвертый уровень достигается, когда процесс тестирования эффективно измеряется и управляется на организационном уровне в интересах конкретных проектов.

Уровень 5: Улучшаемый

Конечный уровень представляет собой состояние зрелости процесса тестирования, когда данные из процесса тестирования могут быть использованы для предотвращения дефектов и акцент делается на оптимизацию установленного процесса.

Для получения дополнительной информации о TMMi (см. [vanVeenendaal11]) и [www.tmmi.org].

5.5 Улучшение процесса тестирования с TPI Next

TPI Next определяет 16 ключевых областей, каждая из которых охватывает определенный аспект процесса тестирования, такие как стратегия тестирования, метрики, инструменты тестирования и тестовое окружение.

Четыре уровня зрелости определены в модели:

- начальный
- управляемый
- эффективный
- улучшаемый

Определяются конкретные контрольные точки для оценки каждой ключевой области на каждом из уровней зрелости. Результаты обобщаются и визуализируются с помощью матрицы зрелости, которая охватывает все ключевые области. Определение целей улучшения и их реализации могут быть адаптированы в соответствии с потребностями и возможностями организации.

Общий подход делает TPI Next независимым от любой модели улучшения процесса разработки ПО. Он охватывает как инженерные аспекты тестирования, так и поддержку принятия управленческих решений (см. [DeVries 09]).

Для получения дополнительной информации о TPI Next см [www.tpinext.com].

5.6 Улучшение процесса тестирования с СТР

Основная предпосылка модели оценки критических процессов тестирования (СТР) является то, что некоторые процессы тестирования имеют решающее значение. Если эти критические процессы выполняются хорошо, то они будут поддерживать успешные команды тестирования. И наоборот, если эти активности проводятся плохо, даже талантливые тестировщики и руководители тестирования вряд ли будут успешными. Модель определяет двенадцать критических процессов тестирования. СТР в первую очередь эталонная модель содержимого.

Модель СТР является контекстно-зависимым подходом, который позволяет настраивать модель, в том числе:

- Определить конкретные изменения.
- Обозначить атрибуты хороших процессов.
- Выбрать порядок и важность выполнения улучшений процесса.

Модель СТР адаптируется в контексте всех моделей жизненного цикла разработки программного обеспечения.

Помимо интервью участников, модель СТР включает в себя использование метрик для сравнения организаций со средними показателями и лучшими практиками по отрасли.

Для получения дополнительной информации о СТР (см. [Black03]).

5.7 Улучшение процесса тестирования со STEP

STEP (Процесс систематического тестирования и оценки), как и СТР, в отличие от TMMi и TPI Next, не требует, чтобы улучшения проводились в определенном порядке.

STEP является прежде всего эталонной моделью содержимого, основанной на идее, что тестирование - это активность на протяжении всего жизненного цикла, которая начинается во время формулирования требований и продолжается пока идут работы с системой. Методология STEP делает акцент на парадигме "сначала тест - затем код", используя стратегию тестирования на основе требований, чтобы гарантировать, что раннее создание тестовых примеров подтверждает спецификации по требованиям до проектирования и кодирования.

Основные постулаты методологии:

- Стратегия тестирования на основе требований.
- Тестирование начинается в начале жизненного цикла.
- Тесты используются в качестве требований и моделей использования.
- Тестовый дизайн определяет проектирование программного обеспечения.
- Раннее обнаружение дефектов или предотвращение их в целом.
- Дефекты систематически анализируются.
- Тестировщики и разработчики работают вместе.

В некоторых случаях модель оценки STEP пересекается с моделью TPI Next.

Для получения дополнительной информации о STEP (см. [Craig02]).

6. Инструменты для тестирования и автоматизация – 135 минут

Ключевые слова

Инструменты с открытым исходным кодом; инструменты, сделанные на заказ.

Цели изучения тестовых инструментов и автоматизации

6.2 Выбор инструмента

ТМ-6.2.1 (K2) Описать задачи руководителя при выборе инструмента с открытым исходным кодом.

ТМ-6.2.2 (K2) Описать задачи руководителя при выборе инструмента, сделанного на заказ.

ТМ-6.2.3 (K4) Оценить текущую ситуацию для того, чтобы разработать план выбора инструмента, включая риски, стоимость и прибыль.

6.3 Жизненный цикл инструмента

ТМ-6.3.1 (K2) Описать различные этапы жизненного цикла инструмента.

6.4 Метрики инструмента

ТМ-6.4.1 (K2) Описать, как сбор метрик и их оценка может быть улучшена с помощью инструмента.

6.1 Введение

Эта глава расширяет программу Базового Уровня, описывая общие концепции, которые должен учитывать руководитель тестирования касательно инструментов и автоматизации.

6.2 Выбор инструмента

Существует достаточно много проблем, которые руководитель тестирования должен учитывать при выборе инструментов тестирования.

Исторически, чаще всего, выбор останавливают на покупке готового инструмента. В большинстве случаев это единственный подходящий вариант. Однако, есть и другие возможности, такие как инструменты, сделанные на заказ.

Независимо от типа инструмента руководитель тестирования обязан оценить общую стоимость владения инструментом на протяжении всего ожидаемого жизненного цикла. Для этого он должен провести анализ затрат и прибыли. Эта тема описана ниже в разделе возврата инвестиций (ROI).

6.2.1 Инструменты с открытым исходным кодом

Инструменты с открытым исходным кодом доступны почти для всех активностей тестирования, начиная с управления тестовыми сценариями и управления дефектами и заканчивая автоматизацией тестовых сценариев. Важное отличие инструментов с открытым исходным кодом – это несмотря на то, что они имеют достаточно низкую исходную стоимость, для некоторых из них может не предоставляться официальная поддержка. Однако многие инструменты с открытым исходным кодом поддерживаются энтузиастами, которые готовы предоставлять нетрадиционную или неофициальную поддержку пользователей.

Кроме того, многие инструменты с открытым исходным кодом создавались для решения конкретной задачи или единственной проблемы, таким образом, они не обладают всей функциональностью, какая может быть у похожего коммерческого продукта. Из-за этого перед выбором инструмента должен быть проведен подробный анализ потребностей тестовой команды и требований к искомому инструменту.

Одним из основных преимуществ использования инструмента с открытым исходным кодом является возможность его изменения или расширения пользователями. Если тестовые инженеры обладают ключевыми навыками, они могут настроить один инструмент для работы с другими или адаптировать его работу под нужды команды. Объединяя несколько инструментов, можно решать задачи, неподвластные коммерческим продуктам. Конечно, чем больше инструментов используется и адаптируется, тем больше сложностей и дополнительной работы. Руководитель тестирования должен быть уверен, что команда не начинает использовать инструмент ради инструмента; как и с другими инструментами, усилия всегда должны быть нацелены на достижение возврата инвестиций (ROI).

Руководитель тестирования должен изучить и понять лицензионную схему выбранного инструмента. Многие инструменты с открытым исходным кодом поставляются с различными свободно поставляемыми лицензиями (GNU GeneralPublicLicense), которая предписывает распространение программы под такими же условиями, под которыми она была получена. В случае если тестовая команда делает изменения в инструменте для поддержки своего тестового процесса, она должна предоставить эти изменения для всех внешних пользователей согласно лицензии. Руководитель должен проверить и предусмотреть все правовые последствия от использования программы его организацией.

Компания, разрабатывающая ПО с особыми требованиями к безопасности или критически важной функциональностью, может столкнуться с трудностями при использовании инструментов с открытым исходным кодом. Несмотря на то, что такие инструменты могут быть очень хорошего качества, их точность, скорее всего, не подтверждена (не сертифицирована). Готовые продукты чаще всего сертифицированы как по точности, так и по соответствию конкретной задаче (например, DO-178B). Если инструмент с открытым исходным кодом достаточно хорош, то его сертификация может стать обязанностью команды, использующей его, и потребовать дополнительных усилий.

6.2.2 Инструменты, сделанные на заказ

Тестовая организация может столкнуться с ситуацией, когда ее требованиям не удовлетворяет ни один готовый продукт или инструмент с открытым исходным кодом. Причиной этого может быть собственная аппаратная платформа, адаптированное оборудование или процесс, измененный необычным способом. В этом случае, если команда обладает необходимыми навыками, руководитель тестирования может принять решение о разработке заказного инструмента.

Плюсы в создании заказного инструмента в том, что он может абсолютно удовлетворить потребности команды и эффективно функционировать в контексте требований команды. Инструмент может быть создан для взаимодействия с другими используемыми программами и для создания данных в том формате, в котором требуется команде. Кроме того, такой инструмент может применяться в дальнейшем в других проектах. Однако перед тем как планировать использование такого инструмента в других проектах, важно проверить его соответствие целям и задачам, а также оценить пользу и возможные издержки.

Перед тем как решить создавать новый инструмент, руководитель тестирования должен также оценить и возможные негативные последствия. Инструмент, сделанный на заказ, зависит от разработчика, который его пишет. Таким образом, весь код должен быть хорошо задокументирован, чтобы его могли поддерживать и другие. Если этого не сделать, то в случае ухода разработчика,

инструмент может “осиротеть” и стать неприменимым из-за отсутствия должной поддержки. Со временем такой инструмент может потребовать расширения функциональности, что может повлиять на качество продукта, приводя к ложноположительным отчетам об ошибках или к некорректным данным. Руководитель тестирования должен помнить, что такой инструмент – это тоже программный продукт, который также может содержать ошибки. Инструмент, создаваемый на заказ, должен быть спроектирован и протестирован, чтобы убедиться в правильности его функционирования.

6.2.3 Возврат инвестиций (ROI)

Руководитель тестирования отвечает за то, что все используемые инструменты приносят пользу работе команды и показывают положительную рентабельность для компании. Для того чтобы быть уверенным в реальной и долгосрочной прибыли, перед заказом или покупкой инструмента, должен быть проведен анализ отношения затрат к прибыли. В этом анализе необходимо учесть как регулярные, так и нерегулярные затраты, некоторые из которых являются монетарными, а некоторые ресурсными или временными затратами. Учесть надо и риски, которые могут привести к уменьшению ценности инструмента.

Нерегулярные затраты могут включать:

- описание требований к инструменту для удовлетворения целей и задач;
- оценка и выбор подходящего инструмента и его поставщика;
- покупка, адаптация или разработка инструмента;
- проведение вводного тренинга по инструменту;
- интеграция инструмента с другими инструментами;
- получение аппаратного и программного обеспечения, необходимого для работы инструмента.

Повторяющиеся затраты включают в себя:

- Владение инструментом:
 - лицензия и стоимость поддержки;
 - затраты на техническое обслуживание самого инструмента;
 - поддержание артефактов, созданных с помощью инструмента;
 - расходы на обучение и наставничество.
- Перенос и настройка инструмента на различном оборудовании.
- Адаптация инструмента для будущих нужд.
- Улучшение качества и процессов для оптимального использования выбранного инструмента.

Руководитель тестирования должен также учесть расходы, возникающие при стечении обстоятельств, связанных с инструментом. Время, потраченное на получение, управление, обучение и использование инструмента могло бы быть потрачено на активности тестирования; таким образом, до того момента как инструмент станет активно применяться, необходимо выделять больше тестовых ресурсов.

Есть много рисков, связанных с использованием инструмента; не все инструменты обеспечивают выгоду, покрывающую риски. Риски, связанные с инструментом, обсуждались в программе Базового уровня. Вот список рисков, относящихся к возврату инвестиций:

- Незрелость организации (неготовность использовать инструмент).

- Артефакты, созданные инструментом, могут быть сложны в поддержке и требовать множественной перепроверки при смене объекта тестирования.
- Уменьшение вовлеченности тестового аналитика в процесс тестирования может привести к уменьшению значимости тестирования (например, эффективность обнаружения дефектов может быть низкой при использовании только автоматических скриптов).

И, наконец, руководитель тестирования должен учитывать накапливаемую выгоду от использования инструмента. Польза от введения инструмента в использование может включать:

- Уменьшение повторяющейся работы.
- Уменьшение времени тестового цикла (например, при использовании автоматического регрессионного тестирования).
- Уменьшение затрат на выполнение тестов.
- Увеличение доли определенного типа тестирования (например, регрессионного).
- Уменьшение ошибок, связанных с человеческим фактором. Например:
 - сгенерированные специальным инструментом данные могут быть более эффективны;
 - сравнение, проведенное с помощью инструмента, может быть более точным;
 - ввод данных может быть более точным при использовании скриптового инструмента.
- Уменьшение усилий на получение информации о тестах. Например:
 - генерация инструментом отчетов и метрик;
 - повторное использование тестовых артефактов, таких как тестовые сценарии, автоматизированные тестовые сценарии или тестовые данные.
- Увеличение объема тестирования в областях, которые не могут быть эффективно протестированы без автоматизации (например, тестирование производительности, нагрузочное тестирование).
- Повышение статуса тестировщика, работающего над автоматизацией, и организации в целом путем демонстрации знания и использования усложненных инструментов.

Редко какая организация пользуется только одним инструментом; общая рентабельность инвестиций – это объединение прибыли от всех используемых инструментов. Между инструментами необходимо обеспечение обмена информацией и согласование их работ. Рекомендуется применять всестороннюю долгосрочную стратегию использования инструментов.

6.2.4 Процесс выбора

Инструмент для тестирования – долгосрочная инвестиция с возможным расширением на протяжении итераций в рамках одного проекта или с возможностью применения в разных проектах. Руководитель тестирования должен рассматривать перспективу использования инструмента с различных точек зрения:

- Для бизнеса необходим возврат инвестиций (ROI). Для получения большей прибыли компания должна убедиться, что инструменты, которые должны взаимодействовать между собой – как тестовые, так и не тестовые – должны работать вместе. Иногда связь между инструментами может быть налажена в процессе работы с ними, но на это может потребоваться время.
- Для проекта инструмент должен быть эффективным (например, позволять избегать ошибок при ручном тестировании, таких как опечатки при вводе данных). Инструменту необходимо ощутимое время перед тем, как он начнет приносить прибыль. Чаще всего возврат инвестиций можно получить во втором релизе продукта, или во время поддержки, а не в исходном проекте, когда автоматизация была впервые применена. Руководитель тестирования должен учитывать полный цикл выпускаемого продукта.

- Для инженеров, использующих инструмент, важно, чтобы инструмент работал эффективно и целесообразно. Для оценки, насколько быстро инженер овладеет инструментом при минимальном стрессе, должны использоваться данные об эффективности обучения. Будучи внедренным в процесс, инструмент требует тренингов и помощи инженерам, его использующим.

Чтобы быть уверенным, что всё учтено для внедрения инструмента, очень важно создать дорожную карту.

Процесс выбора инструмента для тестирования уже обсуждался в программе Базового уровня и состоял в следующем:

- оценить зрелость (готовность) организации;
- определить требования к инструменту;
- оценить различные инструменты;
- оценить поставщиков или службы поддержки (для инструментов с открытым исходным кодом или инструментов на заказ);
- определить внутренние требования для тренингов и поддержки инженеров при использовании инструментов;
- предусмотреть тренинги, требуемые для повышения навыков автоматизации;
- оценить прибыль относительно затрат (как обсуждалось в Разделе 6.2.3).

Для каждого инструмента, независимо от того на какой стадии тестирования он будет использоваться, руководитель тестирования должен оценить следующие возможности:

- Анализ:
 - «Поймет» ли инструмент входные данные?
 - Подходит ли инструмент для намеченной цели?
- Проектирование:
 - Поможет ли инструмент при проектировании тестов, базируясь на имеющейся информации (например, инструмент для разработки тестовых сценариев, базируясь на требованиях)?
 - Может ли тест-дизайн сгенерироваться автоматически?
 - Может ли тестовое окружение полностью или частично быть сгенерировано в поддерживаемом и удобном формате?
 - Могут ли автоматически быть сгенерированы тестовые данные (например, генерация данных на базе анализа кода)?
- Выбор данных и тестовых сценариев:
 - Как инструмент выбирает данные для использования (например, какой тест с какими данными выполнять)?
 - Может ли инструмент применять ручной или автоматический критерий выбора?
 - Может ли инструмент определять, как выбрать выходные результаты на основе входных данных?
 - Может ли инструмент определить, какие тесты необходимы, базируясь на критериях покрытия (например, если даны требования, может ли инструмент анализировать покрытие требований для определения, какие тесты необходимо выполнять)?
- Выполнение:

- Будет ли инструмент выполнять тесты автоматически или понадобится ручное вмешательство?
- Как инструмент останавливается и повторно запускается?
- Должен ли инструмент уметь «слушать» связанные с процессом события (например, должен ли инструмент управления тестами автоматически обновлять результат теста, если заведенный на него дефект был закрыт)?
- Оценка:
 - Как инструмент определяет правильность полученных результатов (например, будет ли инструмент использовать тестовый предсказатель для определения правильности результатов)?
 - Какими возможностями обладает инструмент для восстановления после возникших ошибок?
 - Предоставляет ли инструмент адекватное протоколирование и отчеты?

6.3 Жизненный цикл инструмента тестирования

Руководитель тестирования должен управлять четырьмя различными стадиями жизненного цикла инструмента:

1. Приобретение. Инструмент приобретается по плану, описанному выше (см. Раздел 6.2). После принятия решения о внедрении инструмента руководитель тестирования назначает (чаще всего это тест-аналитик или технический тест-аналитик) администратора инструмента. Этот сотрудник принимает решения, как и когда использовать инструмент, где хранить создаваемые артефакты, соглашения о наименованиях, и т.д. Решение таких вопросов заранее, не оставляя их «на потом», может повлиять на ожидаемый возврат инвестиций (ROI).
2. Поддержка и сопровождение. Потребуется непрерывный процесс поддержки и сопровождения инструмента. Ответственность за сопровождение может быть возложена на администратора инструмента или назначена на специальную группу, отвечающую за инструменты. Если инструмент должен работать с другими инструментами, то необходимо учесть процесс обмен данными и взаимодействия. Нужно также учитывать необходимость резервного копирования и восстановления инструмента и выходных данных.
3. Развитие. Должна учитываться необходимость изменений в инструменте. Так как время идет, потребности бизнеса или проблемы поставщика инструмента могут потребовать существенных изменений в используемом инструменте. Например, поставщик инструмента может внести изменения в инструмент, которые приведут к проблемам взаимодействия инструментов. Изменения в тестовом окружении, связанные с требованиями бизнеса, могут также привести к проблемам с инструментом. Чем более сложное окружение используется инструментом, тем более эволюционное изменение может нарушить его работу. С этой точки зрения, в зависимости от роли инструмента в процессе тестирования, руководитель тестирования должен быть уверен, что в организации есть возможность обеспечения постоянной поддержки.
4. Окончание работы. Настанет время, когда инструмент перестанет быть полезным. В этот момент от него нужно будет изящно избавиться. Необходимо будет заменить функциональность, обеспечиваемую инструментом и архивировать, и сохранить данные. Это может случиться, когда инструмент закончил свой жизненный цикл или

просто достиг момента, когда польза от него и возможности преобразования в новый инструмент не превышают затраты и риски.

На протяжении всего жизненного цикла инструмента руководитель тестирования ответственен за функционирование и корректную поддержку услуг, предоставляемых инструментом тестовой команде.

6.4 Метрики инструмента тестирования

Руководитель тестирования вместе с техническим тест-аналитиком и тест-аналитиком должны планировать и собирать объективные метрики. Различные инструменты могут сохранять ценные данные во время выполнения тестов, что поможет уменьшить усилия, затрачиваемые на сбор данных. Эти данные помогут руководителю тестирования управлять общими тестовыми ресурсами.

Различные инструменты нацелены на сбор различных типов данных. Вот некоторые для примера:

- Инструменты управления тестами могут поддерживать множество различных метрик. Метрики покрытия могут быть получены благодаря связи требований к тестам и автоматических скриптов. В любой момент времени можно получить статус выполненных, планируемых и выполняемых тестов (пройден, не пройден, заблокирован, в очереди).
- Система управления дефектами может предоставлять ценную информацию о дефектах, включая текущий статус, критичность, приоритет, распределение по системе, и т.д. Некоторые инструменты могут указать стадию разработки, на которой дефект был принесен или найден, количество пропущенных дефектов, и т.д. Это помогает руководителю тестирования управлять усовершенствованием процесса.
- Инструмент статического анализа могут помочь найти и сообщить о дефектах сопровождения.
- Инструменты тестирования производительности могут предоставить ценную информацию о масштабируемости системы.
- Инструменты оценки покрытия могут помочь руководителю тестирования понять, какая часть системы уже проверена тестами.

Во время выбора инструмента необходимо выработать требования к отчетности. Затем во время настройки инструмента необходимо правильно внедрить эти требования, чтобы информация, поставляемая инструментами, была понятна всем заинтересованным лицам.

7. Командные навыки – Формирование команды – 210 минут

Ключевые слова

Независимость тестирования.

Цели изучения Командные навыки – Формирования команды

7.2 Личные навыки

TM-7.2.1 (K4) Используя лист оценки навыков, проанализировать сильные и слабые стороны членов команды, относящиеся к использованию программных систем, знанию предметной и бизнес-области, областей разработки систем, тестирования ПО и межличностных навыков.

TM-7.2.2 (K4) Проанализировать полученные оценки навыков с точки зрения команды для определения плана обучения и развития навыков.

7.3 Развитие команды тестирования

TM-7.3.1 (K2) Для заданной ситуации, обсудить необходимые обязательные и желательные навыки для руководства командой тестирования.

7.4 Внедрение тестирования в организации

TM-7.4.1 (K2) Объяснить варианты независимого тестирования.

7.5 Мотивация

TM-7.5.1 (K2) Привести примеры факторов, мотивирующих и демотивирующих тестировщиков.

7.6 Коммуникации

TM-7.6.1 (K2) Объяснить факторы, влияющие на эффективность коммуникаций как внутри команды тестирования, а также команды тестирования с заинтересованными сторонами.

7.1 Введение

Успешные руководители тестирования подбирают, нанимают и поддерживают команды с подходящим набором навыков. Требования к навыкам могут меняться с течением времени, поэтому кроме найма нужных людей для сохранения команды и поддержания ее высокой производительности необходимо, прежде всего, обеспечить адекватное обучение и развитие возможностей команды. Кроме обычных навыков команды, руководитель тестирования также должен поддерживать набор навыков, позволяющих эффективно работать в условиях жесткого давления и быстро меняющейся обстановки.

В этой главе рассматривается, как оценивать навыки, как заполнить пробелы при создании синергетической команды, сплоченной изнутри и эффективной в организации, а также как мотивировать команду и эффективно общаться.

7.2 Личные навыки

Умение конкретного сотрудника тестировать ПО может быть получено с помощью опыта или обучения. Перечисленные ниже пункты могут внести свой вклад в базу знаний тестировщика:

- использование программных систем;
- знание предметной или бизнес-области;
- участие в различных фазах процесса разработки ПО, включая анализ, разработку и техническую поддержку;
- участие в работах по тестированию ПО.

Конечные пользователи программных систем имеют хорошее представление о том, что делает система, сбои в каких областях имеют наибольшее влияние и как система должна реагировать в различных ситуациях. Пользователи с опытом в предметной области знают наиболее важные области для бизнеса, в том числе их критичность для соответствия требованиям. Эти знания могут быть использованы для приоритизации действий тестирования, создания реалистичных тестовых данных и сценариев, а также для верификации или создания сценариев использования.

Знание процесса разработки ПО (анализ требований, архитектура, проектирование и кодирование) дает понимание того, как могут появляться ошибки, где они могут быть найдены и как предотвратить их появление. Опыт технической поддержки дает знания о пользовательском опыте, ожиданиях и требованиях к удобству использования. Опыт в разработке ПО важен для использования инструментов тестирования, требующих экспертизы в программировании и проектировании, участия в анализе кода, рецензировании кода, компонентном тестировании, а также интеграционном тестировании, ориентированном на технические моменты.

Специфические навыки в тестировании ПО включают умения, рассматриваемые в программах обучения Базового уровня и Продвинутого уровня в модулях тест-аналитик и технический тест-аналитик. Например, умение анализировать спецификацию, участвовать в анализе рисков, проектировать тестовые сценарии, а также внимательность при выполнении тестов и фиксации результатов.

Для руководителей тестирования особенно важны знания, навыки и опыт в управлении проектами, поскольку руководство тестированием включает множество активностей из управления проектами, например, планирование, мониторинг и контроль, отчетность перед заинтересованными сторонами. В отсутствие руководителя проекта руководитель тестирования может взять на себя эту роль, совместив руководство тестированием и проектом, например, на завершающих стадиях проекта. Эти навыки дополняют те умения, которые рассматриваются в программе обучения Базового уровня и данной программе обучения.

В дополнение к техническим навыкам в тестировании важны межличностные навыки, такие как подача и восприятие критики, влияние и взаимодействие. Технически компетентные тестировщики, скорее всего, потерпят неудачу, если заранее не овладеют или не приобретут необходимые межличностные навыки. Наряду с эффективным взаимодействием, успешный профессионал в тестировании должен быть также хорошо организованным, внимателен к деталям и владеть сильными навыками письменного и устного общения.

Идеальная команда тестирования должна иметь требуемый набор уровней навыков и опыта, а члены команды - желание и возможность обучать коллег и учиться у них. В разных условиях одни навыки будут более важными и необходимыми, чем другие. Например, в техническом окружении тестирования, где необходимы навыки тестирования интерфейса прикладного программирования и навыки программирования, технические навыки могут быть более востребованными, чем знание

предметной области. В условиях тестирования методом черного ящика может быть более востребован опыт в предметной области. Важно помнить, что окружение и проекты меняются.

При создании листов оценки навыков руководитель тестирования должен перечислить все навыки, важные для работы и подходящие для должности. Как только эти навыки перечислены, каждый член команды может быть оценен по балльной системе (например, от 1 до 5, где 5 – высший ожидаемый уровень навыка для данной области). Члены команды могут оцениваться для определения их сильных и слабых сторон и на основе этой оценки, могут быть разработаны индивидуальные или групповые планы обучения. Руководитель тестирования может задать целевое значение показателя для конкретного сотрудника, чтобы улучшить его навыки в определенных областях, а также определить критерии, используемые для оценки навыков сотрудника.

Люди должны приниматься на работу на длительное время, а не одноразово для отдельного проекта. Когда руководитель тестирования вкладывается в развитие тестировщиков и создает условия для постоянного обучения, члены команды будут мотивированы на развитие своих навыков и знаний, чтобы быть готовыми к следующим задачам.

Руководитель тестирования редко имеет возможность нанять идеальных членов команды. И даже если члены команды идеальны для конкретного проекта, они могут не быть идеальными для следующих проектов. Руководитель тестирования должен нанимать умных, любознательных, гибких, желающих работать эффективно в команде, а также способных обучаться людей. Несмотря на то, что такой набор идеальных людей практически невозможен, сильная команда может быть сформирована посредством баланса сильных и слабых сторон отдельных личностей.

Используя листы оценки навыков, руководитель тестирования может определить сильные и слабые стороны его команды. На основе этой информации формируется основа плана обучения и развития навыков. Прежде всего, надо решить, что делать со слабыми сторонами, имеющими наибольшее влияние на эффективность и результативность команды. Одним из подходов является обучение - послать сотрудников на курс обучения, организовать курс обучения в компании, разработать собственный тренинг, использовать курсы дистанционного обучения. Другим подходом может быть самостоятельное обучение - книги, вебинары, интернет-ресурсы. Еще одним подходом может быть перекрестное обучение - поручить кому-то, имеющему интерес к определенному навыку, поработать над задачей, требующей этого навыка с кем-то, кто его уже имеет, тем самым переняв навык; иметь собственных экспертов, делающих небольшие презентации в области их компетенции и т.п. Наставничество – схожий подход, когда новый специалист работает в паре со старшим коллегой, уже имеющим опыт в данной роли и выступающим в роли советника и помощника. В дополнение к работе со слабыми сторонами руководитель тестирования должен помнить об использовании выявленных сильных сторон как части плана обучения и развития навыков. Дополнительную информацию по планам развития команд тестирования (см. в [McKay07]).

7.3 Развитие команды тестирования

Для построения наилучшей команды подбор персонала является одной из наиболее важных функций руководящей роли в организации. Необходимо учитывать множество факторов в дополнение к специфичным личным навыкам. При выборе нового члена команды необходимо учитывать направление развития команды. Впишутся ли навыки и характер новичка в уже существующую команду тестирования? Важно помнить о преимуществе наличия в команде различных по характеру людей с набором различных технических навыков. Сильная команда тестирования способна работать с несколькими проектами различной сложности, одновременно поддерживая межличностные взаимоотношения с членами других проектов.

Тестирование часто проводится в жестких условиях. Графики разработки ПО зачастую сжатые или даже нереалистичные. Заинтересованные стороны имеют большие (иногда - неоправданно завышенные) ожидания от команды тестирования. Руководитель тестирования должен нанимать людей, которые справятся с такими ситуациями, умеющих не допускать срыва сроков и концентрироваться на работе, даже если график выглядит нереальным. Руководитель тестирования должен решать проблемы, связанные с графиком и ожиданиями, но он также должен понимать, что все эти факторы отражаются на членах команды. При подборе команды важно учитывать окружение и соответствие характера человека этому окружению. В условиях, когда времени на работу недостаточно, руководитель тестирования должен искать людей, которые, завершив свою задачу, спрашивают, что им следует делать дальше.

Сотрудники будут работать лучше и относиться к порученному делу более ответственно, если они почувствуют себя оцененными и востребованными. Каждый из них должен понимать, что он является важным членом команды, а их вклад жизненно необходим для успеха команды в целом. Если это понимание присутствует, перекрестное обучение будет происходить само собой, балансировка загрузки будет выполняться членами команды самостоятельно, а руководитель тестирования сможет сосредоточиться на внешних задачах.

Необходимо как можно быстрее включать новых членов в команду и обеспечивать их необходимым наставничеством и поддержкой. Каждому сотруднику должна быть назначена в команде конкретная роль на основе индивидуальной оценки. Необходимо, чтобы каждый сотрудник добивался личного успеха, одновременно вкладываясь в общий успех команды. Во многом это достигается за счет установления соответствия личных врожденных качеств и командных ролей, а также развития навыков самих сотрудников.

Объективная оценка навыков может быть полезной при принятии решения о найме или пополнении команды тестирования. Оценка проводится с помощью интервьюирования, выполнения кандидатом тестовых заданий, рецензирования результатов работы и отзывов. Навыки, которые необходимо оценивать, включают:

Технические навыки могут быть продемонстрированы:

- Разработкой тестовых сценариев на основе документированных требований.
- Рецензированием технической документации (требований, кода и т.п.).
- Написанием ясных, понятных и объективных комментариев при рецензировании.
- Применение различных техник тестирования в различных ситуациях (например, использованием таблиц альтернатив при тестировании набора бизнес-правил).
- Обнаружением и аккуратным документированием сбоев.
- Демонстрацией понимания классификации дефектов.
- Демонстрацией понимания причин дефектов.
- Использованием инструмента для тестирования конкретного API, включая позитивные и негативные тесты.
- Использованием SQL для поиска и изменения информации в базе данных для тестирования определенного сценария.
- Разработкой окружения автоматизированного тестирования, позволяющего выполнять тесты и анализировать их результаты.
- Выполнением автоматизированных тестов и решением проблем.
- Написанием планов и спецификаций тестирования.
- Написанием сводных отчетов о тестировании, включающих оценку результатов тестирования.

Межличностные навыки могут быть продемонстрированы:

- Представлением информации о проекте тестирования, который отстает от плана.
- Разъяснением отчетов о дефектах разработчикам, считающих, что дефекта нет.
- Обучением коллег.
- Представлением проблемы, связанной с неэффективным процессом, руководству.
- Рецензированием тестовых сценариев, разработанных коллегами, и представлением им комментариев.
- Интервьюированием коллег.
- Совместной работой с разработчиками.

Хотя это неполный перечень, а требуемые навыки могут варьироваться в различных контекстах и организациях, навыки из этого списка необходимы почти всегда. При создании вопросов для интервью и демонстрации навыков, руководитель тестирования может оценить квалификацию кандидата и определить его сильные и слабые стороны. Создав хороший набор оценок, необходимо его применить также и к уже работающим сотрудникам, чтобы определить области их роста и обучения.

В дополнение к навыкам, которые необходимы отдельным членам команды, руководитель тестирования также должен владеть отличными навыками общения и дипломатии. Руководитель тестирования должен уметь разрешать конфликтные ситуации, владеть подходящими инструментами коммуникации, а также уметь сосредоточиться на создании и поддержании хороших отношений в организации.

7.4 Внедрение тестирования в организации

Внедрить тестирование в организационную структуру можно различными способами. Хорошо известно, что в рамках жизненного цикла разработки программного обеспечения за качество отвечает каждый. При этом существенный вклад в создание качественного продукта может внести независимая команда тестирования. На практике понятие независимости функций тестирования сильно различается. Это видно на примере списка функций, упорядоченного по возрастанию степени независимости тестирования:

- Нет независимых тестировщиков
 - В этом случае независимость тестирования отсутствует, и код тестируется тем разработчиком, который его написал.
 - Если разработчику выделяют время для такого тестирования, то он проверит, что код работает, как задумано, однако это может не соответствовать актуальным требованиям.
 - Разработчик может быстро устранить любые обнаруженные дефекты.
- Тестирование выполняется другим разработчиком – не тем, кто написал код
 - Независимость между разработчиком и тестировщиком минимальная
 - Разработчик, тестирующий код другого разработчика, может не быть мотивирован сообщать о дефектах.
 - Разработчик, тестирующий код другого разработчика, как правило, ограничивается позитивными тестовыми сценариями.
- Тестирование выполняется тестировщиком или командой тестировщиков, являющейся частью команды разработчиков

- Тестировщик или команда тестировщиков может отчитываться перед руководством проекта или руководителем разработки.
 - Менталитет тестировщика больше ориентирован на проверку соответствия требованиям, чем менталитет разработчика.
 - Поскольку тестировщик - член команды разработчиков, он может в дополнение к тестированию заниматься и разработкой.
 - Руководитель тестирования может быть более сфокусирован на расписании совещаний, чем на достижении целей по качеству.
 - Тестированию может быть присвоен более низкий статус в команде, чем разработке.
 - Тестировщик может не иметь полномочий влиять на цели по качеству или на достижение этих целей.
- Тестирование проводится специалистами по тестированию из бизнеса, сообщества пользователей или другой организации, не занимающейся разработкой
 - Информация о результатах тестирования объективно предоставляется заинтересованным сторонам.
 - Такая команда в основном нацелена на качество.
 - Профессиональное развитие и обучение нацелены на тестирование.
 - Карьерный рост представляется в рамках тестирования.
 - У группы тестирования существует свое руководство, нацеленное на качество.
 - Внешние специалисты по тестированию выполняют конкретные типы тестирования
 - Применяется экспертиза конкретных типов тестирования, где обычного (функционального и регрессионного) тестирования, вероятно, недостаточно.
 - Конкретными типами тестирования могут быть тестирование удобства использования, безопасности, производительности или другие типы тестирования, где необходима специальная экспертиза.
 - Эти внешние специалисты должны быть нацелены на качество, но их мнение ограничено областями их экспертизы. Так, хорошо работающий продукт может не соответствовать всем функциональным требованиям, но это может остаться незамеченным экспертом по тестированию производительности.
 - Тестирование проводится внешней организацией.
 - Эта модель обеспечивает максимальную независимости тестирования от разработки.
 - Передача знаний тестировщикам может быть недостаточной для компетентного тестирования.
 - Потребуется четкие требования и четко определенная структура коммуникации.
 - Качество внешней организации должно регулярно проверяться.

Этот список основан на типичных подходах, но для конкретной организации он может быть неприменим. Ведь хорошо известно, что должность и звание не обязательно определяют интересы конкретного сотрудника. Руководители разработки могут быть ориентированы на качество и могут быть хорошими руководителями тестирования. Независимые руководители тестирования могут отчитываться перед руководством, нацеленным на соблюдение сроков, и поэтому могут быть нацелены на сроки более, чем на качество. Для оптимального позиционирования команды тестирования в организации руководитель тестирования должен понимать цели этой организации.

Существуют различные степени независимости между организациями, занимающимися разработкой и тестированием. Важно понимать, что независимость - не панацея: большая независимость может привести к большей изоляции и меньшей передаче знаний. Низкий уровень независимости может увеличить знания, но привести к противоречивым целям. Уровень независимости будет определяться также используемой моделью разработки программного обеспечения, например, при разработке в области гибких методологий тестировщики чаще всего входят в команду разработчиков.

Любой из рассмотренных вариантов может возникнуть в конкретной организации. Тестирование может проводиться как внутри организации, занимающейся разработкой, так и в независимой организации, занимающейся тестированием, а окончательный вердикт может быть вынесен внешней организацией. Важно понимать ответственность и ожидания для каждого этапа тестирования и использовать это понимание для получения наивысшего качества продукта при соблюдении плана-графика и бюджетных ограничений.

7.5 Мотивация

Тестировщиков можно мотивировать разными способами. Приведем несколько примеров:

- Публичное признание выполненной работы.
- Поощрение руководства.
- Уважение со стороны проектной команды и коллег.
- Повышение персональной ответственности и самостоятельности.
- Адекватное вознаграждение за проделанную работу (включая материальное поощрение, повышение ответственности, признание).

Бывают, однако, проектные особенности, которые могут затруднить применение этих мотивационных инструментов. Например, тестировщик может очень усердно работать над проектом, срок завершения которого нереальный. Тестировщик может сделать все возможное, чтобы сосредоточить внимание команды на качестве, потратить дополнительные часы и усилия, однако продукт все равно будет поставлен раньше, чем было запланировано, из-за внешних факторов. В результате, несмотря на все усилия тестировщика, поставляется некачественный продукт. Если вклад тестировщика не понят и не измерен, это демотивирует его вне зависимости от качества конечного продукта.

Команда тестирования должна знать, что она отслеживает показатели, позволяющие доказать, что была проделана хорошая работа по проведению тестирования, снижению рисков и фиксации результатов. Если эти данные не собраны и не опубликованы, команду легко демотивировать, поскольку они не получают того признания, которое является следствием хорошо выполненной ими работы.

Признание не ограничивается только нематериальными отношениями уважения и поощрения - оно может проявиться в карьерном росте, повышении ответственности и признании заслуг. Если группа тестирования не получает признания, эти возможности могут оказаться недоступными.

Признание и уважение возникают тогда, когда всем ясно, что тестировщик вносит вклад в повышение ценности проекта. В конкретном проекте для этого лучше всего привлечь тестировщика с начала проекта и в течение всего жизненного цикла. Со временем тестировщики получают признание других участников проекта благодаря их вкладу в проект. Этот вклад также должен быть выражен в терминах снижения затрат на обеспечение качества и сдерживания рисков.

Руководитель тестирования играет важную роль в мотивации команды тестирования, а также выступает в качестве лидера команды тестирования для крупной организации. Руководитель тестирования должен признавать достижения отдельных тестировщиков, а также давать справедливые и честные оценки их ошибок. Справедливый и квалифицированный руководитель тестирования будет мотивировать членов команды личным примером.

7.6 Коммуникации

Коммуникации с командой тестирования осуществляется, как правило, следующими способами:

- Документирование продуктов тестирования: стратегия тестирования, план тестирования, тестовые сценарии, сводные отчеты о тестировании, отчеты о дефектах и т. д.
- Результаты рецензирования документов: требований, функциональных спецификаций, сценарии использования системы, документация по тестированию компонентов и т. д.
- Сбор и распространение информации: взаимодействие с разработчиками, другими членами группы тестирования, руководством и т. д.

Коммуникации между тестировщиками и другими заинтересованными сторонами должны быть профессиональными, объективными и эффективными, чтобы создать и поддерживать уважение к команде тестирования. При предоставлении конструктивной обратной связи о рабочих продуктах других членов проектной команды необходимы дипломатичность и объективность. Кроме того, коммуникации должны быть направлены на достижение целей тестирования и улучшение качества - как продуктов, так и процессов, используемых для создания программных систем.

Руководители тестирования общаются с широкой аудиторией, включая пользователей, членов проектной команды, руководство, внешние группы тестирования и клиентов. Коммуникации должны быть эффективными для целевой аудитории. Например, отчет, подготовленный для команды разработчиков с указанием тенденции числа и серьезности обнаруженных дефектов, может быть слишком подробным, чтобы его можно было использовать для руководителей. При любой коммуникации, но особенно при презентациях, важно понимать, что сообщается, как оно воспринимается и понимается, и какую правильную среду надо создать для принятия сообщения. Поскольку руководитель тестирования часто может предоставлять информацию о состоянии проекта, важно, чтобы эта информация была на соответствующем уровне детализации (например, менеджеры обычно хотят видеть тенденции дефектов, а не отдельные дефекты) и была представлена так, чтобы ее было легко правильно понять (например, в виде простых диаграмм и красочных графиков). Эффективные коммуникации помогают удерживать внимание аудитории, передавая правильное сообщение. Каждая презентация должна рассматриваться руководителем тестирования как возможность повысить качество продуктов и процессов.

Руководитель тестирования общается с не только людьми вне команды тестирования (внешняя коммуникация). Важной частью работы руководителя тестирования является эффективная коммуникация внутри группы тестирования (внутренняя коммуникация) для передачи новостей, инструкций, изменений в приоритетах и другой стандартной информации, уместной в нормальном процессе тестирования. Руководитель тестирования также может общаться как с вышестоящими, так и с нижестоящими сотрудниками в организации. Независимо от направления сообщения применяются одни и те же правила - коммуникация должна быть подходящей для аудитории, сообщение должно быть отправлено эффективно и его понимание должно быть подтверждено.

Руководители тестирования должны уверенно владеть различными средствами коммуникации. Большая часть информации передается по электронной почте, в устной форме, на официальных или неофициальных заседаниях, в официальных или неофициальных отчетах и даже с использованием инструментов управления тестированием, таких как инструменты управления

дефектами. Все коммуникации должны быть профессиональными и объективными. Правка формы текста для повышения качества как формы, так содержания, является необходимым шагом даже при самом срочном общении. Письменные сообщения живут гораздо дольше реального проекта. Для руководителя тестирования важно подготовить документацию профессионального качества, представляющую организацию, которая повышает качество.

8. Ссылки

8.1 Стандарты

- [BS7925-2] BS 7925-2, Software Component Testing Standard.
Глава 2
- [FDA21] FDA Title 21 CFR Part 820, Food and Drug Administration, USA
Глава 2
- [IEEE829] IEEE Standard for Software and System Test Documentation
Главы 2 и 4
- [IEEE1028] IEEE Standard for Software Reviews and Audits
Глава 2
- [IEEE1044] IEEE Standard Classification for Software Anomalies
Глава 4
- [ISO9126] ISO/IEC 9126-1:2001, Software Engineering - Software Product Quality
Главы 2 и 4
- [ISO12207] ISO 12207, Systems and Software Engineering, Software Life Cycle Processes
Глава 2
- [ISO15504] ISO/IEC 15504, Information Technology - Process Assessment
Глава 2
- [ISO25000] ISO/IEC 25000:2005, Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE)
Главы 2 и 4
- [RTCA DO-178B/ED-12B]: Software Considerations in Airborne Systems and Equipment Certification, RTCA/EUROCAE ED12B.1992.
Глава 2

8.2 Документы ISTQB

- [ISTQB_AL_OVIEW] ISTQB Advanced Level Overview, Version 1.0
- [ISTQB_ALTM_SYL] ISTQB Advanced Level Test Manager Syllabus, Version 1.0
- [ISTQB_ALTTA_SYL] ISTQB Advanced Level Technical Test Analyst Syllabus, Version 1.0
- [ISTQB_ETM_SYL] ISTQB Expert Test Management Syllabus, Version 1.0
- [ISTQB_FL_SYL] ISTQB Foundation Level Syllabus, Version 2011
- [ISTQB_GLOSSARY] Standard glossary of terms used in Software Testing, Version 2.2, 2012
- [ISTQB_ITP_SYL] ISTQB Expert Improving the Test Process Syllabus, Version 1.0

8.3 Торговые знаки

В документе используются следующие зарегистрированные торговые знаки:

- CMMI®, IDEALSM, ISTQB®, ITIL®, PRINCE2®, TMMi®, TPI Next®
- CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.
- IDEAL is a service mark of Software Engineering Institute (SEI), Carnegie Mellon University
- ISTQB is a registered trademark of the International Software Testing Qualifications Board
- ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office
- PRINCE2 is a registered trademark of the Cabinet Office
- TMMi is a registered trademark of TMMi Foundation

- TPI Next is a registered trademark of Sogeti Nederland B.V.

8.4 Книги

- [Black03]: Rex Black, "Critical Testing Processes," Addison-Wesley, 2003, ISBN 0-201-74868-1
- [Black09]: Rex Black, "Managing the Testing Process, third edition," John Wiley & Sons, 2009, ISBN 0-471-22398-0
- [Black11]: Rex Black, Erik van Veenendaal, Dorothy Graham, "Foundations of Software Testing," Thomson Learning, 2011, ISBN 978-1-84480-355-2
- [Craig02]: Rick Craig, Stefan Jaskiel, "Systematic Software Testing," Artech House, 2002, ISBN 1-580-53508-9
- [Crispin09]: Lisa Crispin, Janet Gregory, "Agile Testing: A Practical Guide for Testers and Agile Teams", Addison-Wesley, 2009, ISBN 0-321-53446-8
- [de Vries09]: Gerrit de Vries, et al., "TPI Next", UTN Publishers, 2009, ISBN 90-72194-97-7
- [Goucher09]: Adam Goucher, Tim Riley (editors), "Beautiful Testing," O'Reilly, 2009, ISBN 978-0596159818
- [IDEAL96]: Bob McFeeley, "IDEAL: A User's Guide for Software Process Improvement," Software Engineering Institute (SEI), 1996, CMU/SEI-96-HB-001
- [Jones07]: Capers Jones, "Estimating Software Costs, second edition," McGraw-Hill, 2007, ISBN 978-0071483001
- [Jones11]: Capers Jones and Olivier Bonsignour, "Economics of Software Quality," Pearson, 2011, ISBN 978-0132582209
- [McKay07]: Judy McKay, "Managing the Test People," Rocky Nook, 2007, ISBN 978-1933952123
- [Musa04]: John Musa, "Software Reliability Engineering, second edition," Author House, 2004, ISBN 978-1418493882
- [Stamatis03]: D.H. Stamatis, "Failure Mode and Effect Analysis," ASQC Quality Press, 2003, ISBN 0-873-89300
- [vanVeenendaal11] Erik van Veenendaal, "The Little TMMi," UTN Publishers, 2011, ISBN 9-490-986038
- [vanVeenendaal12] Erik van Veenendaal, "Practical Risk-based Testing," UTN Publishers, 2012, ISBN 978-9490986070
- [Whittaker09]: James Whittaker, "Exploratory Testing," Addison-Wesley, 2009, ISBN 978-0321636416
- [Wiegers03]: Karl Wiegers, "Software Requirements 2," Microsoft Press, 2003, ISBN 978-0735618794

8.5 Прочие ссылки

Следующие ссылки указывают на информацию, доступную в интернете. Эти ссылки проверялись во время публикации настоящей программы обучения продвинутого уровня.

- <http://www.istqb.org>
- <http://www.sei.cmu.edu/cmml/>
- <http://www.tmmi.org/>
- <http://www.tpinext.com/>