Управление процессом тестирования

Что такое управление тестированием?

Важной частью качества программного обеспечения является процесс тестирования и проверки корректности его работы. *Управление тестированием* представляет собой мероприятия по организации и управлению процессом и артефактами, необходимыми для проведения тестирования. К традиционным инструментальным средствам, используемым для этого, являются:

- Бумага и карандаш
- Текстовые редакторы
- Электронные таблицы

Более масштабное тестирование может использовать доморощенные программные решения, обычно построенные на основе электронных таблиц или баз данных, либо коммерческие приложения, такие как IBM® Rational® ClearQuest® Test Manager или Mercury TestDirector.

Общая цель управления тестированием должна *позволять группам разработчиков планировать, разрабатывать, выполнять и оценивать всю деятельность по тестированию в общем процессе разработки программного обеспечения*. Сюда относятся мероприятия по координации всех действий, вовлеченных в процесс тестирования, отслеживание зависимостей и взаимоотношений между тестовыми активами (test assets) и, самое важное, определение, измерение и отслеживание показателей качества.

Аспекты управления тестированием

Управление тестированием может быть разбито на различные фазы: организация, планирование, авторинг, выполнение и составление отчетности. Ниже эти фазы рассматриваются более детально.

Организация тестовых артефактов и ресурсов является, несомненно, необходимой частью управления тестированием. Она требует организации и управления запасами элементов для тестирования вместе с различными вещами, используемыми для выполнения тестирования. Это формирует способ отслеживания группой разработчиков зависимостей и взаимодействия между тестовыми активами. Самыми широко используемыми тестовыми активами, которые нуждаются в управлении, являются:

- Тестовые сценарии
- Тестовые данные
- Тестовое программное обеспечение
- Тестовое аппаратное обеспечение

Планирование тестирования - это полный набор задач, отвечающих на вопросы зачем, что, когда и где тестировать. Причина создания того или иного теста называется мотиватором теста (test motivator) (например, необходимость проверки специального требования). То, что должно быть протестировано, разбивается на много тестовых примеров (test cases) для проекта. Для ответа на вопрос где тестировать, определяются и документируются необходимые конфигурации программного и аппаратного

обеспечения. Вопрос, когда тестировать, решается путем отслеживания итераций (или циклов, или временных интервалов) тестирования.
Авторинг тестирования представляет собой процесс определения конкретных шагов, необходимых для завершения данного теста. Этот процесс решает вопрос, как что-то будет тестироваться. Именно здесь некие абстрактные тестовые примеры развиваются в более подробные шаги тестирования, которые, в свою очередь, станут тестовыми сценариями (test scripts) (либо ручными, либо автоматизированными).
Выполнение тестирования влечет за собой запуск тестов путем объединения последовательностей тестовых сценариев в набор тестов. Это продолжение ответа на вопрос, как тестировать (более конкретно, как провести тестирование).
Составление отчетности по тестированию отвечает на вопрос, как различные результаты тестирования анализируются и соотносятся между собой. Отчетность используется для определения текущего состояния процесса тестирования проекта, а также общего уровня качества приложения или системы.
Тестирование генерирует много информации. Из этой информации можно извлечь показатели (metrics), которые определяют, измеряют и отслеживают качество проекта. Затем эти **показатели качества** нужно передать в какой-нибудь коммуникационный механизм, используемый для остальных показателей проекта.
Самые типичные генерируемые при тестировании данные, которые часто служат источником для показателей качества, - это *дефекты*. Дефекты не являются статическими, а меняются во времени. Кроме того, некоторые дефекты часто связаны друг с другом. Эффективное **отслеживание дефектов** имеет решающее значение как для групп тестирования, так и для групп разработчиков.
Другие факторы в управлении тестированием
Кроме программных и аппаратных артефактов и ресурсов тестирования необходимо управлять **группой тестирования**. Управление тестированием должно координировать действия всех участников проекта, вовлеченных в процесс тестирования. Это требует контроля **пользовательских защиты и полномочий** для участников тестирования и артефактов. Для проектов, распределенных по нескольким центрам разработки или по нескольким группам разработчиков (что быстро становится нормой), необходима также организация координации центров и групп.
Конкретный **процесс тестирования** проекта будет иметь очевидное влияние на управление тестированием. Для итеративного проекта управление тестированием должно обеспечивать фундамент и руководство действиями по планированию, выполнению и оценке тестирования в итеративном режиме. Исходя из этого, **стратегия тестирования** должна следовать среде управления тестированием.
Связанные дисциплины разработки программного обеспечения

Хотя все дисциплины в разработке программного обеспечения имеют отношение к дисциплине тестирования, некоторые из них особенно важны для тестирования:

- Управление требованиями
- Управление изменениями
- Управление конфигурациями

Управление требованиями - это предшественник большого объема работ по тестированию, обеспечивающий значительное количество мотиваций для тестирования и требований для проверки. Процесс управления конкретными требованиями к проекту может иметь решающее влияние на процесс управления тестированием. Одной из аналогий этой взаимосвязи может быть эстафета, где первый бегун представляет управление требованиями, а следующий бегун, получающий эстафетную палочку, представляет управление тестированием. IBM® Rational® RequisitePro® является инструментальным средством для поиска, документирования, организации и отслеживания требований. Более подробная информация приведена на странице продукта IBM® developerWorks® Requisitepro.

Управление изменениями влияет на все части разработки программного обеспечения, но отслеживаемыми изменениями, наиболее релевантными для работ по тестированию, являются **дефекты**. Дефекты часто являются основным каналом передачи информации между тестированием и разработкой. Числа и показатели, определенные из дефектов, также часто используются как единицы измерения качества. ClearQuest представляет собой мощное, хорошо настраиваемое инструментальное средство для управления многочисленными типами изменений и работ в цикле разработки программного обеспечения. Более подробная информация приведена на странице продукта IBM® developerWorks® ClearQuest.

Управление конфигурациями важно для управления тестированием по причине слежения за тем, какие сборки в какое время необходимо тестировать. Управление конфигурациями контролирует сборки, а также среды, отслеживаемые системой управления тестами для выполнения тестирования. IBM® Rational® ClearCase® является ведущим инструментальным средством управления конфигурациями. Дополнительная информация приведена на странице продукта IBM® developerWorks® Clearcase.

Проблемы управления тестированием

Одним из способов подвести итог выполнения задач управления тестированием является ответ на следующие вопросы:

- Зачем нужно тестировать?
- Что нужно тестировать?
- Где выполнить тестирование?
- Когда выполнить тестирование?
- Как провести процесс тестирования?

Хотя это может выглядеть достаточно понятным на высоком уровне, существует много препятствий, которые часто возникают в типичном

процессе разработки программного обеспечения. Эти проблемы описываются ниже.

Недостаточно времени для тестирования

За исключением некоторых специализированных или очень ответственных приложений, очень мало программных проектов имеют достаточное количество времени в жизненном цикле разработки, для того чтобы достичь высокого уровня качества. Очень часто почти неминуемые задержки в программном проекте переносятся на и так короткий "цикл тестирования". Даже лучшие проекты, очень вероятно, имеют жесткие временные ограничения на выполнение тестирования. Результатами этого препятствия для управления тестированием являются постоянно меняющиеся приоритеты и переключение задач, а также уменьшенный объем данных для результатов тестирования и показателей качества.

Недостаточно ресурсов для тестирования

Кроме недостатка времени очень часто сложно получить нужные ресурсы, доступные для выполнения требуемых работ по тестированию. Ресурсы могут совместно использоваться другими задачами или другими проектами. В то время как аппаратные ресурсы для тестирования могут добавить задержки и трудности, проблему недостатка человеческих ресурсов может быть даже еще труднее решить. Результаты этого препятствия для управления тестированием приблизительно такие же, как и для временных ограничений.

Группа тестирования не всегда расположена в одном месте

Все чаще в наши дни ресурсы для тестирования могут быть доступны, но не в одном географическом местоположении. Использование талантливых разработчиков по всему миру становится общепринятым, но это вносит значительные технические препятствия. Как группам с разных континентов совместно использовать артефакты и сохранить согласованность без задержек и разногласий, влияющих на весь проект? Как можно максимизировать эффективность проекта при географически распределенной разработке?

Трудности с требованиями

Хотя существует много стратегий тестирования, проверка корректности требований обычно имеет главный, наивысший приоритет при тестировании, которое должно быть завершено. Это требует наличия полных, недвусмысленных и пригодных для тестирования требований. Управление не вполне точными требованиями может привести к более серьезным проблемам при выполнении тестирования. Использование такого инструментального средства как RequisitePro может значительно улучшить управление требованиями, а также содействует разработке хороших требований.

Для того чтобы управление тестированием было эффективным, должен иметься прозрачный доступ к самым последним (меняющимся) системным и бизнес-требованиям. Этот доступ должен быть не только к формулировке требований, но также к приоритету, состоянию и другим атрибутам. Кроме

того, это требует предельной координации и взаимодействия между группами, разрабатывающими требования, и группами, выполняющими тестирование. Это взаимодействие должно идти в обоих направлениях, чтобы гарантировать качество.

Сохранение синхронизации с процессом разработки

Еще одним необходимым для обеспечения качества программного обеспечения типом взаимодействия групп является взаимодействие между тестировщиками и разработчиками. За исключением критических дефектов почти традицией в разработке программного обеспечения стало то, что работа группы тестировщиков касается только самих тестировщиков. Однако эта работа имеет большое значение для того, чтобы каждый (особенно разработчики) понимал, каков текущий уровень качества и то, что уже было протестировано, а что нет.

Для того чтобы группы тестировщиков использовали свое драгоценное время эффективно, они должны всегда быть в курсе постоянных изменений в коде, сборках и средах. Управление тестированием должно точно идентифицировать, какую сборку тестировать, а также правильные среды, в которых нужно выполнять тестирование. Тестирование неправильных сборок (или функций) приведет к трате времени и может серьезно повлиять на график выпуска проекта. Тестировщики должны также знать, какие дефекты уже известны (и, следовательно, не должны тестироваться повторно), а какие уже исправлены. Тестировщики должны сообщать разработчикам об обнаруженных дефектах вместе с достаточной информацией, полезной для их устранения.

Составление отчетов с корректной информацией

Работа по тестированию полезна только тогда, когда можно выразить состояние тестирования и некоторые показатели качества проекта. Генерирование отчетов является достаточно простой процедурой, но представить корректную информацию (в нужное время и всем заинтересованным лицам) может быть сложнее, чем кажется, по нескольким причинам:

- Если имеется **слишком мало информации**, заинтересованные участники проекта не будут полностью понимать проблемы, влияющие на качество. Кроме того, это повлияет на восприятие ими группы тестировщиков – ее значимость уменьшится.
- Если имеется **слишком много информации**, значение и воздействие значимой информации становятся незаметными.
- Существует много **технических препятствий**, которые могут затруднить совместное использование информации различными группами участников проекта в различных месторасположениях.

В результатах, отображаемых в отчетах, важен также способ размещения информации и ее формат (то есть, информация может предоставляться на основе инструментальных средств, в браузере или в виде документов). Эффективность освоения заинтересованными участниками проекта результатов и информации о качестве будет снижена при наличии

технических или других ограничений в размещении или формате отчетов. Данные должны быть представлены в ясной и логичной форме, выражающей нужное значение, а не в схеме, ограниченной инструментальными средствами или технологией. Следовательно, для управления тестированием важно рассмотреть необходимость обеспечения гибкости и возможности предоставления широкого диапазона форматов отчетности.

Что такое показатели качества?

Одной из основных целей группы тестирования является оценка и определение качества, но как точно измерить качество? Существует много средств для этого, и они различны для разных типов систем или приложений, а также специфичны для проекта разработки. Все эти показатели качества должны быть понятны и недвусмысленны, для того чтобы избежать неправильной интерпретации. Еще важнее то, что показатели должны быть пригодны для сбора и хранения, в противном случае они могут не стоить затрат или могут быть неполными или неточными.