

Стратегии и планирование испытаний программных продуктов

Стратегия испытаний — совокупность выбранных методов и решений, вытекающих из целей и задач проекта и его тестирования, общие правила и принципы, способствующие достижению целей разработки программного продукта высокого качества.

Ведущий испытатель или тест-менеджер, готовящий план, должен обдумывать идеи, выводы и решения таким образом, чтобы специалисты, использующие план, четко понимали, что от них требуется: стратегия тестирования, планирование и использование ресурсов, управление конфигурацией проекта и минимизация рисков. При формировании стратегии испытаний необходимо соблюдать следующие рекомендации.

Тестирование в первую очередь следует проводить для требований с наивысшим приоритетом, которые либо представляют для заказчика наибольшую важность, либо могут причинить заказчику наибольшие неприятности в случае проявления дефектов программного продукта. Если запланировано тестирование всех требований и ресурсы это позволяют, то необходимо проверить выполнение всех требований. В случае недостаточных ресурсов перед предъявлением продукта заказчику необходимо тщательно протестировать требования с наивысшими приоритетами. Целесообразно получить согласие заказчика на то, что требования, которые были проверены частично или не были проверены вообще, не будут использоваться и поддерживаться вплоть до следующей версии продукта.

Приоритет при планировании испытаний целесообразно отдавать:

- покрытию критических вариантов использования системы и пользовательских данных;
- областям повышенного риска наличия дефектов;
- покрытию критических пробелов в тестовом покрытии;
- функциям, наиболее значимым для работы системы, и функциям повышенного риска.

Тестирование сначала осуществляют для новых функциональных возможностей, которые изменялись в целях исправления или совершенствования функций и харак-

теристик. Если версия является очередным обновлением или эксплуатируемой версией, то особое внимание следует уделить новым функциям и характеристикам. Любые изменения, внесенные в программы, могут исказить даже те части комплекса программ, которые непосредственно не затрагиваются изменениями. В этом случае следует как можно шире выполнять регрессионные тесты для всех функций и характеристик комплекса программ, какими бы ни были изменения.

Следует проводить тестирование тех компонентов и участков, в которых наиболее вероятно присутствие дефектов. Если обнаружен какой-то дефект, то зачастую рядом может быть еще несколько аналогичных дефектов. Если есть сведения от разработчиков, что часть компонентов породили проблемы во время модульного тестирования или проверки взаимодействия и функционирования компонентов, то такие участки необходимо отметить, чтобы обратить на них внимание при комплексном тестировании или системных испытаниях.

Тестирование желательно начинать с функций и конфигураций, с которыми наиболее часто будет иметь дело конечный пользователь или система. Для этого в спецификации требований полезно включать методику оценки случаев использования некоторых функций и иметь доступ к функциональному разрезу конечного пользователя — математическому ожиданию использования каждой функции. Это — дополнительный источник информации для установки приоритетов тестирования. Большая часть времени, отведенного на тестирование, должна затрачиваться на проверку наиболее часто используемых функций, конфигураций и операций.

Стратегия испытаний функций и характеристик комплекса программ должна последовательно:

- разрабатывать автоматизированные и (или) ручные тесты для покрытия всех функций, характеристик и рисков качества, которые определены как нуждающиеся в полном или сбалансированном приоритетном тестировании, с особым вниманием к факторам функционирования системы, наблюдаемым со стороны пользовательского интерфейса или доступного программного интерфейса системы;
- дополнять тестовые данные или условия тестовых сценариев для покрытия критических вариантов использования системы, пользовательских данных или сокращения известных дефектов в программном продукте;
- использовать исследовательское тестирование в областях, которые ранее не были покрыты тестами и которые, с точки зрения интуиции или результатов тестирования, являются областями повышенного риска наличия дефектов;

- использовать методы структурного покрытия для выявления не подвергавшихся тестированию областей и затем дополнять тесты для покрытия критических пробелов, если позволяют время и ресурсы или вызывают беспокойство возможные невыявленные риски.

При разработке требований к тестам группа тестирования должна выполнить несколько предварительных шагов, в том числе получить четкое представление о потребностях заказчика. Необходимо также изучить системные требования, сценарии использования системы и (или) описание назначения системы, для того чтобы лучше понять цель ее разработки. Еще один шаг — определение функций, наиболее значимых для работы системы, и функций повышенного риска. Определяются также инструменты тестирования, которые будут применяться для выполнения проекта. Требования к тестам для комплексов программ можно также получить на основе представлений о логике архитектуры системы. Разрабатывая требования к тестам на основе системных требований или сценариев использования системы, группа тестирования должна создавать, по крайней мере, одно требование к тестам на каждое системное требование.

Управление требованиями к тестам включает в себя хранение требований, отслеживание связей, оценку рисков требований к тестам, выстраивание последовательности требований к тестам и определение методов верификации тестов. Отслеживание связей предполагает отображение тестовых процедур на требования к тестам и дефектов на тестовые процедуры. Группа тестирования должна описать способ управления требованиями к тестам в плане тестирования.

Заказчик и разработчики должны анализировать и, в конечном счете, утверждать план испытаний комплекса программ, в котором описаны требования к тестированию и представлена матрица соответствия системных требований и тестовых сценариев. Матрица соответствия должна содержать информацию о требованиях, а также показывать взаимосвязь между требованиями и другими результатами проекта. Планирование испытаний включает в себя как определение требований к тестам, так и разработку процессов управления этими требованиями. Оно предполагает, что процессы тестирования, методы, методики, персонал, инструменты, план-график и оборудование организованы и эффективно применяются.

Когда план испытаний комплекса программ разработан, обновлен и полностью описывает процессы тестирования, он становится руководящим инструментом для

выполнения программы испытаний. Он уточняется путем корректировок целей, задач и стратегий тестирования, а также изменений требований к тестам, фиксирует параметры программы испытаний, которые должны быть документированы.

Планирование работ по испытаниям должно учитывать ресурсы и работы, которые необходимо выполнить, чтобы своевременно подготовить тестовую среду. Испытатели должны определить требования к аппаратному, программному и сетевому обеспечению в целях создания и поддержки адекватных изменений тестовой среды. Нужно спланировать работы по приобретению, установке и настройке компонентов, моделей или динамических генераторов тестовой среды. Заказчик должен утвердить стратегию испытаний и тестовые процедуры, которые должны быть подробно описаны в плане тестирования, и определить, какие сценарии и тесты когда будут выполняться.

План испытаний должен определять объем работ по испытаниям. Обычно выстраивают структуру работ, в которой на одном уровне определяются категории работ по тестированию, а на другом уровне даются подробные описания работ. Структура детализации работ используется в сочетании с хронометражем для определения времени выполнения каждого из этапов испытаний. Кроме того, план тестирования должен отражать оценки затрат на тестирование. Оценка затрат может определять число сотрудников группы тестирования в проекте в часах или в количестве людей, если на выполнение определенного объема работ оговаривается конкретный срок.

Должны быть документированы квалификация и навыки сотрудников, необходимых для проведения испытаний. Состав группы испытателей с требуемыми навыкам может быть обозначен в требованиях к их знаниям. Тест-менеджер должен оценить разницу между требуемой квалификацией и реальной подготовкой персонала, чтобы определить возможные направления обучения. Также необходимо определять и документировать в плане роли и ответственность сотрудников группы испытаний с учетом особенности продукта.

Любая программа испытаний должна иметь определенные рамки, отражающие ограничения по сотрудникам, человекочасам и графику. В плане тестирования также следует отражать допущения, предварительные условия и риски тестирования. Сюда включаются все события, действия или обстоятельства, которые могут помешать выполнению испытаний в срок. С разработкой плана должно быть связано определение функций, разработка которых имеет большое значение для успеха проекта, и функций, разработка которых связана с наибольшим риском. Определение наивысшего риска дает возможность группе испытателей сосредоточить усилия на

функциях высокой значимости с точки зрения достоверности результатов тестирования.

Требования к тестам должны заноситься в базу данных и (или) матрицу отслеживания требований. В базе данных или матрице с каждым требованием к тестам сопоставляется идентификационный номер компонента системной архитектуры программного продукта. Затем компонент архитектуры изучается, вплоть до детализированных требований к программным компонентам и системных требований или сценариев использования системы. После определения требований к тестам группа тестирования должна принять предварительное решение по методам тестирования, которые наилучшим образом будут проверять каждое требование.

В плане испытаний должны быть определены требования к аппаратному, сетевому и программному продукту, что позволит создать тестовую среду, являющуюся зеркальным отражением среды применения программного продукта, предназначенного для испытаний. Приобретение, установка и настройка различных компонентов тестовой среды должно тщательно планироваться. При составлении плана испытаний определяют требования к тестовым данным и средства для их получения, генерации или разработки. План тестирования должен указывать механизм управления целостностью тестовой модели, такой как обновление тестовой базы данных до начального состояния базовой версии для возможности поддержки регрессионного тестирования.

К ключевым элементам планирования испытаний относится тестирование эксплуатационной и технологической документации. Группа тестирования должна исчерпывающе документировать планы Программы испытаний, а испытатели обязаны подробно изучить содержание этих планов. Эти специалисты должны получить одобрение плана тестирования у конечного пользователя или заказчика. Заказчик должен утвердить стратегию тестирования и тестовые процедуры, которые подробно описаны в плане тестирования и определяют, какие тесты и когда будут выполняться. Кроме того, предполагается, что заказчик согласен с тем, что план тестирования и связанные с ним тестовые сценарии правильно проверяют удовлетворительное покрытие тестами системных требований или сценариев использования системы.

Разработка тестов включает в себя создание испытателями сопровождаемых, многократно применяемых, простых и надежных тестовых процедур, что может потребовать не меньше усилий, чем разработка программистами тестируемых текстов программ. Чтобы добиться максимального эффекта от автоматизации тестирования,

испытатели должны вести разработку тестов параллельно с созданием программистами текстов программного продукта. Схема структуры комплекса программ является графическим представлением основных работ, которые должны быть выполнены во время разработки тестов.

Анализ полноты покрытия тестами требований при исполнении программного продукта должен определять, какие требования не были протестированы и какие части структуры программного продукта не были исполнены при испытаниях. Тестирование структурного покрытия должно устанавливать, не пропущены ли элементы структуры программы, которые не проверены тестовыми процедурами, и покрыли ли тесты всю структуру комплекса программ.

Анализ покрытия версии программного продукта тестовыми данными, основанными на требованиях, должен определить, насколько полно тестирование проверило реализацию всех требований, и показать потребность в дополнительных тестовых сценариях. Поэтому должны выполняться анализ полноты структурного покрытия и проводиться его верификация. При тестировании реализации требований к функциям и характеристикам программных компонентов полнота их покрытия тестами редко достигает 90 %; хорошо, если она составляет около 80 %.

При проведении испытаний на основе требований к программному продукту, анализа технического задания и совокупности требований к функциям и характеристикам целесообразно оценивать представление о том, сколько нужно тестов для полного испытания версии программного продукта. Если составлять тесты, руководствуясь этим принципом, можно оценивать, какое время потребуется на разработку, исполнение и анализ полного комплекта тестов при создании очередной версии программного продукта.

Испытатели должны составлять план-график разработки и выполнения тестовых процедур по шкале времени реализации проекта как средство определения временных рамок для разработки и выполнения различных тестов (графики Ганта). План-график определяет зависимости между тестами и общие сценарии, которые будут постоянно использоваться при испытаниях. Перед созданием полного набора тестовых процедур для очередной версии программного продукта испытатели должны провести анализ связей между компонентами.

Результаты этого анализа помогут выявить независимые компоненты, спланировать зависимости между тестами и выделить общие сценарии, которые могут повторно применяться в процессе тестирования последующих версий. Для этого строится матрица связей, которая показывает взаимозависимость различных тестовых

сценариев. Графическое представление помогает испытателям определять возможности многократного применения сценариев в различных комбинациях с использованием различных оболочек, что позволяет свести к минимуму объем работ по созданию и сопровождению версий тестовых сценариев. В ходе разработки тестовых процедур группа тестирования должна проводить конфигурационное управление и контроль для тестовых сценариев и тестовых данных, а также для каждой отдельной тестовой процедуры.

При группировании автоматизированных тестовых процедур необходимо исключать их дублирование. Нужно изучать программу, план испытаний и тестовые сценарии, чтобы проверить, как результаты анализа и проектирования тестов учтены в план-графике. При составлении графика разработки и выполнения тестовых процедур необходимо учитывать приоритеты и риски, присвоенные различным тестам. График тестирования должен уделять большее внимание проверке функций, наиболее значимых для программного продукта, и функций повышенного риска. Такие тесты должны выполняться в первую очередь, и график обязан предусматривать достаточно времени для проверки этих функций и, в случае необходимости, для регрессионного тестирования.

Прогнозирование затрат на испытания комплексов программ возможно на основе обобщения статистических данных ряда предшествующих проектов. В данном случае задача ограничена только ориентировочным перечнем основных составляющих затрат, которые целесообразно учитывать в процессе испытаний. Этот перечень может использоваться как ориентир при подготовке программы испытаний. Обычно наиболее важным для реализации проекта и зависящим от большинства его особенностей и факторов является трудоемкость, непосредственно определяющая стоимость испытания создаваемого комплекса программ.

Значения длительности разработки и числа специалистов взаимосвязаны и в некоторых пределах могут размениваться. Поэтому оценки этих показателей затрат можно варьировать. При недостаточном числе специалистов естественно возрастает длительность разработки, хотя трудоемкость может остаться неизменной. Многократное применение одних и тех же апробированных компонентов и (или) многократная адаптация комплекса к различным условиям применения является одним из перспективных методов повышения качества и снижения затрат труда специалистов на испытания комплексов программ.

Получение оценки трудозатрат на выполнение испытаний и обнаружение дефектов версии программного продукта включают в себя следующие этапы:

- определение перечня и состава задач испытаний, которые должны быть выполнены;
- оценка трудозатрат на решение отдельных задач и всего процесса испытаний;
- определение времени, требуемого для решения каждой задачи и длительности всего квалификационного тестирования;
- построение подробного расписания и поэтапного графика решения каждой тестовой задачи;
- оценка рисков невыполнения графика работ и формулировка планов их снижения;
- затраты на обеспечение требований надежности и безопасности функционирования комплекса программ.

Подробное рассмотрение этапов получения оценки трудозатрат на выполнение испытаний проекта крупного комплекса программ позволяет определить их содержание.

Определение перечня и состава задач испытаний, которые должны быть выполнены. Эта оценка начинается с определения работ, которые необходимо выполнить, для того чтобы тестирование программного продукта считалось состоявшимся. На этом этапе может быть достаточно разбить работу на крупные функциональные задачи, отражающие проверку реализации конкретных требований к функциям и характеристикам комплекса программ. Если используются менее формальные методы, то результатом этого этапа может быть простой список основных задач.

Оценка трудозатрат на решение отдельных задач и всего процесса испытаний. Каждая задача, выявленная на первом этапе, требует для своего решения определенных трудозатрат, представляющих собой объем работ, необходимых для выполнения соответствующей задачи. Оценки этих трудозатрат могут быть представлены в виде произведения количества исполнителей на затраченное ими время и измеряться в таких единицах, как человекодень или человекомесяц.

При тестировании комплексов программ основным лимитирующим ресурсом обычно являются допустимые трудозатраты специалистов, а также ограничения на сроки разработки версии программного продукта, параметры ЭВМ и технологию проектирования. Одним из наиболее важных компонентов планирования тестирования является оценка трудоемкости и времени, необходимых для его выполнения. Затраты на тестирование реализации требований версий сложных программных продуктов могут составлять существенную часть стоимости проекта; при этом необходимо, чтобы тестирование проводило достаточное число специалистов и у них было

достаточно времени на качественное выполнение задач по корректировкам комплекса программ. Ограничения реальных ресурсов на верификацию и тестирование определяют достижимое качество версий программных продуктов.

Определение времени, требуемого для решения каждой задачи и длительности всего квалификационного тестирования. Время, необходимое для решения задачи, измеряется в днях, неделях или месяцах. Время, необходимое для выполнения той или иной задачи, зависит от количества исполнителей, однако эта зависимость не обязательно линейная. Суммарная продолжительность работ по тестированию зависит от продолжительности решения отдельных функциональных задач. Однако это не простое суммирование, поскольку некоторые задачи можно решать и одновременно с другими.

Построение подробного расписания и поэтапного графика решения каждой тестовой задачи. На основе результатов предыдущих этапов можно построить график выполнения работ (возможно, в виде диаграммы Ганта) и вычислить сумму наиболее важных трудовых и временных затрат на испытания комплекса программ.

Оценка рисков невыполнения графика работ и формулировка планов их снижения. Следует оценивать возможные проблемы и риски, которые могут возникнуть при решении задач в запланированные промежутки времени, и предусмотреть средства решения этих проблем. Прежде чем приступить к более подробному анализу перечисленных шагов, важно учитывать, насколько точными могут быть оценки. После того как все требования будут проанализированы и утверждены, оценка стоимости продукта обычно отличается от окончательной, фактической, стоимости примерно в два раза. При этом общие затраты на тестирование на всех этапах разработки могут достигать 30 ... 40 % полной трудоемкости создания комплекса программ. Из них затраты на испытания критических систем совместно с заказчиком могут составлять до половины этих затрат. В связи с неточностями, свойственными таким процессам оценки, целесообразны мониторинг и периодическое возвращение к оценке трудозатрат и графика выполнения тестирования (это — неотъемлемая часть организации работ по испытаниям).

Затраты на обнаружение и устранение дефектов и ошибок в программе определяются двумя факторами: затратами на обнаружение каждой ошибки и затратами на устранение выявленных ошибок при формировании очередной версии. Чем меньше ошибок в программе, тем труднее они обнаруживаются, т. е. тем выше затраты на выявление каждой ошибки. Затраты на устранение ошибок и корректировку программ пропорциональны числу дефектов, выявляемых между очередными версиями

программного продукта. Непрерывно требуются затраты для контроля состояния версий комплекса программ и обеспечения их сохранности. Исходя из опыта работ, широко тиражируемый комплекс программ объемом около 105 строк может требовать при испытаниях непрерывных усилий коллектива в составе десятка и более специалистов для устранения ошибок, корректировок версий и документации.

Затраты на завершающие испытания программного продукта в целом обычно могут быть достаточно четко выделены из остальных затрат, так как в этих процессах непосредственно участвуют заказчик и пользователи. Величина этих затрат без учета ресурсов, необходимых для моделирования динамической внешней среды, может составлять около 10 ... 15 % общих затрат труда и 10 % времени на разработку программного продукта. При этом практически невозможно разделять затраты на оценивание реализации отдельных требований и характеристик.

Возрастание относительных затрат всех видов при квалификационном тестировании программного продукта систем реального времени (СРВ) обусловлено высокой трудоемкостью динамической отладки и тестирования в реальном времени. При этом следует учитывать, что абсолютные значения трудоемкости испытаний программ СРВ приблизительно в пять раз выше, чем программ административных систем. Относительная трудоемкость и длительность при испытаниях различных классов программных продуктов приблизительно одинакова и составляет 8 ... 10 % совокупных затрат на разработку. Однако абсолютная трудоемкость для испытаний административных систем также приблизительно в пять раз меньше, чем программ СРВ. При этом следует учитывать, что при изменении трудоемкости полной разработки в пять раз абсолютная длительность этого процесса сокращается всего в 1,5—1,7 раза. В результате полная длительность отладки и испытаний программ административных систем также приблизительно в полтора раза меньше, чем программы СРВ того же размера.

Затраты на обеспечение требований надежности и безопасности функционирования комплекса программ. Такие затраты определяются требуемым их уровнем и сложностью (размером) комплекса программ. При наличии особенно высоких требований к безопасности критических программных продуктов эти затраты могут в 2—3 раза превышать затраты на решение базовых, функциональных задач. Для типовых административных систем трудоемкость создания программных средств защиты обычно составляет 20 ... 40 % затрат на решение основных, функциональных задач. В более простых случаях, доля таких затрат может снижаться до 5 ... 10 %.

Затраты на обеспечение высокой надежности близки к затратам на обеспечение безопасности. В составе разработки они могут достигать 2—3-кратного увеличения затрат при требованиях наработки на отказ в десятки тысяч часов. Для минимального обеспечения автоматического рестарта в ординарных системах затраты составляют около 10 ... 20 %.

В любых системах должен присутствовать минимум программных компонентов, обеспечивающих надежность и защиту от преднамеренных и случайных угроз.

Затраты на сопровождение и корректировку дефектов версий программных продуктов можно считать аддитивными и включающими в себя затраты:

- на тестирование для обнаружения и устранения ошибок и дефектов в каждой версии программного продукта;
- доработку и совершенствование программ, формирование и испытание новых модернизированных версий;
- конфигурационное управление, тиражирование каждой новой версии и ее внедрение в эксплуатируемых и новых системах.

Доля каждой составляющей в общих затратах на сопровождение может значительно изменяться в зависимости от особенностей сферы применения и жизненного цикла конкретного программного продукта. Для долгоживущих (около 10 лет), многократно тиражируемых (1 000 ... 100 000 экземпляров) комплексов доминирующими обычно являются затраты на модернизацию и доработку версий программного продукта.

Затраты на совершенствование и модернизацию комплексов программ близки по содержанию (но не по величине) к затратам на их первичную разработку. Модернизация обычно производится поэтапно. Для каждой новой версии изменяется (разрабатывается) только некоторая часть от всего объема программного продукта. Экспериментально установлено, что эта часть при вводе очередной версии обычно составляет 10 ... 20 % объема всего комплекса. Сложность связей компонентов приводит к тому, что удельные затраты на изменяемые программы при модернизации каждой версии могут быть в 2—3 раза больше, чем затраты на создание программ такого же объема при разработке первой версии. Эта величина зависит от того, насколько путем стандартизации архитектуры и интерфейсов предусматривались перспективы совершенствования комплекса программ.

Затраты на модернизацию зависят от тиража косвенно, вследствие расширения условий применения конкретного комплекса и увеличения потока запросов пользователей на развитие программ. Также косвенно влияет тираж на запросы для устранения выявленных ошибок. Для выполнения этих работ иногда привлекают коллектив специалистов, осуществивших первичную разработку проекта. Такая организация наиболее характерна для уникальных, заказных программных продуктов. В этих случаях первичную разработку и модернизацию трудно разделить. Для широко тиражируемых программных продуктов к сопровождению часто привлекается специальный коллектив, не проводивший первичную разработку.

Финансирование испытаний программного продукта целесообразно определять специальным разделом договора между разработчиком и заказчиком на разработку версии программного продукта. В техническом задании и в контракте следует четко определять порядок квалификации видов и причин изменений в программах при испытаниях, а также распределение ответственности за их инициализацию, реализацию и финансирование. Ошибки, выявленные в компонентах и комплексе программ, которые искажают реализацию функций, согласованных с заказчиком в контракте и требованиях спецификаций, а также отраженные в документации на версию, должны устраняться за счет разработчика. Модификацию и расширение функций и характеристик компонентов или создание новых версий программного продукта, ранее не отраженных в требованиях технического задания и контракте с заказчиком, следует квалифицировать как дополнительную работу с соответствующим финансированием заказчиком.

После передачи версии программного продукта в эксплуатацию затраты ресурсов на обнаружение и первичную квалификацию дефектов ложатся в основном на непосредственных пользователей. На разработчиков (поставщиков) программного продукта ложатся затраты на анализ и локализацию источников и причин дефектов и их устранение, которые зависят от характеристик выявляемых дефектов, масштаба комплекса программ, организации и технологии его разработки, инструментальной оснащенности сопровождения, квалификации специалистов, а также от тиража и активности применения данного программного продукта. Заранее перечисленные факторы прогнозировать невозможно, оценки этой составляющей затрат целесообразно проводить по результатам начальных этапов сопровождения и модификаций первых версий.

Затраты на тиражирование и адаптацию к параметрам среды пользователей зависят от широты распространения программного продукта и могут оцениваться по типовым прецедентам аналогичных проектов или предшествующих версий.

При сопоставлении результатов оценивания функций и характеристик качества с требованиями технического задания и спецификаций разработчик или поставщик обязан удовлетворять требования заказчика только в пределах согласованных параметров модели внешней среды и системы. Оценивание функционирования комплекса программ за этими пределами должно дополнительно согласовываться испытателями с разработчиком. При этом невыполнение требований может квалифицироваться как их расширение за пределы, ограниченные контрактом, и не учитываться при оценивании заказчиком характеристик качества программного продукта, или как дополнительные работы, подлежащие соответствующему финансированию со стороны заказчика, для доработки комплекса программ в целях удовлетворения этих требований.