

Системные основы разработки требований к программным продуктам

Стоимость аппаратных средств постепенно снижается, тогда как стоимость программных продуктов стремительно возрастает. Возникла необходимость в новых технологиях и методах управления комплексными проектами разработки больших программных продуктов. Такие методы составили программную инженерию. Возрастает как размер целостного программного продукта, так и его сложность.

Программная инженерия не рассматривает технические аспекты детального создания компонентов; в ее ведение входят такие задачи, как управление проектами и разработка методов, теорий и средств, необходимых для обеспечения жизненного цикла сложных комплексов программ. Программирование компонентов — это в основном индивидуальная работа, а программная инженерия систем — всегда коллективная работа.

Технологии программной инженерии часто являются критическим фактором при разработке сложных вычислительных систем. Высокие темпы роста основных ресурсов аппаратных средств (приблизительно на порядок каждые пять лет) и сохраняющаяся потребность в увеличении их использования со стороны различных пользователей и сфер применения приводят к необходимости совершенствования технологий создания крупных комплексов программ. Методологической базой такого развития может служить Свод знаний о программной инженерии — стандарт ISO/IEC 19759:2005 — SWEBOOK, в котором сосредоточены основные концепции в этой сфере деятельности.

Системные, функциональные требования должны отражать высокоуровневые требования к задачам и функциям программного комплекса, содержащего ряд взаимосвязанных подсистем. При этом система может или быть целиком программной, или состоять из программной и аппаратной частей. В общем случае частью системы может быть персонал, выполняющий определенные функции системы. Программные средства все больше встраиваются в различные системы. Работа с такими проектами требует от программного инженера широкого взгляда на общие задачи проектирования систем. Программному инженеру необходимо участвовать в выработке требований для всей системы, а также освоить прикладную область, требованиям которой должен будет отвечать программный продукт.

В требованиях к продукту и процессу должны быть разграничены соответствующие требования как свойств продукта, который необходимо получить, так и процесса, с помощью которого продукт будет создаваться. При этом ряд требований может быть заложен неявно и программные требования могут порождать требования к процессу. Они устанавливают основные соглашения между пользователями (заказчиками) и разработчиками в отношении того, что должна делать система и чего от нее не стоит ожидать. Документ должен включать в себя процедуры проверки получаемого программного продукта на соответствие предъявляемым ему требованиям, характеристики, определяющие качество и методы его оценки, вопросы безопасности и другие свойства.

Задача заключается в том, чтобы программные требования были ясны, связи между ними прозрачны, а содержание спецификаций не допускало разночтений и интерпретаций, способных привести к созданию программного продукта, не отвечающего потребностям заинтересованных лиц.

Исходные проектные данные и требования к программному продукту, включая установленные законодательные и регламентирующие нормативные требования, должны быть оформлены документально, а их выбор — проанализирован поставщиком на адекватность. Спецификацию требований должен представить потребитель-заказчик. Однако по взаимному согласию ее может подготовить поставщик-разработчик в тесном сотрудничестве с потребителем для предупреждений разногласий путем, например, уточнения определений терминов, объяснения предпосылок и обоснования требований. Неполные, двусмысленные или противоречивые требования должны быть предметом урегулирования с лицами, ответственными за их предъявление.

Для конкретного комплекса программ требования выделяются и определяются его функциональным назначением.

Программы для ЭВМ как объекты проектирования, разработки, испытаний и оценки качества характеризуются в жизненном цикле обобщенными характеристиками:

- проблемно-ориентированной областью применения, техническим и социальным назначением программного комплекса;
- конкретным классом и назначением решаемых функциональных задач с достаточно определенной областью применения соответствующими пользователями;

- масштабом и сложностью совокупности программ и базы данных, решающей единую целевую задачу системы;
- архитектурой комплекса программ и базы данных;
- необходимыми составом и требуемыми значениями характеристик качества функционирования программ и величиной допустимого ущерба — риска из-за недостаточного их качества;
- составом потребителей характеристик комплекса программ, для которых важны соответствующие атрибуты качества;
- комплектом стандартов и их содержания, которые целесообразно использовать при выборе характеристик комплекса программ;
- реальными ограничениями всех видов ресурсов проекта;
- степенью связи решаемых задач с реальным масштабом времени или допустимой длительностью ожидания результатов решения задач;
- прогнозируемыми значениями длительности эксплуатации и перспективой создания множества версий программного продукта;
- предполагаемым тиражом производства и применения комплекса программ;
- степенью необходимой документированности программного продукта.

Исходные требования к комплексу программ могут быть представлены и согласованы в составе спецификации всей системы. Если программный продукт нуждается во взаимодействии с другими программными или аппаратными продуктами, то в спецификации требований должны быть оговорены непосредственно или при помощи ссылок интерфейсы между разрабатываемыми и другими применяемыми продуктами. В этом случае должны быть разработаны процедуры, обеспечивающие четкое распределение системных требований между аппаратными и программными компонентами, а также соответствующие спецификации интерфейсов с внешней средой. При заключении контракта спецификация требований может быть определена не полностью. Она может дорабатываться в ходе реализации проекта; могут уточняться:

- требования к входной информации;
- источники информации и их идентификаторы;

- перечень и описание входных сообщений (идентификаторы, формы представления, регламент, сроки и частота поступления);
- перечень и описание структурных единиц информации входных сообщений или ссылка на документы, содержащие эти данные;
- описание и оценка преимуществ и недостатков разработанных альтернативных вариантов функций в концепции создания проекта;
- обоснование выбора оптимального варианта требований к содержанию и приоритетам функций компонентов;
- общие требования к структуре, составу компонентов и интерфейсам с внешней средой.

Требования к назначению и характеристикам функциональной пригодности должны выполняться для любых классов программных средств. Однако номенклатура учитываемых требований к конструктивным характеристикам качества существенно зависит от назначения и функций комплексов программ. Например, при проектировании систем управления объектами в реальном времени наибольшее значение имеют время реакции (отклика на задание), защищенность, корректность и надежность функционирования стабильного комплекса программ и менее важными могут быть качество обеспечения сопровождения и конфигурационного управления, способность к взаимодействию и практичность.

Для административных систем кроме корректности важно обеспечивать практичность применения, комфортное взаимодействие с пользователями и внешней средой. Может не иметь особого значения эффективность использования вычислительных ресурсов и обеспечение мобильности комплекса программ.

Для пакетов прикладных программ вычислений или моделирования процессов можно не всегда учитывать их мобильность, защищенность и временную эффективность, но обязательно учитывать корректность и точность расчетов.

При планировании, разработке и реализации требований к характеристикам программного средства необходимо в первую очередь учитывать:

- функциональную пригодность (функциональность) конкретного проекта программного продукта;
- возможные конструктивные характеристики качества комплекса программ, необходимые для улучшения функциональной пригодности;
- доступные ресурсы для создания и обеспечения всего жизненного цикла программного средства с требуемым качеством.

Эти факторы целесообразно применять последовательно, итерационно на каждом из основных этапов жизненного цикла комплекса программ. При первоначальном определении требований к функциональной пригодности и конструктивным характеристикам качества заданные заказчиком ограничения ресурсов не всегда могут учитывать ряд особенностей проекта, что обусловит недопустимое снижение (или завышение) требований к некоторым характеристикам качества. Анализ и разработку требований к качеству программного средства целесообразно проводить в два этапа: сначала предварительно увеличивая функциональную пригодность и конструктивные характеристики качества, а затем уменьшая риски снижения требуемого качества или используемых ресурсов.

Понятие дефекта или ошибки в программе в общем случае подразумевает неправильность, погрешность или неумышленное искажение объекта или процесса, что может стать причиной ущерба — риска при функционировании и применении программного продукта. При этом должно быть известно или задано правильное (эталонное) состояние объекта или процесса, по отношению к которому может быть определено наличие отклонения: ошибка или дефект. Исходным эталоном обычно является спецификация требований заказчика или потенциального пользователя, предъявляемая к программному продукту.

Любое отклонение результатов функционирования программы от предъявляемых к ней требований и сформированных по ним эталонов-тестов следует квалифицировать как ошибку. Различие между ожидаемыми и полученными результатами функционирования комплекса программ могут быть следствием ошибок не только в созданных программах, но и в первичных требованиях спецификаций, явившихся базой при создании эталонов. Тем самым проявляется объективная реальность, заключающаяся в невозможности абсолютной корректности и полноты исходных требований и эталонов для проектов крупных программных продуктов.

Изучение характеристик ошибок непосредственно связано с достигаемой корректностью, безопасностью и надежностью функционирования комплексов программ и помогает:

- оценивать реальное состояние проекта и планировать необходимую трудоемкость и длительность для его завершения и устранения ошибок;
- выбирать методы и средства автоматизации тестирования и отладки комплекса программ, адекватные текущему состоянию разработки и сопровождения, наиболее эффективные для устранения определенных видов дефектов;

- рассчитывать необходимую эффективность контрмер и дополнительных средств оперативной защиты от потенциальных дефектов и невыявленных ошибок;
- оценивать требующиеся ресурсы ЭВМ по расширению памяти и производительности с учетом затрат на реализацию контрмер при модификации программ для устранения дефектов и ошибок.

На практике исходные требования — эталоны — поэтапно уточняются, модифицируются, расширяются и детализируются по согласованию между заказчиком и разработчиком. Базой таких уточнений являются неформализованные представления и знания специалистов — заказчиков и разработчиков, а также результаты промежуточных этапов проектирования и тестирования. Однако установить некорректность таких эталонов еще труднее, чем обнаружить дефекты в программах, так как отсутствуют точные, формализованные данные, которые можно использовать как исходные.

Особенностью процесса выявления ошибок в программах является отсутствие полностью определенной программы-эталона, которой должны соответствовать результаты функционирования разрабатываемой программы. Поэтому установить наличие и локализовать дефект в программе непосредственным ее сравнением с эталонной программой абсолютно без ошибок невозможно. При отладке и тестировании обычно сначала обнаруживаются вторичные ошибки, т. е. последствия и результаты проявления некоторых внутренних дефектов или некорректностей требований или функционирования программ.

Эти внутренние дефекты следует квалифицировать как источники — первичные ошибки или причины обнаруживаемых дефектов результатов. Последующая локализация и корректировка таких первичных ошибок должна приводить к устранению ошибок, первоначально выявляемых в результатах функционирования программ.

Потери эффективности и качества программ за счет неполной корректности в первом приближении можно считать прямо пропорциональными (с коэффициентом) вторичным ошибкам. Оценка последствий, отражающихся на вторичные ошибки при функционировании программ, может производиться по значениям ущерба (риска) вследствие неустраненных их причин — первичных ошибок в программе. Вторичные ошибки являются определяющими для эффективности функционирования программ, однако не каждая первичная ошибка вносит заметный вклад в выход-

ные результаты. Вследствие этого ряд первичных ошибок может оставаться необнаруженным и, по существу, не влияет на функциональные характеристики комплексов программ.

Ошибкам в программах предшествуют их обнаружение и устранение на основе вторичных проявлений. Наибольшее число первичных ошибок вносится на этапах системного анализа, разработки или модификаций программ. При этом на долю системного анализа приходится наиболее сложные для обнаружения и устранения дефекты. Общие тенденции заключаются в быстром росте затрат на устранение ошибок на последующих этапах процессов разработки комплекса программ.

Интенсивность проявления и обнаружения вторичных ошибок наиболее высока на этапе активного тестирования и автономной отладки программных компонентов. Затем она снижается примерно экспоненциально. Различия интенсивностей устранения первичных ошибок на основе их вторичных проявлений и внесения первичных ошибок при корректировках программ определяют скорость достижения заданного качества версий программного продукта.

Отчеты об ошибках — экспертные оценки, инспекции, быстрые просмотры — мощные инструменты для раннего обнаружения проблем в продуктах и документах любых видов, включая ошибки в программах. Полезный отчет об ошибке должен включать в себя краткое описание, по которому специалисты по управлению изменениями или комиссии по разбору ошибок могут оценить проблему. Краткое описание необходимо для расстановки приоритетов устранения ошибок.

Подготовка полезного отчета об ошибке должна начинаться с надежного, хорошо организованного тестирования. Воспроизведение проблемы (дефекта) способствует ее пониманию и позволяет фиксировать последовательность шагов, которые заново приведут к проявлению ошибки. На поведение программы могут влиять многие факторы, а некоторые из них воздействуют на симптомы ошибки. Поиск общего правила возникновения ошибок углубляет представление о системе. Отчеты об ошибках после того, как их занесли в систему управления дефектами, остаются там навсегда и могут быть прочитаны людьми, не участвующими в проекте.

Источниками ошибок в комплексах программ являются специалисты — конкретные люди с их индивидуальными особенностями, квалификацией, талантом и опытом (табл. 5.1). Выделяют предсказуемые дефекты требований, модификаций, расширения и совершенствования ПС и необходимые изменения, обусловленные выяв-

лением случайных, непредсказуемых дефектов и ошибок. Вследствие этого плотность потоков и размеры необходимых корректировок в требованиях к комплексу и компонентам программ могут различаться в десятки раз.

Таблица 5.1. Специалисты и типы первичных дефектов и ошибок

Заказчик проекта программного продукта	Дефекты организации проекта и исходных требований заказчика на программный продукт
Менеджер проекта программного продукта	Дефекты, обусловленные реальной сложностью программного продукта
Менеджер — архитектор комплекса программ	Ошибки организации проектирования и архитектуры комплекса программ
Проблемно-ориентированные аналитики и системные архитекторы	Системные и алгоритмические дефекты и ошибки комплекса программ
Спецификаторы компонентов комплекса программ	Алгоритмические ошибки компонентов и документации комплекса программ
Системные интеграторы комплекса программ	Системные ошибки и дефекты реализации версий программного продукта и документации

Таблица 5.1. Специалисты и типы первичных дефектов и ошибок

Тестировщики требований и реализации компонентов и комплекса программ	Пропущенные системные, программные и алгоритмические ошибки компонентов, комплекса программ и документации
Управляющие сопровождением и конфигурацией версий программного продукта	Ошибки проектирования и реализации версий программного продукта
Документаторы программного продукта	Дефекты и ошибки технологических и эксплуатационных документов программного продукта

В крупных комплексах программ статистика и распределение типов ошибок и выполняемых изменений для коллективов разных специалистов нивелируются и проявляются достаточно общие закономерности, которые могут использоваться как ориентиры при их выявлении.

Одной из основных причин ошибок в комплексах программ являются организационные дефекты при определении и использовании требований к программному продукту, которые отличаются от остальных типов и условно могут быть выделены как самостоятельные. Ошибки и дефекты данного типа появляются из-за недостаточного понимания коллективом специалистов целей и функций комплекса программ, а также вследствие отсутствия четкой его организации и поэтапного контроля требований качества продуктов.

При отсутствии планомерной и методичной разработки и тестирования требований может оставаться невыявленным значительное количество ошибок, прежде всего, дефекты требований к взаимодействию отдельных функциональных компонентов между собой и с внешней средой. Для сокращения этого типа массовых ошибок

активную роль должны играть руководители и аналитики-системотехники, способные осуществлять контроль и конфигурационное управление требованиями, изменениями и развитием версий и компонентов комплексов программ.

Ошибки оценки характеристик системы и внешней среды, принятых в процессе разработки комплекса программ за исходные, могут быть результатом аналитических расчетов, моделирования или исследования аналогичных систем. В ряде случаев может отсутствовать полная адекватность предполагаемых и реальных характеристик, что является причиной сложных и трудно обнаруживаемых системных ошибок и дефектов развития проекта. Ситуация с такими ошибками дополнительно усложняется тем, что эксперименты по проверке взаимодействия программного продукта с реальной внешней средой во всей области изменения характеристик зачастую сложны и дороги, а в отдельных случаях, при создании опасных ситуаций, недопустимы. В этих случаях приходится использовать моделирование и имитацию внешней среды с заведомым упрощением ее отдельных элементов и характеристик, хотя степень упрощения не всегда можно оценить с необходимой точностью. Однако полной адекватности моделей внешней среды и реальной системы добиться трудно, а во многих случаях — невозможно, что может стать причиной большого количества дефектов.

Системные ошибки и недостатки требований к программному продукту определяются, прежде всего, неполной информацией о реальных процессах, происходящих в источниках и потребителях информации. Кроме того, эти процессы зачастую зависят от самих алгоритмов и поэтому не могут быть достаточно определены и описаны заранее без исследования функционирования комплекса программ во взаимодействии с внешней средой. На начальных этапах разработки не всегда удается точно и полно сформулировать целевую задачу всей системы, а также целевые задачи основных функциональных групп программ, и эти задачи уточняются в процессе проектирования.

Во многих случаях отсутствует полная адекватность условий получения предполагаемых и реальных характеристик внешней среды, что может являться причиной сложных и трудно обнаруживаемых ошибок. Это усугубляется тем, что очень часто невозможно заранее предусмотреть все разнообразие возможных внешних условий и реальных вариантов сценариев функционирования и применения версий программных продуктов. При автономной и в начале комплексной отладки версий компонентов относительная доля системных ошибок может быть невелика (около 10%), но она существенно возрастает (до 35 ... 40%) на завершающих этапах комплексной

отладки новых версий программного продукта. В процессе сопровождения системные ошибки обычно являются преобладающими (около 60 ... 80% от всех ошибок).

Системные ошибки корректности выполнения требований к программному продукту считаются наиболее критичными для общего успеха его версий и системы в целом. Ошибки требований являются наиболее трудными для обнаружения и наиболее сложными для исправления. Вот почему исправление ошибок выполнения требований может быть в 15—70 раз дороже, чем ошибок программирования.

Требование может быть пропущено в спецификации к системе и комплексу программ. Это ведет к неудовлетворенности пользователя, и комплекс программ считается заказчиком и пользователем дефектным. Пропуск части требований — наиболее обычная проблема среди ошибок требований. Ошибки требований могут также представлять собой конфликтующие требования в спецификации. Может проявляться неопределенность требования, т. е. требование сформулировано недостаточно ясно, чтобы привести к единственному, конструктивному решению при разработке комплекса программ, даже если и не конфликтует с другим требованием. Конечный пользователь часто называет это ошибкой, хотя на самом деле это выбор конструктивного решения на основе неполного или неопределенного требования.

Многочисленные исследования показали, что ошибки требований труднее всего обнаруживать и дороже всего исправлять.

Ошибки проектирования и разработки архитектуры программного комплекса определяются процессами перевода неопределенных и общих положений, сделанных на этапе первичных спецификаций требований, в более точные технические описания сценариев того, как программный продукт и система должны работать.

Ошибки структуры инспекциями легче обнаружить, чем ошибки требований, но они в конечном итоге могут оказаться при корректировках столь же дорогостоящими. Главная причина того, что ошибки структуры дорого исправлять, заключается в том, что они могут влиять на систему в целом.

Ошибки архитектуры можно разделить на три категории: пропуски, конфликты и ошибки перевода. Пропуски означают неспособность включить изменения одного или более требований в окончательную структуру комплекса программ. Когда пропуск новой функции или компонента попадает в окончательную структуру, он отражается ошибкой в конечном программном продукте.

Ошибки, которые основаны на конфликтах на этом уровне, часто невозможно исправить без полного исправления версии программного продукта. Ошибки перевода

проявляются, когда требования интерпретируются неправильно, по крайней мере, с точки зрения конечного пользователя. Если разработчик структуры либо неверно прочитает требования, либо не увидит содержание требования (так же, как заказчик или конечный пользователь), то появится ошибка разработки структуры данного компонента или комплекса программ.

Для организации эффективного тестирования конкретного комплекса программ целесообразно унифицировать регистрацию ошибок и дефектов на основе шаблона типовых отчетов специалистов о выявленных дефектах и предложениях по корректировке комплекса программ, включающих в себя:

- идентификатор специалиста, представившего отчет о дефектах и (или) предложениях;
- дату фиксирования дефекта или предложения на изменение комплекса программ;
- номер и параметры адаптации пользовательской версии программного продукта, на которой обнаружен дефект;
- идентификацию и регистрацию аномального функционирования программного продукта, несогласованности процессов и результатов с требованиями, планами и стандартами разработки, недостатков документации;
- описание ошибки и (или) дефекта, достаточное для его понимания и его устранения, описание возможных корректирующих действий, предназначенных для устранения зарегистрированного дефекта;
- подробное описание сценария и исходных данных, при которых выявлен дефект и документы результатов его регистрации;
- предположение о причине, вызвавшей проявление ошибки и (или) дефекта;
- описание возможных корректирующих действий, предназначенных для устранения зарегистрированного дефекта;
- тесты, исходные данные и сценарий, при которых повторно проявляется выявленный дефект;
- результаты анализа и рекомендации о возможных способах устранения дефекта или о реализации предложения по совершенствованию комплекса программ;
- оценки сложности, трудоемкости, эффективности и срочности корректировки комплекса программ для устранения дефекта;
- оценки влияния предлагаемых изменений на возможность эксплуатации версий программного продукта, имеющихся у пользователей.

Приведенные характеристики типов дефектов и количественные данные могут служить ориентирами при прогнозировании возможного уровня невыявленных ошибок во время тестирования различных сложных программных продуктов высокого качества. Следующим логическим шагом процесса их оценивания может быть усреднение для большого числа проектов фактических данных о количестве ошибок на конкретном предприятии, приходящихся на 1 000 строк кода, которые обнаружены в различных комплексах программ.

Непредсказуемость конкретных ошибок в программах приводит к целесообразности последовательного тестирования, методичного фиксирования и анализа возможности проявления любого типа дефектов и необходимости их исключения на наиболее ранних этапах разработки при минимальных затратах.

Любые испытания ограничены допустимым количеством и объемом тестирования, а также длительностью работы комиссии испытателей, поэтому не могут гарантировать абсолютную проверку программного продукта соответствию требованиям функций и характеристик. Для повышения достоверности определения и улучшения оценивания характеристик после внутренних или квалификационных испытаний некоторые типы комплексов программ целесообразно передавать на опытную эксплуатацию в типовых условиях. Это позволит более глубоко оценить эксплуатационные характеристики созданного комплекса и устранить некоторые дефекты и ошибки. Опытную эксплуатацию целесообразно проводить разработчиками с участием испытателей-заказчиков и некоторых пользователей, назначаемых заказчиком. Результаты и характеристики качества опытной эксплуатации после внутренних испытаний могут учитываться при проведении заказчиком квалификационных испытаний для их сокращения.

Для заказчика и пользователей доминирующее значение могут иметь номенклатура и особенности реализации некоторых отдельных функций комплекса программ, которые, как правило, требуют наибольших затрат на тестирование и определяют основной эффект от применения программного продукта, а также потенциальный уровень спроса на рынке. Если затраты на разработку комплекса программ можно оценивать и прогнозировать с некоторой достоверностью, то эффективность применения, и особенно будущий спрос на конкретную версию программного продукта со стороны пользователей, заранее оценить трудно. Такие оценки могут проводиться на основе специальных маркетинговых исследований и опыта эксплуатации аналогичных комплексов программ или достаточно близких их прототипов.