

# Испытания характеристик качества программного продукта

В составе требований к системам и программным продуктам, функционирующим в реальном времени, особое значение имеют динамические функции и характеристики. Для обеспечения их высокого качества недостаточны отдельные сценарии и процедуры тестов; необходимо создавать потоки динамических тестов в реальном времени, адекватные соответствующим данным при функционировании внешней среды систем и (или) пользователей. Эти потоки тестов должны обеспечивать динамическую проверку комплексов программ на соответствие требованиям, выявление дефектов и ошибок в реализации их функций и характеристик в реальном времени. Основная особенность такого тестирования заключается в создании динамической среды функционирования программного продукта, максимально приближенной к реальной, при его практическом применении.

Задача заключается в определении соответствия требованиям функций и характеристик программного продукта при различной интенсивности потоков тестов, адекватных нормальным условиям применения программного продукта, а также критическим по составу и интенсивности для выявления предельных условий его работоспособности. Такие условия тестирования отражаются на интегральных характеристиках, снижении надежности и (или) безопасности, а также на повышении рисков применения программного продукта. Для комплексов программ реального времени особое значение могут иметь причины и методы уменьшения рисков надежности и производительности вследствие дефектов и ошибок, а также при формировании и реализации требований к этим характеристикам программных продуктов. Эти взаимосвязанные характеристики качества программного продукта зависят от одних и тех же свойств воздействий из внешней среды, требуют совместного анализа и методов для выявления и устранения дефектов. Локализация и устранение таких динамических дефектов обычно осуществляется вне реального времени, путем применения детерминированных сценариев и тестовых процедур, а иногда за счет изменения требований заказчика.

Оценивание надежности программных комплексов включает в себя измерение количественных характеристик: завершенности, устойчивости к дефектам, восстанавливаемости и доступности-готовности. При этом предполагается, что в контракте, техническом задании или спецификации требований зафиксированы и утвержде-

ны заказчиком определенные значения этих характеристик и их приоритеты. Измерения проводятся при функционировании готового программного продукта для сопоставления с заданными требованиями и оценивания рисков соответствия этим спецификациями требований. Тестирование для оценки надежности комплекса программ должно проводиться в тестовом окружении, которое максимально приближено к реальным условиям применения системы. Входные параметры тестов следует задавать на основе вероятностного распределения соответствующих характеристик или их наборов при эксплуатации программного продукта исходя из частоты возможных сценариев работы пользователей или системы.

Значения надежности коррелированы с характеристикой «корректность», однако можно достигнуть высокой надежности функционирования комплекса программ при их невысокой корректности за счет сокращения времени восстановления при отказах. Кроме того, надежность можно оценивать косвенно в процессе разработки по прогнозируемой плотности обнаружения скрытых дефектов и ошибок, а также по плотности выявляемых и устраняемых ошибок выходных результатов при тестировании динамического функционирования комплекса программ. Степень покрытия тестами структуры функциональных компонентов и комплекса программ в целом может служить ориентиром для прогнозирования их потенциальной надежности. Распределение реальных длительностей и эффективности восстановления при ограниченных ресурсах для функционирования программ может рассматриваться как дополнительная составляющая при оценивании надежности.

Для прямых, количественных измерений надежности необходимы инструментальные средства, встроенные в операционную систему или в соответствующие компоненты комплекса программ. Эти средства должны в динамике функционирования программного продукта автоматически выявлять и регистрировать аномальные ситуации, дефекты и искажения вычислительного процесса, программ и данных, выявляемые аппаратным, программно-алгоритмическим контролем или пользователями. Накопление и систематизация проявлений дефектов при исполнении программ позволяет оценивать основные показатели надежности, помогает определять причины сбоев и отказов и подготавливать данные для повышения надежности программных продуктов.

Прямые экспериментальные методы оценивания интегральных характеристик надежности (безопасности и рисков) программных продуктов в ряде случаев трудно реализовать при нормальных штатных условиях функционирования крупных ком-

плексов программ из-за больших значений времени наработки на отказ (сотни и тысячи часов), которых необходимо достигать при разработке и фиксировать при испытаниях. Сложность выявления и регистрации редких отказов, а также высокая стоимость экспериментов при длительном многосуточном функционировании крупных комплексов программ приводят к тому, что на испытаниях получаются малые выборки зарегистрированных отказов и низкая достоверность оценки надежности. Кроме того, при таких экспериментах трудно гарантировать полную представительность выборки исходных данных, так как проверки определяются конкретными условиями применения данного программного продукта на испытаниях.

При испытаниях надежности в первую очередь обнаруживаются отказы — потери работоспособности. Однако в большинстве случаев первоначально остается неизвестной причина происшедшего отказа. Для выявления фактора, вызвавшего отказ (первичной ошибки или дефекта) и устранения его причины, прежде всего необходимо определить, каким компонентом системы стимулирован данный отказ. Наиболее крупными источниками отказов являются частичные физические неисправности или сбои аппаратуры ЭВМ, а также дефекты и ошибки программных продуктов.

Стабильные неисправности аппаратуры диагностируются достаточно просто, соответствующими аппаратными тестами, после чего должен следовать ремонт или замена определенных блоков. Однако при возникновении случайного отказа, после которого происходит автоматически полное восстановление нормального функционирования, во многих случаях трудно однозначно выявить его первичный источник, особенно при очень редких отказах.

Для диагностики и устранения случайных редких отказов должна быть организована служба их регистрации с максимально полным фиксированием характеристик ситуаций, при которых проявился каждый отказ. Сбои в аппаратуре носят более или менее случайный характер, и полное повторение ситуации отказа маловероятно. Ошибки и дефекты программ содержатся в определенном месте и должны регулярно проявляться при полном повторении внешних ситуаций. На основе таких признаков и, по возможности, детального описания ситуаций возникновения отказа могут строиться предположения о его причине. Для повышения надежности при высокой наработке на отказ необходима тщательная систематическая работа специалистов, накапливающих, регистрирующих и анализирующих все отказы при функционировании комплекса программ. Эти специалисты должны также регистрировать все проведенные корректировки для прогнозирования причин появления возможных дополнительных источников отказов, вызванных дефектами корректировок.

Для выявления тенденции изменения показателей надежности их зарегистрированные значения необходимо связывать во времени с моментами корректировки программ. Анализируя корреляцию между значениями надежности и процессом изменения программ, можно выявлять некоторые корректировки, которые содержат ошибки и снижают надежность. Получающиеся при этом показатели позволяют прогнозировать число ошибок, подлежащих исправлению, для достижения требуемых значений надежности в зависимости от длительности испытаний. В результате может быть оценена наработка до следующего выявления ошибки или отказа.

При заключительных приемо-сдаточных и сертификационных испытаниях для достоверного определения надежности организуются многочасовые и многосуточные прогоны динамического функционирования комплекса программ в реальной и (или) имитированной внешней среде в условиях широкого варьирования исходных данных с акцентом на стрессовые ситуации, стимулирующие проявления угроз надежности. Такие прогоны позволяют измерять достигнутые характеристики надежности и определять степень их соответствия требованиям технического задания, а также закреплять их в технических условиях и документации на программный продукт.

Если интенсивное тестирование программ в течение достаточно длительного времени не приводит к обнаружению дефектов или ошибок, то у специалистов, ведущих испытания, создается ощущение бесполезности дальнейшего тестирования программного продукта и он передается на эксплуатацию. Экспериментальное исследование характеристик сложных ПС позволило оценить темп обнаружения дефектов (риски), при котором крупные программные продукты передаются на регулярную эксплуатацию, — 0,002 ... 0,005 дефектов в день на человека, т. е. специалисты по испытаниям или все пользователи в совокупности выявляют только около одной ошибки (или дефекта) каждые два-три месяца использования ПС.

Интенсивность обнаружения ошибок — ниже 0,001 ошибок в день на человека, т. е. меньше одной ошибки в год на трех-четырех специалистов, непосредственно выполняющих динамическое квалификационное тестирование и (или) эксплуатацию комплекса программ (по-видимому, это может служить эталоном высокой надежности обработки информации). Если динамическое функционирование программного продукта происходит непрерывно, то эти показатели соответствуют высокой наработке на обнаружение дефекта или отказа около 5 ... 10 тыс. ч и коэффициенту готовности выше 0,99. При использовании этого критерия обычно учитывается календарное время испытаний, включающее в себя длительность непосредственного

тестирования, как для обнаружения, так и для локализации дефектов, а также длительность корректировки программ и других вспомогательных работ для восстановления нормального функционирования программного продукта.

Форсированные (стрессовые) испытания для оценивания надежности, а также функциональной безопасности и рисков программных продуктов значительно отличаются от традиционных методов испытаний аппаратуры. Основными факторами, влияющими на надежность ПС, являются исходные данные и их взаимодействие с дефектами и ошибками программ или сбоями в аппаратуре ЭВМ. Поэтому форсирования испытаний надежности добиваются повышением интенсивности искажений исходных данных и расширением варьирования их значений, а также специальным увеличением интенсивности потоков информации и загрузки программ на ЭВМ выше нормальной.

Планирование форсированных динамических испытаний должно предусматривать последующий пересчет полученных значений надежности на условия нормального функционирования. Для этого необходимо оценивать надежность испытываемых программ в зависимости от интенсивности искажений данных или характеристик перегрузки ЭВМ, а также применять способы корректного пересчета получаемых показателей на нормальные условия эксплуатации. При форсированных испытаниях целесообразно выделять следующие режимы тестирования:

- полное искажение, предельные и критические значения ключевых параметров тестов каждого типа внешней информации и воздействий пользователей;
- предельные и критические сочетания значений различных взаимодействующих параметров тестов при эксплуатации программного продукта;
- предельно большие и малые интенсивности суммарного потока и каждого типа внешней информации;
- умышленные нарушения пользователями определенных положений инструкций и рекомендаций эксплуатационной документации на программный продукт.

Как вид форсированных испытаний можно рассматривать тестирование и контроль результатов функционирования одних и тех же ПС при увеличении числа испытываемых экземпляров и нормальных исходных данных — бета-тестирование. На этапе тестирования, в соответствии с эксплуатационной документацией, пользователями некоторого предварительного тиража программного продукта происходит естественное расширение вариантов исходных данных, если они взаимно незави-

симы. Это увеличивает наборы тестов и тем самым дает возможность оценивать наработки на отказ в сотни и тысячи часов. Они позволяют выявлять и устранять значительное число дефектов за небольшое календарное время и тем самым доводить надежность до требуемого уровня. Однако следует учитывать, что при этом пропорционально возрастает суммарная трудоемкость таких испытаний.

Особым видом форсированных испытаний является целенаправленное тестирование эффективности средств оперативного контроля и восстановления программ, данных и вычислительного процесса для оценивания восстанавливаемости. При таких испытаниях основная задача заключается в оценивании качества динамического функционирования средств автоматического повышения надежности и измерении характеристик восстанавливаемости. Для этого имитируются запланированные условия функционирования программ, при которых в наибольшей степени стимулируется срабатывание средств программного рестарта и оперативного автоматического восстановления работоспособности.

Следует особо отметить трудности достижения и регистрации надежности программ, характеризующейся наработкой на отказ более 100 ч. При такой надежности резко возрастают сложность обнаружения возникающих отказов и диагностирования их причин. Нарботка на отказ в тысячи часов в ряде случаев достигалась только при эксплуатации и сопровождении сложных программных продуктов в течение нескольких лет. При требовании особо высокой надежности функционирования суммарные затраты ресурсов на ее достижение и оценивание могут увеличиваться на порядок, однако требуемое увеличение затрат для получения такой высокой надежности программного продукта в процессе разработки трудно обеспечить практически. Поэтому для ее достижения активно применяются различные методы программной защиты от сбоев и отказов программ (методы оперативного рестарта). Они позволяют замедлить рост затрат ресурсов на разработку при повышении требований к их надежности.

Изменение надежности комплексов программ реального времени можно оценивать путем применения оперативного контроля и рестарта. В любых ситуациях функционирования сложных комплексов программ прежде всего должны исключаться катастрофические последствия и длительные отказы или в максимальной степени должно смягчаться их негативное влияние на результаты, выдаваемые пользователю. Неизбежность дефектов и ошибок в сложных комплексах программ, искажений исходных данных и аппаратурных сбоев приводит к необходимости регулярного контроля процесса исполнения комплекса программ и сохранности данных.

Предвидеть заранее все ситуации исполнения программ и протестировать при них крупные программные продукты невозможно из-за их огромного количества, поэтому применяются методы оперативного обнаружения последствий дефектов и аномального функционирования программ, а также автоматического восстановления (рестарт) нормального вычислительного процесса и искаженных текстов программ и данных. Для обеспечения высокой надежности функционирования необходимо максимально быстро обнаруживать аномалии, достаточно точно классифицировать тип уже имеющихся и возможных последствий искажений, а также осуществлять мероприятия, обеспечивающие быстрое восстановление нормального функционирования программного продукта и системы.

Для снижения влияния негативных возмущений различных типов и происхождения на результаты, а также для защиты вычислительного процесса и информации программно-алгоритмическими методами в комплексах программ реального времени должна иметься избыточность ресурсов. Избыточность вычислительных ресурсов необходима, прежде всего, для селекции оперативных искажений процесса функционирования программного продукта и выработки решений по снижению последствий этих аномалий.

Основная цель использования избыточности заключается в ограничении или исключении возможности аварийных последствий от динамических возмущений, соответствующих отказу системы. Любые дефекты при исполнении программ необходимо блокировать и по возможным последствиям сводить до уровня сбоя путем оперативного автоматического восстановления. При этом необязательно сразу устанавливать причины искажения, главная задача сводится к максимально быстрому восстановлению нормального функционирования и ограничению его негативных последствий.

Введение средств контроля функционирования и помехозащиты в программы позволяет скомпенсировать влияние на надежность неполной корректности программных продуктов, а также снизить негативные воздействия внешних возмущений различных типов. Однако только средствами контроля и обеспечения программной помехозащиты невозможно достигнуть высокой надежности динамического функционирования ПС. Требуемое сокращение вероятности отказа может достигаться путем повышения затрат ресурсов на тестирование и увеличения его длительности, что допускает соответствующее снижение затрат на средства обнаружения искажений и восстановление.

Контроль работоспособности комплекса программ, исправление дефектов и восстановление при возникновении отказов сокращают ресурсы времени функционирования ЭВМ, доступные для выполнения основных функций, и в общем случае негативно отражаются на производительности системы. Однако если некоторые ситуации контроля и восстановления проводятся достаточно быстро (так, что их последствия не фиксируются как отказ и не отражаются на снижении работоспособности), то такие затраты времени полезны для решения функциональных задач и должны быть отнесены к улучшению качества программного продукта.

Функциональная безопасность программных продуктов и систем зависит от отказов, негативно отражающихся на работоспособности и реализации их основных функций, причинами которых могут быть дефекты и аномалии в аппаратуре, комплексах программ, данных или вычислительных процессах. При этом катастрофически, критически или существенно искажается процесс функционирования программных продуктов и (или) систем, что наносит значительный ущерб при их применении.

Основными источниками отказов могут быть некорректные исходные требования, сбои и отказы в аппаратуре, дефекты или ошибки в программах и данных функциональных задач, проявляющиеся при их динамическом исполнении в соответствии с назначением. При таких воздействиях внешняя, функциональная, работоспособность систем может разрушаться не полностью, однако невозможно полноценное выполнение заданных функций и требований к программному продукту. В реальных сложных системах, связанных с безопасностью, возможны катастрофические последствия и отказы функционирования с большим ущербом при отсутствии воздействия лиц, заинтересованных в нарушениях работоспособности систем и ПС.

Понятия, методы тестирования и характеристики функциональной безопасности программных продуктов и систем близки к аналогичным для надежности. Поэтому способы оценки и испытаний функциональной безопасности могут базироваться на методах тестирования, определения и обеспечения надежности функционирования комплексов программ. Чем сложнее системы и выше требования безопасности, предъявляемые к ним, тем неопределеннее функции и характеристики тестирования требований для обеспечения их безопасности.

Неопределенности начинаются с требований заказчиков, которые при формулировке технического задания и спецификаций не полностью формализуют и принципиально не могут обеспечить достоверное содержание всего адекватного набора



характеристик и значений требований безопасности, которые должны быть при завершении проекта и предъявлении конечного программного продукта заказчику. Эти требования итерационно формируются, детализируются и уточняются по согласованию между всеми участниками проекта вследствие естественной ограниченности первичных исходных данных и их изменения под влиянием объективных и субъективных воздействий со стороны различных процессов на последовательных этапах жизненного цикла комплекса программ.

Всегда не полностью, с необходимой детализацией определяются и описываются все характеристики, особенности функционирования и безопасности объектов внешней среды. Квалификация и субъективные свойства потребителей и пользователей изменяются по мере освоения функциональных возможностей системы и ее работоспособности, что увеличивает неопределенность ее реальной безопасности. Различия свойств персонала, применяющего систему, дополнительно увеличивают неопределенность значений безопасности и трудности ее прогнозирования при тестировании с учетом множества субъективных факторов различных специалистов, участвующих в эксплуатации.

При анализе характеристик функциональной безопасности целесообразно выделять и учитывать особенности классов систем и их программных продуктов. *Первый класс* составляют системы, имеющие встроенные комплексы программ жесткого регламента реального времени, автоматизированно управляющие внешними объектами или процессами. Время необходимой реакции на отказ таких систем обычно исчисляется секундами или долями секунды. Процессы восстановления работоспособности должны проходить за это время в достаточной степени автоматизированно (бортовые системы в авиации, на транспорте, в некоторых средствах вооружения, системы управления атомными электростанциями).

Системы *второго класса* применяются для управления процессами и обработки деловой информации из внешней среды, в которых активно участвуют специалисты-операторы (банковские, административные, штабные военные системы). Допустимое время реакции на опасные отказы в этих системах может составлять десятки секунд и минуты, и операции по восстановлению работоспособности частично могут быть доверены специалистам-администраторам по обеспечению функциональной безопасности.

Тестирование и обеспечение функциональной безопасности сложных систем должны решаться с учетом одновременного динамического развития всех компонентов внешней среды и факторов, непрерывно изменяющихся и воздействующих

на результаты их решения. Эти факторы влияют на неопределенность критериев, методов оценивания значений эффективности тестирования и функциональной безопасности конкретных программных продуктов и систем.

Существующие технологии тестирования способствуют повышению функциональной безопасности, снижению потенциального ущерба и рисков, однако практически всегда остается открытым вопрос, насколько применяемые методы оправдывают затраты на реализацию требований заказчика. Испытания, эксплуатация и сертификация способствуют снижению неопределенности оценок эффективности и итерационному приближению к практически приемлемому уровню функциональной безопасности программных продуктов.

Роль негативных воздействий и их разрушительные последствия быстро усиливаются с ростом сложности разработки и применения современных систем на базе ЭВМ и степени ответственности решаемых ими задач. Одновременно возрастают сложность внешней и операционной среды, в которой функционируют комплексы программ, и ответственность функций систем, связанных с безопасностью. Объективное повышение сложности функций, реализуемых программными продуктами в современных системах, приводит к увеличению их объема и трудоемкости создания. В соответствии с ростом сложности программ увеличивается относительное и абсолютное количество выявляемых и остающихся в них дефектов и ошибок, что отражается на снижении потенциальной безопасности их функционирования.

Работоспособность программных продуктов может быть обеспечена при исходных данных, которые использовались при их разработке, отладке и испытаниях. Реальные исходные данные могут иметь значения, отличающиеся от заданных техническим заданием и используемых при эксплуатации программ и баз данных. При таких исходных данных функционирование программного продукта трудно предсказать заранее и весьма вероятны различные аномалии, завершающиеся отказами, отражающимися на безопасности. Это приводит к практической невозможности достоверных априорных аналитических оценок функциональной безопасности комплексов программ при ее высоких значениях.

Требуемая функциональная безопасность систем, содержащих программные продукты реального времени, достигается путем использования современных регламентированных технологических процессов динамического тестирования, подобных применяемым при обеспечении надежности. Они должны быть поддержаны группой международных стандартов, определяющих состав и процессы выполнения

требований к заданной функциональной безопасности систем и комплексов программ. Для систематической, координированной борьбы с угрозами безопасности программных продуктов необходимы исследования факторов, влияющих на функциональную безопасность со стороны случайных дефектов и ошибок, существующих и потенциально возможных в конкретных системах и комплексах программ.

Требования к функциям систем и программных продуктов, а также к безопасности их функционирования должны соответствовать доступным ресурсам для их реализации с учетом допустимого ущерба — рисков вследствие отказов при неполном выполнении требований. Ограниченность ресурсов различных видов для тестирования и обеспечения функциональной безопасности значительно влияет на технико-экономические показатели, качество и функциональную безопасность всей системы и ПС. В результате сложность комплексов программ, а также доступные ресурсы для их реализации становятся косвенными критериями или факторами, влияющими на выбор методов разработки, достигаемую безопасность программных продуктов.

Разработку систем должны завершать сертификационные испытания и удостоверение достигнутой функциональной безопасности и надежности систем с программным продуктом, предусматривающие возможность совершенствования их характеристик путем соответствующих корректировок программ. Повысить функциональную безопасность можно также путем анализа выявленных дефектов и оперативного восстановления вычислительного процесса, программ и данных (рестарта) после обнаружения аномалий и отказов функционирования программного продукта. Этому может способствовать накопление, мониторинг и хранение данных о выявленных дефектах, сбоях и отказах в процессе исполнения программ и обработки данных.

Оценивание ресурсной эффективности заключается в измерении количественных характеристик: временной эффективности и используемости динамических ресурсов ЭВМ комплексом программ. При этом предполагается, что в контракте, техническом задании и спецификации требований зафиксированы и утверждены требуемые значения этих характеристик и их приоритетов. Оценивание динамических характеристик программ в реальном времени может проводиться при функционировании готового программного продукта или расчетными методами, при разработке для сопоставления с заданными требованиями и оценки степени соответствия этим требованиям.

Цель испытаний производительности — продемонстрировать заказчику, что в реальном времени система функционирует в соответствии с требованиями, содержащимися в спецификациях на производительность и касающимися приемлемого времени отклика при обработке заданного количества транзакций. При тестировании производительности в реальном времени должны применяться промышленные нагрузки. Это позволит предсказать поведение программного продукта и системы при реальной эксплуатации. Средства, обеспечивающее тестирование производительности, должны позволять оценивать влияние перегрузок.

**Оценивание перегрузок** — процесс тестирования работоспособности вычислительных машин при выполнении динамических сценариев большого потока данных в реальном времени в целях определения того, когда и где программный продукт выйдет из строя, работая под высокой нагрузкой.

При тестировании перегрузок система подвергается предельным и максимальным нагрузкам для определения, выйдет ли она из строя и где это произойдет, а также для идентификации того, что выйдет из строя. Эти допустимые пределы должны быть определены в системных требованиях к программному продукту, где также должна быть определена реакция системы на перегрузки. Данный вид тестирования необходим для систем, работающих с максимальной спроектированной нагрузкой, для проверки того, что они в реальном времени динамически функционируют в соответствии с требованиями.

Адекватное проведение динамического испытания программного продукта на перегрузки при использовании ручных методов подготовки тестов дорого, трудоемко, неточно и занимает много времени. Необходимы средства автоматизированного тестирования, которые включают в себя имитаторы нагрузки, позволяющие испытателям динамически имитировать в реальном времени сотни и тысячи виртуальных пользователей или объектов, одновременно работающих с целевым программным продуктом. Не нужно никому присутствовать лично, чтобы запустить тесты или управлять ими; можно установить таймер, определив, когда следует запустить конкретный тест, и они могут выполняться без участия человека.

Для измерения характеристик временной эффективности необходимы инструментальные средства, встроенные в операционную систему или в соответствующий комплекс программ. Эти средства должны в динамике реального функционирования программного продукта регистрировать:

- загрузку вычислительной системы функционирующими программами;

- значения интенсивности потоков данных от конкретных внешних абонентов;
- длительность исполнения заданий при реализации конкретных функций;
- характеристики функционирования устройств ввода-вывода;
- время ожидания результатов (отклика) на функциональные задания пользователей или системы;
- заполнение памяти данными обмена с внешними абонентами в различных режимах применения программного продукта.

Значения этих характеристик зависят не только от свойств и функций комплекса программ, но и от особенностей архитектуры и операционной системы ЭВМ. Регулярная регистрация и обобщение таких данных позволяют выявлять ситуации, негативно влияющие в реальном времени на функциональную пригодность, надежность и другие важные характеристики программного продукта.

Существует особый вид тестов для проверки удовлетворения специфических требований, предъявляемых к параметрам производительности программного продукта, когда делается попытка достижения количественных пределов, обусловленных характеристиками самой системы и ее операционного окружения. При этом может производиться динамическое тестирование в реальном времени в целях достижения наибольших возможностей по производительности и выполнения функций программного продукта с повышением нагрузки, вплоть до достижения запланированных характеристик требований.

При очень высокой интенсивности поступления исходных данных может нарушаться временной баланс между длительностью решения требуемой совокупности задач программным продуктом в реальном времени и производительностью ЭВМ при решении этих задач. Также возможно нарушение баланса между имеющейся в ЭВМ памятью и памятью, необходимой для хранения всей поступившей и обрабатываемой информации. Для выявления подобных ситуаций и определения характеристик программного продукта в условиях недостаточности ресурсов ЭВМ проводятся испытания при высокой, но допустимой (в соответствии с требованиями) интенсивности поступления исходных данных.

Наиболее сложно оценивать эффективность использования ресурсов производительности ЭВМ в реальном времени. При этом должна быть определена зависимость качества решения задач от интенсивности поступающей информации различных типов. Основная задача испытаний заключается в определении рисков — вероятностей, с которыми будет нарушаться соответствие между потребностями

в производительности для решения всей требуемой совокупности задач и реальными возможностями ЭВМ и других компонентов системы. Если эта вероятность невелика и можно считать допустимым эпизодическое снижение качества за счет получающихся задержек и пропусков в обработке сообщений или заданий, то делается вывод о соответствии производительности ЭВМ заданным функциям данного программного продукта.

Для оценивания характеристик использования производительности при тестировании крупных программных продуктов в реальном времени должны быть измерены:

- реальные значения интенсивностей поступающих исходных данных и заданий на вызов функциональных программ, а также распределения вероятностей этих интенсивностей для различных источников и типов заданий;
- длительности автономного решения отдельно каждой функциональной задачи, обрабатывающей исходные данные или включаемой внешними заданиями, а также периодически;
- загрузка ЭВМ в нормальном режиме поступления сообщений и заданий, а также вероятность перегрузки заданиями различных типов и возможные распределения длительностей перегрузки в реальных условиях;
- влияние пропуска в обработке заданий или сообщений каждого типа и снижения темпа решения определенных задач на функциональную пригодность и другие важные характеристики программного продукта.

Перечисленные задачи могут быть решены экспериментально в процессе тестирования в реальном времени завершеного разработкой программного продукта, однако при этом велик риск, что производительность ЭВМ окажется недостаточной для решения заданной совокупности задач в реальном времени, что отразится на качестве и возможности использования системы. Кроме того, не всегда условия испытаний или опытной эксплуатации системы соответствуют режимам массового ее применения. Поэтому при оценивании требуется принимать специальные меры для создания реальных, а также контролируемых, наиболее тяжелых по загрузке, в реальном времени условий функционирования комплекса программ и внешней среды.

Для корректного оценивания предельной пропускной способности системы в реальном времени с данным программным продуктом необходимо измерять следующие характеристики реализации функциональных программ в процессе разработки:

- экстремальные значения длительностей их исполнения и маршруты, на которых эти значения достигаются;
- среднее значение длительности исполнения каждой функциональной группы программ на всем возможном множестве маршрутов и его дисперсию;
- распределение вероятностей и значений длительности исполнения функциональных групп программ.

В общем случае для оценивания длительностей исполнения и определения качества динамического функционирования программ в зависимости от загрузки необходимо оценивать вероятность каждой комбинации тестовых данных и измерять соответствующую ей длительность исполнения компонента программы. После упорядочения значений длительностей можно получить распределение вероятностей в зависимости от длительностей исполнения. Влияние таких ситуаций перегрузки ЭВМ по производительности может быть ослаблено в реальном времени путем применения приоритетных дисциплин оперативной диспетчеризации исполнения заданий на решение функциональных задач.

Достоверность оценивания пропускной способности системы в реальном времени с конкретным программным продуктом зависит от корректности моделирования потоков внешних сообщений, а также от используемых распределений длительности исполнения программ. Для оценивания ресурсной эффективности при подготовке технического задания и спецификаций требований следует согласовывать с заказчиком модель и динамические характеристики внешней среды, в которой будет применяться комплекс программ, а также динамику приема и передачи данных.

Для определения использования комплексами программ ресурсов ЭВМ в реальном времени полезно применять рекомендации стандарта ISO/IEC 14756:1999, который ориентирован на динамическое оценивание программных продуктов, операционных систем и вычислительных комплексов, включающих в себя аппаратные и программные средства. Описание метода измерения производительности начинается с имитации пользователей и потоков данных из внешней среды: их случайных временных характеристик и процессов; функционирования терминалов; установления параметров рабочих нагрузок пользователей и вычислительных средств.

Оценивание производительности в реальном времени рекомендуется для определения загрузки операторов-пользователей, пропускной способности программных продуктов по числу задач в единицу времени, временной шкалы событий обработки

заданий и данных. Эти результаты предлагается сравнивать с требованиями заказчика и пользователей для оценивания допустимых рабочих нагрузок и достаточности производительности в реальном времени в конкретной внешней среде.

Детальные процедуры измерений и оценивания в стандарте подразделены на шесть разделов:

- исходные требования;
- процессы измерений;
- результирующие данные;
- проверка корректности результатов;
- расчеты производительности;
- оценивание достоверности измерений производительности.

Если предварительно в процессе проектирования производительность системы в реальном времени не оценивалась или определялась слишком грубо, то велик риск, что доработки будут большими или может потребоваться замена ЭВМ на более быстродействующую. Это обусловлено, как правило, оптимизмом разработчиков, что приводит к занижению интуитивных оценок длительностей решения функциональных задач и возможных предельных интенсивностей потоков внешней информации.

Длительная регистрация и накопление значений ресурсной эффективности способствуют выявлению ситуаций, при которых проявляются некоторые динамические дефекты функциональной пригодности.