ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 56920— 2024

Системная и программная инженерия

ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Общие положения

(ISO/IEC/IEEE 29119-1:2022, NEQ)

Издание официальное

Москва
Российский институт стандартизации
2024

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Обществом с ограниченной ответственностью «Информационно-аналитический вычислительный центр» (ООО ИАВЦ)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 22 «Информационные технологии»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 10 июня 2024 г. № 752-ст

4 Настоящий стандарт разработан с учетом основных нормативных положений международного документа ISO/IEC/IEEE 29119-1:2022 «Системная и программная инженерия. Тестирование программного обеспечения. Часть 1. Общие понятия» (ISO/IEC/IEEE 29119-1:2022 «Software and system engineering — Software testing — Part 1: General concepts», NEQ)

5 ВЗАМЕН ГОСТ Р 56920—2016/ISO/IEC/IEEE 29119-1:2013

*Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Оформление. ФГБУ «Институт стандартизации», 2024

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Термины и определения 4
4 Сокращения 7
5 Общие положения 7
5.1 Роль тестирования программного обеспечения 7
5.2 Цели тестирования 8
5.3 Принципы тестирования 8
5.4 Жизненный цикл тестирования 9
5.5 Анализ требований и определение стратегии тестирования 10
5.6 Планы тестирования 14
5.7 Среда тестирования 14
5.8 Процессы тестирования 15
5.9 Проведение тестирования 18
5.10 Документирование 21
5.11 Управление инцидентами 21
5.12 Использование тестирования при верификации и валидации 21
Приложение А (справочное) Примеры учета особенностей при тестировании 22
Приложение Б (справочное) Пример возможных участников тестирования 26
Библиография 27

Введение

Целью настоящего стандарта является определение общих положений по тестированию программного обеспечения (ПО) в жизненном цикле таких систем, как автоматизированные системы, информационно-вычислительные системы, устройства и системы искусственного интеллекта, включая киберфизические системы.

В настоящем стандарте представлена роль тестирования ПО в управлении качеством, сформулированы цели и принципы тестирования, рассмотрен жизненный цикл тестирования, определены основные положения по анализу требований и стратегии тестирования, формированию планов и среды тестирования, рассмотрены процессы и методы тестирования, изложены общие положения по проведению тестирования, документированию, управлению инцидентами и использованию тестирования при верификации и валидации систем.

В приложении А кратко описаны примеры учета особенностей при тестировании ПО.

В приложении Б представлены основные участники тестирования ПО.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Системная и программная инженерия

ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Общие положения

System and software engineering. Software testing. General provisions

Дата введения — 2024—09—30

1 Область применения

Настоящий стандарт определяет общие положения по тестированию программного обеспечения (ПО) для различных автоматизированных систем (АС), информационно-вычислительных систем (ИВС), устройств и систем искусственного интеллекта (СИИ), включая киберфизические системы (далее по тексту — систем).

Настоящий стандарт предназначен для использования организациями, участвующими в создании (модернизации, развитии), эксплуатации систем и их ПО, а также при выведении их из эксплуатации.

Примечание — Настоящий стандарт не содержит деталей оценки, способов демонстрации и всестороннего анализа различных свойств качества и безопасности систем и их ПО. Эти вопросы являются предметом более специализированных стандартов, часть из которых отражена в разделе 2.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ 34.201 Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем

ГОСТ 34.602 Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы

ГОСТ 28806 Качество программных средств. Термины и определения

ГОСТ IEC 61508-3 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 3. Требования к программному обеспечению

ГОСТ Р 2.601 Единая система конструкторской документации. Эксплуатационные документы

ГОСТ Р 22.10.01 Безопасность в чрезвычайных ситуациях. Оценка ущерба. Термины и определения

ГОСТ Р 27.101 Надежность в технике. Надежность выполнения задания и управление непрерывностью деятельности. Термины и определения

ГОСТ Р 27.102 Надежность в технике. Надежность объекта. Термины и определения

ГОСТ Р 51583 Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. Общие положения

ГОСТ Р 51897/Руководство ИСО 73:2009 Менеджмент риска. Термины и определения

ГОСТ Р 51901.1 Менеджмент риска. Анализ риска технологических систем

ГОСТ Р 51901.5 (МЭК 60300-3-1:2003) Менеджмент риска. Руководство по применению методов анализа надежности

ГОСТ Р 51901.7/ISO/TR 31004:2013 Менеджмент риска. Руководство по внедрению ИСО 31000

Издание официальное

ГОСТ Р 51901.16 (МЭК 61164:2004) Менеджмент риска. Повышение надежности. Статистические критерии и методы оценки

ГОСТ Р 51904 Программное обеспечение встроенных систем. Общие требования к разработке и документированию

ГОСТ Р 53622 Информационные технологии. Информационно-вычислительные системы. Стадии и этапы жизненного цикла, виды и комплектность документов

ГОСТ Р 53647.1 Менеджмент непрерывности бизнеса. Часть 1. Практическое руководство

ГОСТ Р 54124 Безопасность машин и оборудования. Оценка риска

ГОСТ Р 54145 Менеджмент рисков. Руководство по применению организационных мер безопасности и оценки рисков. Общая методология

ГОСТ Р 56921 Системная и программная инженерия. Тестирование программного обеспечения. Часть 2. Процессы тестирования

ГОСТ Р 56922 Системная и программная инженерия. Тестирование программного обеспечения. Часть 3. Документация тестирования

ГОСТ Р 56939 Защита информации. Разработка безопасного программного обеспечения. Общие требования

ГОСТ Р 57102/ISO/IEC TR 24748-2:2011 Информационные технологии. Системная и программная инженерия. Управление жизненным циклом. Часть 2. Руководство по применению ИСО/МЭК 15288

ГОСТ Р 57193 Системная и программная инженерия. Процессы жизненного цикла систем

ГОСТ Р 57272.1 Менеджмент риска применения новых технологий. Часть 1. Общие требования

ГОСТ Р 57839 Производственные услуги. Системы безопасности технические. Задание на проектирование. Общие требования

ГОСТ Р 58412 Защита информации. Разработка безопасного программного обеспечения. Угрозы безопасности информации при разработке программного обеспечения

ГОСТ Р 58494 Оборудование горно-шахтное. Многофункциональные системы безопасности угольных шахт. Система дистанционного контроля опасных производственных объектов

ГОСТ Р 58771 Менеджмент риска. Технологии оценки риска

ГОСТ Р 59276 Системы искусственного интеллекта. Способы обеспечения доверия. Общие положения

ГОСТ Р 59277 Системы искусственного интеллекта. Классификация систем искусственного интеллекта

ГОСТ Р 59330 Системная инженерия. Защита информации в процессе управления моделью жизненного цикла системы

ГОСТ Р 59334 Системная инженерия. Защита информации в процессе управления качеством системы

ГОСТ Р 59335 Системная инженерия. Защита информации в процессе управления знаниями о системе

ГОСТ Р 59336 Системная инженерия. Защита информации в процессе планирования проекта

ГОСТ Р 59341 Системная инженерия. Защита информации в процессе управления информацией системы

ГОСТ Р 59342 Системная инженерия. Защита информации в процессе измерений системы

ГОСТ Р 59352 Системная инженерия. Защита информации в процессе верификации системы

ГОСТ Р 59354 Системная инженерия. Защита информации в процессе аттестации системы

ГОСТ Р 59356 Системная инженерия. Защита информации в процессе сопровождения системы

ГОСТ Р 59709 Защита информации. Управление компьютерными инцидентами. Термины и определения

ГОСТ Р 59710 Защита информации. Управление компьютерными инцидентами. Общие положения

ГОСТ Р 59711 Защита информации. Управление компьютерными инцидентами. Организация деятельности по управлению компьютерными инцидентами

ГОСТ Р 59712 Защита информации. Управление компьютерными инцидентами. Руководство по реагированию на компьютерные инциденты

ГОСТ Р 59793 Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания

ГОСТ Р 59795 Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов

ГОСТ Р 59853 Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения

ГОСТ Р 59898 Оценка качества систем искусственного интеллекта. Общие положения

ГОСТ Р 59989 Системная инженерия. Системный анализ процесса управления качеством системы

ГОСТ Р 59990 Системная инженерия. Системный анализ процесса оценки и контроля проекта

ГОСТ Р 59991 Системная инженерия. Системный анализ процесса управления рисками для системы

ГОСТ Р 59992 Системная инженерия. Системный анализ процесса управления моделью жизненного цикла системы

ГОСТ Р 59994 Системная инженерия. Системный анализ процесса гарантии качества для системы

ГОСТ Р 70250 Системы искусственного интеллекта на автомобильном транспорте. Варианты использования и состав функциональных подсистем искусственного интеллекта

ГОСТ Р 70462.1 Информационные технологии. Искусственный интеллект. Оценка робастности нейронных сетей. Часть 1. Обзор

ГОСТ Р 70921 Системная и программная инженерия. Требования и оценка качества систем и программной продукции (SQuaRE). Концепция требований к качеству

ГОСТ Р 71199 Системы киберфизические. Умный дом. Термины и определения

ГОСТ Р ИСО 9000 Системы менеджмента качества. Основные положения и словарь

ГОСТ Р ИСО 9001 Системы менеджмента качества. Требования

ГОСТ Р ИСО 14258 Промышленные автоматизированные системы. Концепции и правила для моделей предприятия

ГОСТ Р ИСО 15704 Моделирование и архитектура предприятия. Требования к стандартным архитектурам и методологиям предприятия

ГОСТ Р ИСО 17359 Контроль состояния и диагностика машин. Общее руководство

ГОСТ Р ИСО 31000 Менеджмент риска. Принципы и руководство

ГОСТ Р ИСО/МЭК 12207 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств

ГОСТ Р ИСО/МЭК 15026 Информационная технология. Уровни целостности систем и программных средств

ГОСТ Р ИСО/МЭК 15026-1[1)] Системная и программная инженерия. Гарантирование систем и программного обеспечения. Часть 1. Понятия и словарь

ГОСТ Р ИСО/МЭК 15026-4 Системная и программная инженерия. Гарантирование систем и программного обеспечения. Часть 4. Гарантии жизненного цикла

ГОСТ Р ИСО/МЭК 16085 Менеджмент риска. Применение в процессах жизненного цикла систем и программного обеспечения

ГОСТ Р ИСО/МЭК 25000 Системная и программная инженерия. Требования и оценка качества систем и программных средств (SQuaRE). Руководство

ГОСТ Р ИСО/МЭК 25010 Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов

ГОСТ Р ИСО/МЭК 25020 Системная и программная инженерия. Требования и оценка качества систем и программной продукции (SQuaRE). Основные принципы измерения качества

ГОСТ Р ИСО/МЭК 25023 Системная и программная инженерия. Требования и оценка качества систем и программной продукции (SQuaRE). Измерения качества системы и программной продукции

ГОСТ Р ИСО/МЭК 25040 Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Процесс оценки

ГОСТ Р ИСО/МЭК 27001 Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования

ГОСТ Р ИСО/МЭК 27002 Информационные технологии. Методы и средства обеспечения безопасности. Свод норм и правил применения мер обеспечения информационной безопасности

[1)] Одновременно с введением в действие настоящего стандарта планируется введение в действие взамен ГОСТ Р ИСО/МЭК 15026-1—2016 рекомендуемого к использованию ГОСТ Р 71304—2024 «Системная и программная инженерия. Гарантии обеспечения качества систем и программных средств. Часть 1. Общие положения».

ГОСТ Р ИСО/МЭК 27005 Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности

ГОСТ Р ИСО/МЭК 27036-4 Информационные технологии. Методы и средства обеспечения безопасности. Информационная безопасность во взаимоотношениях с поставщиками. Часть 4. Рекомендации по обеспечению безопасности облачных услуг

ГОСТ Р МЭК 61069-1 Измерение, управление и автоматизация промышленного процесса. Определение свойств системы с целью ее оценки. Часть 1. Терминология и общие концепции

ГОСТ Р МЭК 61069-2 Измерение, управление и автоматизация промышленного процесса. Определение свойств системы с целью ее оценки. Часть 2. Методология оценки

ГОСТ Р МЭК 61069-3 Измерение, управление и автоматизация промышленного процесса. Определение свойств системы с целью ее оценки. Часть 3. Оценка функциональности системы

ГОСТ Р МЭК 61069-4 Измерение, управление и автоматизация промышленного процесса. Определение свойств системы с целью ее оценки. Часть 4. Оценка производительности системы

ГОСТ Р МЭК 61069-5 Измерение, управление и автоматизация промышленного процесса. Определение свойств системы с целью ее оценки. Часть 5. Оценка надежности системы

ГОСТ Р МЭК 61069-6 Измерение, управление и автоматизация промышленного процесса. Определение свойств системы с целью ее оценки. Часть 6. Оценка эксплуатабельности системы

ГОСТ Р МЭК 61069-7 Измерение, управление и автоматизация промышленного процесса. Определение свойств системы с целью ее оценки. Часть 7. Оценка безопасности системы

ГОСТ Р МЭК 61069-8 Измерение, управление и автоматизация промышленного процесса. Определение свойств системы с целью ее оценки. Часть 8. Оценка других свойств системы

ГОСТ Р МЭК 61508-1 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 1. Общие требования

ГОСТ Р МЭК 61508-2 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 2. Требования к системам

ГОСТ Р МЭК 61508-4 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 4. Термины и определения

ГОСТ Р МЭК 61508-6 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 6. Руководство по применению ГОСТ Р МЭК 61508-2 и ГОСТ Р МЭК 61508-3

ГОСТ Р МЭК 61508-7 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 7. Методы и средства

ГОСТ Р МЭК 62264-1 Интеграция систем управления предприятием. Часть 1. Модели и терминология

ГОСТ Р МЭК 62508 Менеджмент риска. Анализ влияния на надежность человеческого фактора

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

3 Термины и определения

В настоящем стандарте применены термины по ГОСТ 28806, ГОСТ Р 27.101, ГОСТ Р 27.102, ГОСТ 34.201, ГОСТ Р 22.10.01, ГОСТ Р ИСО 9000, ГОСТ Р ИСО/МЭК 15026, ГОСТ Р ИСО/МЭК 15026-1, ГОСТ Р ИСО/МЭК 27001, ГОСТ Р 51897, ГОСТ Р 59709, ГОСТ Р 59989, ГОСТ Р 59991, ГОСТ Р 59994, ГОСТ Р МЭК 61069-1, ГОСТ Р МЭК 61508-4, ГОСТ Р МЭК 62264-1, ГОСТ Р 71199, а также следующие термины с соответствующими определениями:

3.1

автоматизированная система; АС: Система, состоящая из комплекса средств автоматизации, реализующего информационную технологию выполнения установленных функций, и персонала, обеспечивающего его функционирование.

Примечания

В зависимости от вида деятельности выделяют, например, следующие виды АС: автоматизированные системы управления (АСУ), системы автоматизированного проектирования (САПР), автоматизированные системы научных исследований (АСНИ) и др.

В зависимости от вида управляемого объекта (процесса) АСУ подразделяют, например на АСУ технологическими процессами (АСУТП), АСУ предприятиями (АСУП) и т. д.

[ГОСТ Р 59853—2021, статья 3.2]

3.2

дефект: Каждое отдельное несоответствие продукции установленным требованиям.
[ГОСТ 15467—1979, пункт 3.8]

3.3

информационно-вычислительная система (программно-технический комплекс): Совокупность данных (баз данных) и программ, функционирующих на вычислительных средствах как единое целое для решения определенных задач.
[ГОСТ Р 53622—2009, статья 3.5]

3.4

искусственный интеллект; ИИ: Способность технической системы имитировать когнитивные функции человека (включая самообучение и поиск решений без заранее заданного алгоритма) и получать при выполнении конкретных практически значимых задач обработки данных результаты, сопоставимые, как минимум, с результатами интеллектуальной деятельности человека.
[ГОСТ Р 59276—2020, статья 3.6]

3.5

киберфизическая система: Информационно-технологическая концепция, подразумевающая интеграцию вычислительных ресурсов в физические процессы. В такой системе датчики, оборудование и информационные системы соединены на протяжении всей цепочки в логику управления для создания стоимости, выходящей за рамки одного предприятия или бизнеса. Эти системы взаимодействуют друг с другом с помощью стандартных интернет-протоколов для прогнозирования, самонастройки и адаптации к изменениям.
[ГОСТ Р 59277—2020, статья 3.29]

3.6 **метаморфическая связь:** Описание влияния изменений во входных данных для тестового примера на ожидаемые выходные результаты на основе требуемого поведения объекта тестирования.

3.7

ошибка: Недопустимое состояние, которое испытывает система.

Примечание — Примером такой ошибки является попытка деления на ноль.

[ГОСТ 37707—2016, пункт 4.871]

3.8

программное обеспечение автоматизированной системы; программное обеспечение АС: Совокупность программ и программных документов, предназначенная для отладки, функционирования и проверки работоспособности АС.
[ГОСТ Р 59853—2021, статья 62]

3.9

программное средство в автоматизированной системе; программное средство в АС: Объект, состоящий из программ, процедур, правил, относящихся к функционированию автоматизированной системы, а также сопутствующих им документации и, если предусмотрено, данных.

[ГОСТ Р 59853—2021, статья 63]

3.10

программное обеспечение ИВС (программа, программное средство): Упорядоченная последовательность инструкций (кодов) для вычислительного средства, находящаяся в памяти этого средства и представляющая собой описание алгоритма управления вычислительными средствами и действий с данными.

[ГОСТ Р 53622—2009, статья 3.8]

3.11

программное обеспечение СИИ (программа, программное средство): Упорядоченная последовательность инструкций (кодов) для вычислительного сродства, находящаяся в памяти этого средства и представляющая собой описание алгоритма управления вычислительными средствами и действий с данными.

[Адаптировано из ГОСТ Р 59276—2020, статья 3.38]

3.12 **программное обеспечение системы;** программное обеспечение АС, ИВС, ПТК, устройства, СИИ: Совокупность программ и программных документов, предназначенная для отладки, функционирования и проверки работоспособности АС, ИВС, ПТК, устройства, СИИ.

3.13 **повторное тестирование:** Подтверждающее тестирование, проводимое для проверки успешности устранения дефекта или ошибки внесенными изменениями.

3.14 **регрессионное тестирование:** Тестирование, проводимое после внесения изменений в объект тестирования или в его операционную среду, с целью выявления дефектов в немодифицированных частях объекта тестирования.

Примечание — Регрессионное тестирование отличается от повторного тестирования тем, что проверяется не то, что модификация работает правильно, а то, что другие части системы не были случайно затронуты изменением.

3.15 **ручное тестирование:** Проведение тестов людьми путем ввода информации в объект тестирования и проверки результатов.

Примечание — Автоматизированное тестирование подразумевает использование инструментариев, роботов и других подсистем выполнения тестов. При ручном тестировании эти элементы не используются.

3.16

система искусственного интеллекта; СИИ: Техническая система, в которой используются технологии искусственного интеллекта, и обладающая искусственным интеллектом.

[ГОСТ Р 59276—2020, статья 3.16]

3.17

технические средства ИВС: Аппаратные и программные средства, используемые для сбора, обработки, хранения, манипуляции и выдачи данных.

[ГОСТ Р 53622—2009, статья 3.12]

3.18

технологии искусственного интеллекта: Комплекс технологических решений, направленных на создание систем искусственного интеллекта.

[ГОСТ Р 59277—2020, статья 3.44]

3.19 **тестирование:** Комплекс мероприятий, проводимых для упрощения выявления и оценки свойств объекта тестирования.

Примечание — Деятельность по тестированию включает в себя планирование, подготовку, выполнение, отчетность и управление в той мере, в какой они связаны с тестированием.

3.20 **тестовый пример:** Набор предварительных условий, входных данных и ожидаемых результатов, разработанный для наблюдения за поведением объекта тестирования для достижения целей тестирования.

4 Сокращения

АС — автоматизированные системы;

ИВС — информационно-вычислительные системы;

ПО — программное обеспечение;

ПТК — программно-технические комплексы;

СИИ — системы искусственного интеллекта.

5 Общие положения

5.1 Роль тестирования программного обеспечения

Программное обеспечение является одним из видов обеспечения различных автоматизированных систем наряду с математическим, информационным, лингвистическим, техническим, метрологическим, организационным, методическим и другими видами обеспечения (по ГОСТ 34.602, ГОСТ Р 59853) информационно-вычислительных систем, устройств и систем искусственного интеллекта, включая киберфизические системы (по ГОСТ Р 53622, ГОСТ Р 59276, ГОСТ Р 59277).

Примечания

1 Для ИВС, ПТК, устройства и СИИ под ПО и программными средствами понимают также упорядоченную последовательность инструкций (кодов) для вычислительного средства, находящуюся в памяти этого средства и представляющую собой описание алгоритма управления вычислительными средствами, устройства и действий с данными (по ГОСТ Р 53622, ГОСТ Р 59277).

2 В отличие от ПО, напрямую относящемуся к АС согласно своему назначению, некоторые программные средства могут не иметь статуса ПО. Например, программные средства обучения или рекламы могут не входить в состав ПО как вида обеспечения АС, если они не предназначены для отладки, функционирования или проверки работоспособности АС.

В общем случае проведение тестирования ПО должно быть направлено на проверку реализации возможностей ПО на уровне типовых и/или специфических показателей качества ПО, используемого по назначению в конкретных системах — см. подробнее 4.4. На семантическом уровне тестирование ПО должно быть напрямую связано с результативным решением задач создания, эффективного функционирования и развития систем, включая решение глобальных задач обеспечения национальной безопасности Российской Федерации [1] — [11]:

- реализации государственной стратегии в экономике;
- эффективного функционирования, обеспечения безопасности и развития сложных народнохозяйственных, инженерно-технических, транспортных систем, систем связи и коммуникаций, электроэнергетики, топливно-энергетического комплекса, нефтяной, газовой и нефтехимической промышленности, трубопроводного транспорта, строительного комплекса;
- защиты населения и территорий от чрезвычайных ситуаций природного и техногенного характера;
- технической диагностики, управления ресурсом эксплуатации, снижения экономических, экологических и социальных ущербов от природных и природно-техногенных катастроф и нарушений качества, безопасности и эффективности критически и стратегически важных систем.

Примечание — Примером ПО, подлежащего всестороннему тестированию, может служить ПО Координационного центра Правительства Российской Федерации, в интересах которого предусмотрено создание, функционирование и развитие информационной системы управления инцидентами, аналитической информационной

системы и иных информационных систем для обеспечения оперативных и согласованных действий федеральных органов исполнительной власти, органов исполнительной власти субъектов Российской Федерации и организаций при разрешении инцидентов (штатных и нештатных ситуаций) и реализации приоритетных задач Правительства Российской Федерации [12].

Результаты тестирования ПО используются для оценки, контроля и обеспечения качества, безопасности и эффективности АС, ИВС, ПТК, устройств или СИИ, в состав которых это ПО входит, а также для управления рисками в их жизненном цикле (см. ГОСТ Р 27.102, ГОСТ Р 57193, ГОСТ Р 59989, ГОСТ Р 59990, ГОСТ Р 59991, ГОСТ Р 59992, ГОСТ Р 59994).

5.2 Цели тестирования

Главной целью тестирования ПО является выявления дефектов и ошибок в ПО для повышения его качества и эффективного использования по назначению в конкретной системе, например: в АС, ИВС, ПТК, устройстве и/или СИИ.

Основными ожидаемыми результатами тестирования ПО являются:

- обнаружение явных и скрытых дефектов и ошибок в элементах тестирования до их передачи на следующий этап жизненного цикла вплоть до передачи в эксплуатацию в системе;
- получение информации о качестве элементов тестирования и остаточных рисках в жизненном цикле ПО для принятия решений о возможности использования ПО согласно своему назначению в конкретной системе;
- определение и выработка мер смягчения рисков, связанных с невыявленными дефектами и ошибками по результатам тестирования ПО.

Примечание — Необходимо осознавать, что результаты тестирования ПО показывают наличие выявленных дефектов и ошибок в ПО, а не их отсутствие, поскольку в общем случае не существует физической возможности проверить ПО за ограниченное время во всех возможных условиях его эксплуатации. Самое качественное тестирование ПО не может стопроцентно гарантировать, что в условиях неопределенностей в той или иной ситуации при реализации тех или иных информационных, природных, техногенных и/или социально-экономических угроз не проявятся недопустимые состояния или недопустимые отклонения в функционировании конкретной системы.

Тестирование обычно преследует не одну цель. Типичными целями, среди прочего, могут быть:

- сбор информации об элементе тестирования, что может быть использовано, например, программистами для обучения созданию улучшенного кода в будущем, тестировщиками для создания более качественных тестовых примеров, руководителями проектов для принятия решения о прекращении тестирования, пользователями для более высокой уверенности в качестве используемого ПО;
- обеспечение обоснованности при принятии решений. Предоставляемые доказательства того, что элемент тестирования работает правильно при определенных обстоятельствах, служат аргументами для заинтересованных сторон в том, что элемент ПО, прошедший тестирование, отвечает предъявляемым требованиям по качеству.

Для тестирования ПО могут применяться любые законные подходы, которые обеспечивают приемлемое выявление дефектов и ошибок в ПО для повышения его качества и эффективного использования по назначению в конкретной системе. Некоторые доступные подходы изложены в 5.5.2.

5.3 Принципы тестирования

При тестировании ПО руководствуются следующими основными принципами:

- системности, предполагающим наличие целеполагания и связанности действий в тестировании для приемлемых условий создания (модернизации, развития) конкретной системы;
- целенаправленности при выборе и применении методов и инструментариев тестирования, предполагающим использование доступных методов и инструментариев для выявления дефектов и ошибок в проверяемых элементах ПО, начиная с самых ранних этапов создания ПО за счет проверки наиболее вероятных и востребованных ситуаций и условий функционирования системы, для которой это ПО предназначено.

Примечание — Выявление дефектов и ошибок и их устранение повысит качество ПО, но по-прежнему не гарантирует их отсутствия для не проверенных, но возможных ситуаций и условий;

- сбалансированности эффектов от тестирования различных элементов ПО в условиях контекста тестирования, неопределенностей, возможных угроз и объективных ограничений, свойственных для условий эксплуатации конкретной системы;

- независимости тестировщиков от разработчиков (независимость тестирования считается высокой, если тест разрабатывает один человек, а проводит другой, например, потому что независимый тестировщик с меньшей вероятностью сделает те же ошибочные предположения, что и разработчик, и поэтому с большей вероятностью выявит допущенные ошибки разработчика);

- систематизации выявленных дефектов и ошибок для определения причин их возникновения (например, ошибки при выборе ненадежных составных компонентов, недостаточный охват тестового покрытия, низкая квалификация программистов или тестировщиков на предыдущих этапах тестирования) и их устранения;

- регулярного критичного пересмотра и обновления тестовых сценариев и примеров для обеспечения разнообразия подходов к тестированию, совершенствованию и развитию тестировщиков;

- ориентации на управление рисками при разработке стратегии тестирования ПО, системном анализе результатов тестирования и оценке влияния остаточных рисков на качество, безопасность и эффективность конкретной системы, для которой предназначено ПО.

Применение всех принципов включает в себя использование системного анализа требований, предметной области, контекста и рисков, расстановку приоритетов, моделирование вероятных сценариев для применения ПО, работу с конечными пользователями, применение специальных показателей, методов и инструментариев тестирования с учетом специфики конкретных систем (см. ГОСТ Р ИСО 31000, ГОСТ Р 51901.1, ГОСТ Р 51901.5, ГОСТ Р 51901.7, ГОСТ Р 51904, ГОСТ Р 54124, ГОСТ Р 54145, ГОСТ Р 58771, ГОСТ Р МЭК 62508).

5.4 Жизненный цикл тестирования

Жизненный цикл тестирования ПО характеризуется последовательностью действий по анализу требований и определению стратегии тестирования ПО для системы, разработке или уточнению критериев качества ПО, разработке планов тестирования на детальном уровне, разработке тестовых данных, выполнению тестирования с использованием тестовых данных и фиксации выявленных при этом дефектов и ошибок, системнному анализу результатов тестирования и составления отчетности.

Жизненный цикл тестирования ПО имеет свое место в жизненном цикле самого ПО и в жизненном цикле конкретной системы, для которой это ПО предназначено. Формально само тестирование ПО может рассматриваться как проект, а процесс тестирования может включать в себя процессы технического управления по ГОСТ Р 57193 (процессы планирования, оценки и контроля проекта, управления решениями, рисками, конфигурацией, информацией, процессы измерений и гарантий качества) применительно к жизненному циклу системы (см. ГОСТ Р 59793, ГОСТ 34.602, ГОСТ IEC 61508-3, ГОСТ Р ИСО 14258, ГОСТ Р ИСО/МЭК 15026, ГОСТ Р 53647.1, ГОСТ Р 57102, ГОСТ Р 59330, ГОСТ Р 59992).

Примечание — Длительность жизненного цикла тестирования ПО может варьироваться от единиц часов до многих месяцев. При этом в начале больше времени уходит на планирование, в конце — на отчетность.

В жизненном цикле тестирования ПО следует учитывать возможные возникающие риски с учетом угроз и специфики конкретной системы. Во время сопровождения ПО в рисках дополнительно должны учитываться внесенные изменения и ожидаемое влияние этих изменений на функционирование конкретной системы (см. ГОСТ Р 56921, ГОСТ Р 56922, ГОСТ Р 59356, ГОСТ Р 59990, ГОСТ Р 59994).

Действия по анализу требований и определению стратегии тестирования ПО необходимы для понимания объекта проверки (т. е. для понимания того, что подлежит тестированию), определения объема работы, анализа сложности и наличия необходимых ресурсов, инструментариев и знаний для работы по тестированию ПО применительно к конкретной системе, для которой это ПО предназначено.

Действия по разработке или уточнению критериев качества ПО при тестировании позволяет сформулировать или уточнить проверяемые характеристики и оцениваемые показатели качества ПО, признаки возможности начала тестирования.

В результате действий по разработке планов тестирования на детальном уровне определяют планы по реализации составных частей стратегии тестирования применительно к элементам проверяемого ПО (например, к документации, отдельным модулям, взаимосвязям), порядок и очередность выполнения работ, связанных с тестированием ПО.

Действия по разработке тестовых данных ПО включают в себя разработку непосредственно сценариев и данных для тестирования, охватывающих наиболее вероятные и востребованные ситуации и условия функционирования конкретной системы, для которой это ПО предназначено, пересмотр, уточнение, расширение, детализацию и прочие необходимые действия с тестовыми сценариями и данными, их наборами и иными артефактами, которые подлежат использованию при выполнении тестирования ПО.

Выполнение тестирования ПО с использованием тестовых данных и фиксация выявленных при этом дефектов и ошибок тесно связаны между собой и фактически выполняются параллельно: дефекты и ошибки фиксируются сразу по факту их обнаружения в процессе прогона тестовых данных. После выполнения всех тестов проводят обобщение для единого понимания критичности дефектов и ошибок, их взаимосвязей и срочности внесения соответствующих исправлений в ПО.

Системный анализ результатов тестирования и составление отчетности также связаны между собой, их выполняют практически параллельно. Формулируемые по результатам системного анализа выводы напрямую зависят от стратегии и плана тестирования, критериев качества ПО. Тестовые данные, порядок и очередность выполнения тестирования ПО, полученные результаты и выводы документируют, они служат аргументацией при обосновании принимаемых решений на текущем этапе жизненного цикла конкретной системы, для которой это ПО предназначено, а также основой для следующих итераций работ, связанных с тестированием различных версий этого ПО, их верификацией и валидацией, а также с предварительными и приемочными испытаниями конкретной системы, например: АС, ИВС, ПТК, устройства и/или СИИ).

5.5 Анализ требований и определение стратегии тестирования

5.5.1 Анализ требований

Анализ требований для тестирования ПО осуществляют по результатам выполнения технических процессов применительно к конкретной системе, для которой это ПО предназначено.

Для достижения целей тестирования выполняют процессы анализа бизнеса или назначения системы, определения потребностей и требований заинтересованных сторон, определения системных требований, определения архитектуры системы, определения проекта и системного анализа этого проекта (в качестве проекта выступает непосредственно комплекс работ по тестированию ПО), процессы верификации и валидации ПО согласно рекомендациям ГОСТ Р ИСО/МЭК 12207, ГОСТ Р 57193. В приложении А приведены примеры специфичной информации о требованиях к различным системам, которые могут быть проанализированы с точки зрения тестирования ПО.

В результате анализа требований к ПО конкретной системы формируют конкретные цели тестирования (см. 5.2), формируют представление о назначении ПО в системе и получают дополнительные данные, необходимые для выбора подхода к тестированию и определения стратегии тестирования ПО (см. ГОСТ 34.602, ГОСТ IEC 61508-3, ГОСТ Р ИСО 9001, ГОСТ Р ИСО/МЭК 15026, ГОСТ Р ИСО/МЭК 15026-4, ГОСТ Р ИСО 15704, ГОСТ Р ИСО/МЭК 16085, ГОСТ Р ИСО/МЭК 27001, ГОСТ Р ИСО/МЭК 27002, ГОСТ Р ИСО/МЭК 27005, ГОСТ Р ИСО/МЭК 27036-4, ГОСТ Р 51583, ГОСТ Р 51901.1, ГОСТ Р 51901.5, ГОСТ Р 51901.16, ГОСТ Р 57272.1, ГОСТ Р 57102, ГОСТ Р 57272.1, ГОСТ Р 57839, ГОСТ Р 56939, ГОСТ Р 58412, ГОСТ Р 58771, ГОСТ Р МЭК 61508-1, ГОСТ Р МЭК 61508-2.

5.5.2 Выбор подходов к тестированию

В стратегию тестирования для достижения целей тестирования, сформированных по итогам анализа требований (см. 5.5.1), включают результаты выбора подходов к тестированию ПО.

Примеры возможных применяемых подходов к тестированию ПО представлены на рисунке 1.

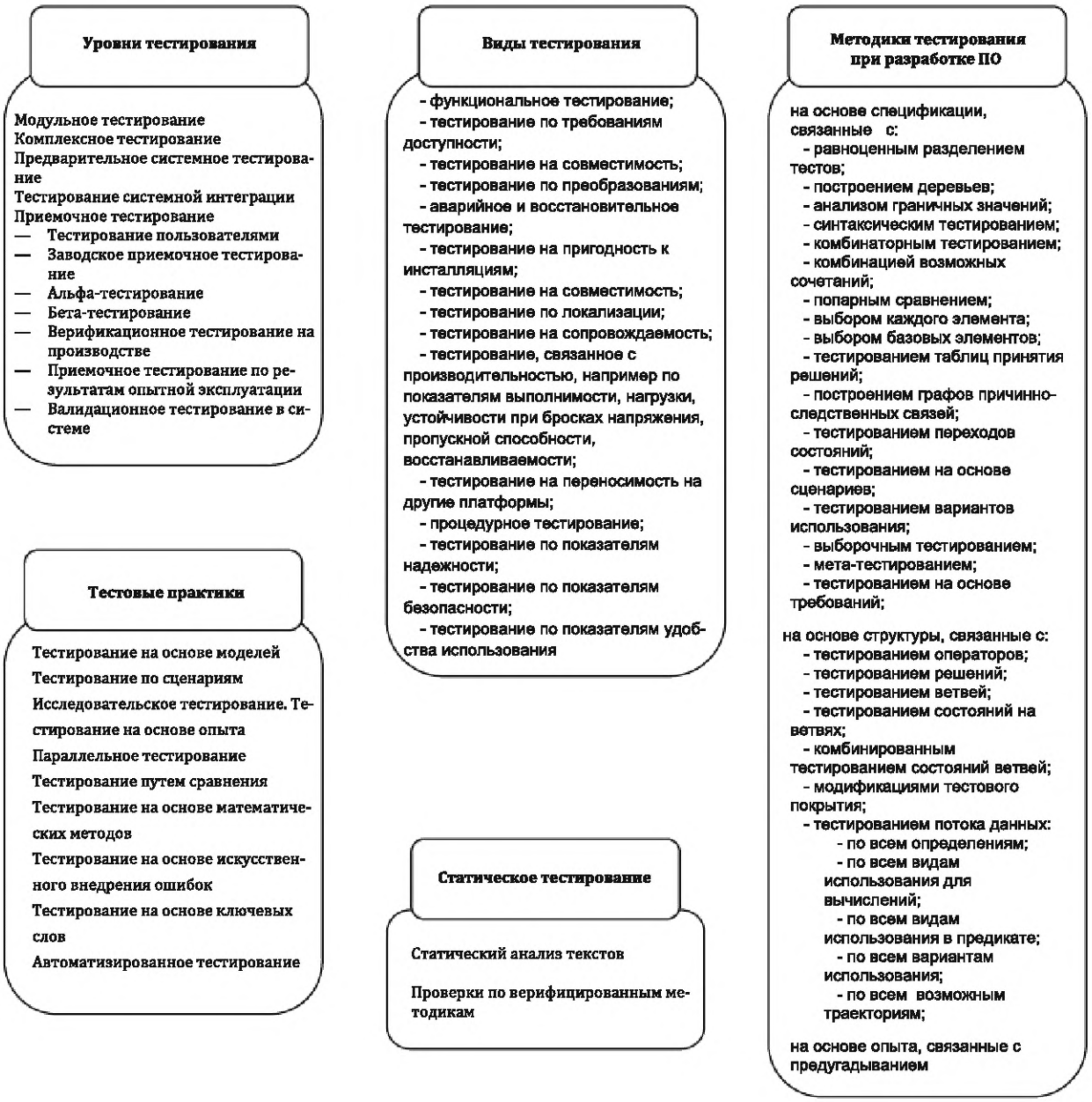


Рисунок 1 — Примеры возможных типовых подходов к тестированию ПО

Согласно ГОСТ Р ИСО/МЭК 12207 тестирование ПО проводят на различных этапах его жизненного цикла. Эти этапы определяют уровни тестирования ПО в его жизненном цикле (см. рисунок 1). Каждый уровень тестирования связан с типом элемента тестирования (например, модуль, подсистема или система), подчинен конкретным целям тестирования. Его применение может быть связано определенными условиями (например, условиями противодействия угрозам безопасности используемой информации) и сопутствующими рисками.

Уровни тестирования могут соотноситься со статическим и динамическим тестированием и тесно связаны с действиями по разработке ПО.

Статическое тестирование — это оценка элемента тестирования, при которой не происходит выполнения кода и которая может быть проведена вручную или с помощью инструментариев. При статическом тестировании объектом тестирования может быть проверяемая документация или исходной текст, а тестирование можно проводить на любом этапе жизненного цикла. Проверки различаются по степени формальности и могут включать в себя инспектирование, технические, ознакомительные и

неформальные проверки. Статический анализ подразумевает использование инструментариев для обнаружения аномалий в документах или программных кодах без выполнения программ (например, путем использования компилятора или анализатора безопасности кода). Статический анализ и проверки рекомендуется применять до динамического тестирования, поскольку они помогают выявить дефекты до того, как станет возможным выполнение тестов.

Динамическое тестирование включает в себя выполнение программного кода и контрольных примеров. Таким образом, динамическое тестирование можно проводить только на тех этапах жизненного цикла, когда доступен исполняемый код ПО.

Для динамического тестирования каждый из уровней тестирования, как правило, требует определенной тестовой среды. Общие уровни тестирования, представленные в порядке их типичного выполнения и в порядке укрупнения элементов тестирования — от отдельных блоков или модулей до полноценных систем: модульное тестирование, комплексное тестирование, системное тестирование, приемочные испытания.

В свою очередь приемочные испытания ПО могут предусматривать тестирование пользователями, заводское приемочное тестирование, альфа-тестирование, бета-тестирование, верификационное тестирование на производстве, приемочное тестирование по результатам опытной эксплуатации, валидационное тестирование в конкретной системе.

Запускать на исполнение можно как код всего приложения ПО целиком (системное тестирование), так и код нескольких взаимосвязанных частей (комплексное тестирование), отдельных частей (модульное или компонентное тестирование) и даже отдельные участки кода. Основная идея динамического тестирования состоит в том, что проверяют реальное поведение ПО в системе.

Примерами видов тестирования могут служить (см. рисунок 1):

- функциональное тестирование;
- тестирование по требованиям доступности;
- тестирование на совместимость;
- тестирование по преобразованиям;
- аварийное и восстановительное тестирование;
- тестирование на пригодность к инсталляциям;
- тестирование на совместимость;
- тестирование по локализации;
- тестирование на сопровождаемость;
- тестирование, связанное с производительностью, например: по показателям выполнимости, нагрузки, устойчивости при бросках напряжения, пропускной способности, восстанавливаемости;
- тестирование на переносимость на другие платформы;
- процедурное тестирование;
- тестирование по показателям надежности;
- тестирование по показателям безопасности;
- тестирование по показателям удобства использования.

Примерами методик тестирования, применимых при разработке ПО, могут служить (см. рисунок 1):

- методики на основе спецификации, связанные с:
  - равноценным разделением тестов;
  - построением деревьев;
  - анализом граничных значений;
  - синтаксическим тестированием;
  - комбинаторным тестированием;
  - комбинацией возможных сочетаний;
  - попарным сравнением;
  - выбором каждого элемента;
  - выбором базовых элементов;
  - тестированием таблиц принятия решений;
  - построением графов причинно-следственных связей;
  - тестированием переходов состояний;
  - тестированием на основе сценариев;
  - тестированием вариантов использования;
  - выборочным тестированием;
  - мета-тестированием;

- тестированием на основе требований;
- методики на основе структуры, связанные с:
  - тестированием операторов;
  - тестированием решений;
  - тестированием ветвей;
  - тестированием состояний на ветвях;
  - комбинированным тестированием состояний ветвей;
  - модификациями тестового покрытия;
  - тестированием потока данных:
    - по всем определениям;
    - по всем видам использования для вычислений;
    - по всем видам использования в предикате;
    - по всем вариантам использования;
    - по всем возможным траекториям;
- методики на основе опыта, связанные с предугадыванием.

Примерами тестовых практик могут служить (см. рисунок 1):

- тестирование на основе моделей;
- тестирование по сценариям;
- исследовательское тестирование;
- тестирование на основе опыта;
- ручное тестирование путем сравнения;
- параллельное тестирование;
- тестирование с использованием математических методов;
- тестирование, связанное с искусственным внедрением ошибок;
- тестирование на основе ключевых слов;
- автоматизированное тестирование (набор методов, методик и инструментариев, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования. Тестовые примеры частично или полностью выполняются специальными инструментариями, однако разработка тестовых примеров и сценариев их применения, подготовка данных, оценка результатов выполнения, написание отчетов об обнаруженных дефектах — все это по-прежнему делает человек).

Более подробно подходы к тестированию приведены в ГОСТ Р ИСО/МЭК 25000, ГОСТ Р ИСО/МЭК 25010, ГОСТ Р ИСО/МЭК 25020, ГОСТ Р ИСО/МЭК 25023, ГОСТ Р 70921, ГОСТ Р ИСО/МЭК 25040.

5.5.3 Определение стратегии

Как правило, типовое содержание стратегии тестирования включает:

- выбранный подход к тестированию с его обоснованием (см. рисунок 1):
  - уровни тестирования (могут быть частью плана тестирования в жизненном цикле ПО и системы);
  - виды тестирования;
  - определенные методики тестирования при разработке ПО;
  - выбранные варианты статического тестирования (анализа);
- методы разработки тестов (часто выбираются на основе навыков команды и степени владения этими методами, формата тестирования и восприятия ожидаемых типов дефектов на основе опыта);
- ожидаемые результаты тестирования (могут быть обусловлены нормативными требованиями с учетом рисков);
- критерии завершения тестирования (должны соответствовать методикам тестирования, так как дают возможность определить степень достаточности выполненных тестов);
- критерии управления процессом тестирования (входа и выхода для действий по тестированию);
- возможные отклонения от сложившейся практики тестирования в организации;
- степень независимости участников тестирования в части:
  - собираемых показателей (которые часто связаны с критериями завершения тестирования, но также могут использоваться в целях улучшения процесса тестирования);
  - требований к тестовым данным (для реагирования на риски, связанные с достоверностью тестовых данных в сравнении с реальными данными);
  - требований к тестовой среде (где следует указать требуемый уровень автоматизации тестирования);

- повторного тестирования (направленного на подтверждение устранения выявленных ранее дефектов);

- регрессионного тестирования (направленного на проверку того факта, что в ранее работоспособной функциональности ПО не появились ошибки, вызванные изменениями разработчиков в ПО или в среде его функционирования);

- критериев приостановки и возобновления тестирования (чтобы предоставить тестировщикам согласованный набор условий, которые разработчики должны выполнять для достижения целей тестирования).

5.6 Планы тестирования

План описывает цели тестирования ПО на определенном этапе его жизненного цикла и действия, которые должны быть выполнены для достижения этих целей. План тестирования описывает последовательность работ по реализации стратегии тестирования с указанием конкретных действий, сроков, участников и ответственных за тестирование.

План и стратегия тестирования могут быть применены к проекту, одному или нескольким уровням тестирования либо одному или нескольким видам тестирования ПО с учетом различных рисков и специфических требований к системе (см. ГОСТ Р 59336).

План тестирования содержит описание среды тестирования и указания, как следует проводить тестирование в рамках соответствующих процессов тестирования. Поскольку процессы тестирования могут быть реализованы для достижения различных целей (например, для управления тестированием проекта или системным тестированием, для тестирования производительности), соответствующие планы тестирования могут иметь различную направленность в зависимости от причин, обусловливающих необходимость проведения работ по тестированию. При этом типы исходной и отчетной информации, содержащейся в плане тестирования, как правило, остаются идентичными, независимо от их содержательного наполнения.

В процессе мониторинга и контроля тестирования план используют в качестве основы для управления тестированием. План определяет порядок проведения тестирования (например, необходимые методы тестирования, критерии завершения тестирования, а также кто и когда должен проводить тестирование). План тестирования также используют для указания того, как следует проводить статическое тестирование (например, проверки и статический анализ), а также организацию и контроль действий в процессе динамического тестирования.

План тестирования может обновляться в процессе мониторинга и контроля тестирования, в том числе в связи с изменением требований или в случае возникновения недопустимых рисков относительно приемлемого качества и сроков выполнения заданного объема тестирования ПО с учетом специфических требований к системе (см. ГОСТ Р 56939, ГОСТ Р 58412, ГОСТ Р 59334, ГОСТ Р 59335, ГОСТ Р 59989, ГОСТ Р 59991).

5.7 Среда тестирования

Для проведения тестирования ПО определяют среду тестирования, включающую в себя помещения, оборудование, различное техническое, математическое, программное, информационное, метрологическое, организационное, методическое и другие виды обеспечения конкретных систем (например: для АС, ИВС, ПТК, СИИ), необходимых для проведения тестирования.

Среда тестирования может содержать несколько сред для проведения определенного уровня тестирования, применения востребованных видов тестирования (см. рисунок 1). Среда тестирования может состоять из нескольких взаимосвязанных виртуальных сред и систем.

Требования к среде тестирования могут включать в себя требования:

- к программному обеспечению, характеризующему среду — например, к операционной системе, системе управления базами данных, приложениям, вспомогательным программам;

- сервисам (например: виртуальным или облачным сервисам);

- техническому обеспечению, оборудованию (например: серверам и настольным компьютерам конечных пользователей, ноутбукам и мобильным устройствам);

- организационному обеспечению, включая требования к квалификации тестировщиков;

- информационному обеспечению и данным, включая данные, хранящиеся в базах данных;

- сетям связи (например: коммутаторам, маршрутизаторам и сетевым соединениям на определенных скоростях);

- интерфейсам (например: к внутренним, виртуальным или сторонним системам);
- периферийным устройствам — таким как принтеры, сканеры;
- инструментариям тестирования (например таким, как инструментарии управления и выполнения тестов);
- доступности к среде тестирования (например: количество часов в день и время, необходимое для выполнения необходимых работ).

Требования к тестовой среде должны быть записаны так, чтобы их можно было передать и понять. Если отдельная группа специалистов создает тестовую среду, то эта группа должна подготовить отчет о ее готовности в соответствии с требованиями.

Требования к среде тестирования формулируют в плане тестирования, чтобы можно было получить, настроить, сконфигурировать и проверить все необходимые компоненты перед выполнением тестов и привести их в исходное или нужное состояние после выполнения тестовых работ. К началу тестирования ПО состояние среды тестирования фиксируют в отчете о готовности среды для проведения тестирования, а по завершении тестирования ПО — в отчетной документации о результатах тестирования. Необходимо предусмотреть порядок сопровождения среды, определяющий, когда и как будут решаться возможные технические проблемы со средой тестирования. Состояние среды тестирования может меняться при реализации процессов управления тестированием с учетом специфики системы (см. ГОСТ Р 56921, ГОСТ Р 56922, ГОСТ Р 56939, ГОСТ Р 58412, ГОСТ Р 59334, ГОСТ Р 59989, ГОСТ Р 59991).

5.8 Процессы тестирования

С различных точек зрения процессы тестирования ПО с учетом специфики систем описаны в ГОСТ Р ИСО/МЭК 12207, ГОСТ Р ИСО/МЭК 16085, ГОСТ Р 56921, ГОСТ Р 56939, ГОСТ Р 57193, ГОСТ Р 59342. В ГОСТ Р 56921 процессы тестирования определены на трех уровнях — на уровне организации, проектно-ориентированного тестирования, управления тестированием, статического и динамического тестирования. По мере необходимости процессы тестирования могут быть дополнены востребованными для достижения целей действиями, процедурами и практиками.

5.8.1 Тестирование на уровне организации

Тестирование на уровне организации связано с определением и поддержанием политики и практик тестирования, которые применяются в рамках всех проектов организации, а не в приложении к конкретному проекту. Правила и рекомендации, определенные в политике тестирования и одном или нескольких документах по практикам тестирования на уровне организации, служат основой для всех мероприятий по тестированию ПО в организации. Существование таких правил и рекомендаций определяет ограничения для проектов, которые устраняют необходимость дублирования в решениях (например, по использованию конкретной системы автоматизации тестирования во всех проектах) и способствуют повторному использованию навыков и взаимодействию между проектами. Документация, относящаяся к политике и практикам тестирования на уровне организации, как правило, имеется в более опытных и крупных организациях, реализующих множество проектов. В организациях менее опытных тестирование может быть выполнено без политики тестирования и документированных практик тестирования (что снижает согласованность, эффективность и результативность тестирования).

5.8.2 Проектно-ориентированное тестирование

При разработке и сопровождении ПО заинтересованные стороны выполняют набор взаимосвязанных процессов, обычно определяемых моделью жизненного цикла системы. Модель жизненного цикла действует как общая основа для общения и понимания как при первоначальной разработке, так и в ходе последующего сопровождения ПО и системы в целом (см. ГОСТ Р 59330, ГОСТ Р 59992).

Процессы тестирования, используемые в проекте, могут быть предписаны нормативными документами (например, ГОСТ Р 56939), организационными ограничениями (например, по практикам тестирования) и конкретными потребностями проекта.

5.8.3 Процессы управления тестированием

Управление проводимым тестированием описывают соответствующими процессами. Первоначально план тестирования создают в процессе разработки стратегии и планирования тестирования на основе анализа выявленных рисков и ограничений проекта, а также с учетом сложившихся практик тестирования на уровне организации. План определяет процесс тестирования, который управляется с помощью процесса мониторинга и контроля тестирования. Процесс завершения тестирования включает в себя, среди прочего, мероприятия по формированию отчета о завершении тестирования, упраздне-

нию тестовой среды, архивированию средств тестирования и усвоению извлеченных уроков. Процессы управления тестированием отражены на рисунке 2.



Рисунок 2 — Процессы управления тестированием

По необходимости может быть создано несколько экземпляров процессов управления тестированием. Пример иерархии созданных экземпляров процессов управления тестированием приведен на рисунке 3.

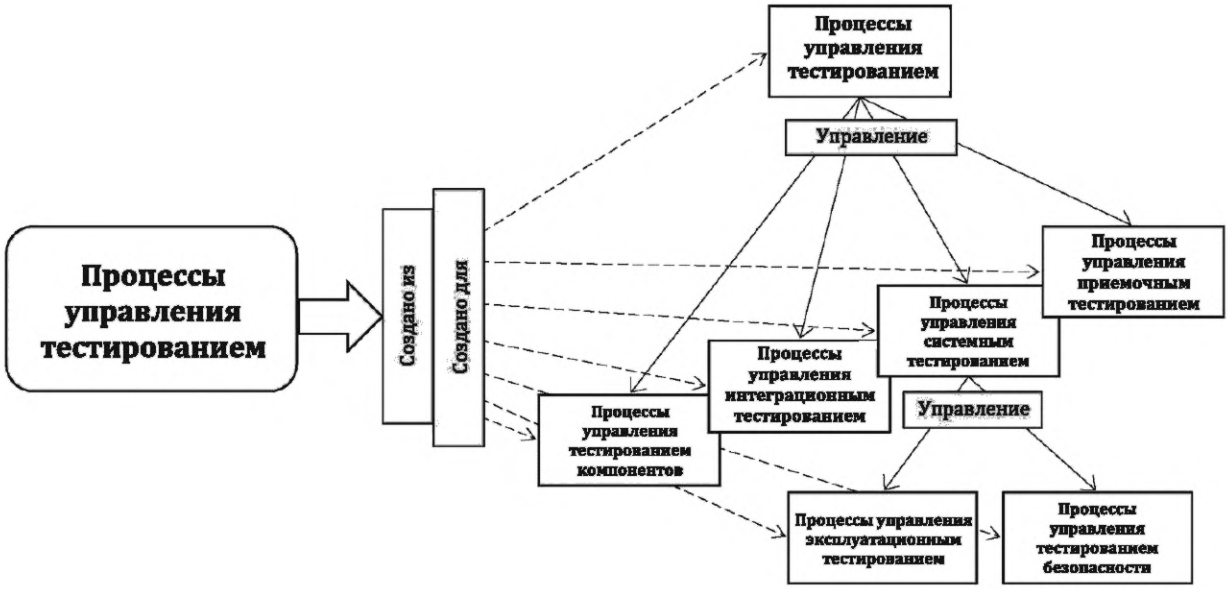


Рисунок 3 — Пример иерархии созданных экземпляров процессов управления тестированием

5.8.4 Процессы статического тестирования

Тестирование ПО возможно без выполнения его на компьютере. Во многих областях промышленности такое тестирование называют статическим тестированием. Действия статического тестирования ПО считаются крайне важными для полного жизненного цикла тестирования, особенно для раннего обнаружения дефектов и ошибок и снижения полной стоимости проекта.

Статическое тестирование может также включать в себя использование инструментариев статического анализа, которые помогают находить дефекты в исходных текстах программ или документации без выполнения программного кода.

5.8.5 Процессы динамического тестирования

Процессы динамического тестирования описывают порядок разработки тестовых примеров, создания тестовой среды, выполнения тестовых примеров и информирования о возникающих проблемах. Взаимодействие процессов динамического тестирования показано на рисунке 4, где отражены логиче-

ски упорядоченные элементы только для одного теста (при тестировании проекта эти процессы будут выполняться многократно). Некоторые процессы могут выполняться чаще, чем другие (например, настройка среды обычно происходит реже, чем выполнение теста). Кроме того, на практике некоторые из этих процессов могут протекать параллельно (например, проектирование теста может выполняться одновременно с созданием тестовой среды).

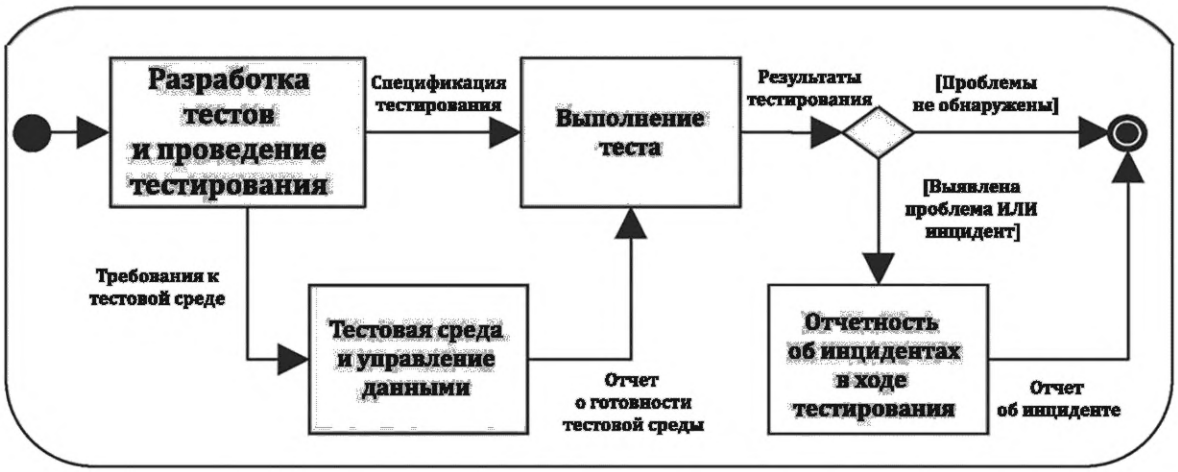


Рисунок 4 — Процессы динамического тестирования

На уровне динамического тестирования экземпляры набора процессов динамического тестирования обычно создаются многократно, отдельно для каждого уровня и вида тестирования (например, для тестирования компонентов, интеграционного тестирования, системного тестирования, приемочного тестирования, тестирования производительности, удобства использования и безопасности).

На практике такие экземпляры также часто создаются отдельно для тестирования различных частей системы (например, они могут быть созданы отдельно для различных модулей в целях тестирования компонентов или для приемочного тестирования функций, связанных с конкретной заинтересованной стороной).

Таким образом, проектирование и выполнение тестов, связанных с конкретной функцией, могут осуществляться отдельно от проектирования и выполнения других функций. На рисунке 5 отражены процессы динамического тестирования, экземпляры которых были созданы отдельно для каждого уровня тестирования и отдельно для тестирования компонентов трех различных модулей.

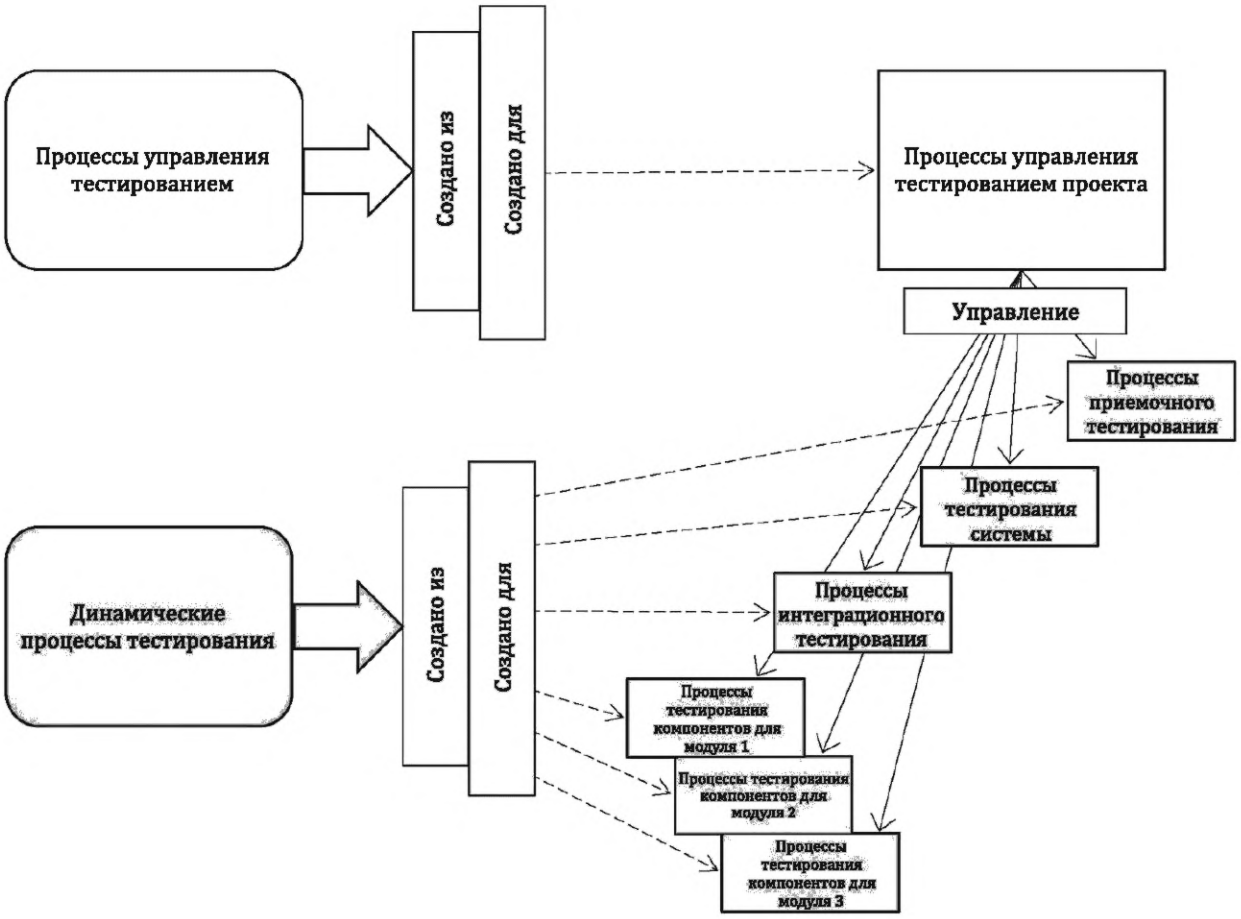


Рисунок 5 — Создание нескольких экземпляров наборов процессов динамического тестирования

5.8.6 Совершенствование процессов

Совершенствование процессов включает в себя определение и внедрение изменений в реализуемые процессы, чтобы они более эффективно отвечали бизнес-целям организации.

В процессах тестирования проходит сбор информации, на основе которой может быть инициировано внесение изменений в цели, методы и инструментарии тестирования. В свою очередь сами процессы тестирования могут оказаться объектом совершенствования.

Совершенствование процессов тестирования может быть выполнено в рамках более масштабной инициативы, охватывающей все процессы разработки, тестирования и сопровождения ПО.

5.9 Проведение тестирования

5.9.1 Объекты тестирования

Проведение тестирования ПО должно быть направлено на получение информации о степени готовности ПО в его жизненном цикле и выявление максимально возможного числа дефектов и ошибок на возможно более ранних этапах разработки ПО при заданных ограничениях стоимости и графика разработки.

Объектами тестирования могут быть как отдельные программные элементы или подсистемы как части более крупной системы, так и полноценные системы (например, АС, ИВС, ПТК, устройство, СИИ), включая оборудование, ПО и сопроводительную документацию. Элемент тестирования может быть как исполняемым (например, двоичный код или модели), так и неисполняемым (например, документированная спецификация). Если элемент тестирования является исполняемым, то может быть проведено динамическое тестирование. В противном случае может быть проведено только статическое тестирование (см. рисунок 1).

При тестировании ПО широко применяются различные инструментарии (которые, в свою очередь, также могут быть объектами тестирования). Примерами областей, охватываемых инструментариями тестирования, являются:

- управление контрольными примерами;
- статический анализ;
- мониторинг и контроль тестирования;
- генерация тестовых данных и примеров;
- выполнение тестовых примеров;
- реализация и обслуживание тестовой среды.

Существует большое разнообразие инструментариев автоматизации тестирования. Они могут быть разработаны собственными силами, приобретены на коммерческой основе или получены от сообществ, работающих над решениями по созданию ПО с открытыми исходными текстами.

На практике используется ручное тестирование, автоматизированное тестирование или их комбинация.

5.9.2 Ручное и автоматизированное тестирование

Контрольные примеры при тестировании ПО могут быть запущены вручную тестировщиком либо с помощью инструментариев в автоматизированном режиме.

Решение о проведении ручного или автоматизированного тестирования зависит от различных факторов, например:

- от вероятного количества повторных запусков тестового примера. Так, традиционные жизненные циклы могут не требовать частого регрессионного тестирования, в то время как гибкие проекты обычно требуют регрессионного тестирования по крайней мере для каждой новой версии, а проекты, использующие непрерывную интеграцию, могут повторно запускать тестовый пример при каждой сборке. Может оказаться, что экономически эффективнее автоматизировать повторяющееся тестирование;
- от уровня квалификации и возможностей тестировщиков в проекте, поскольку большинство инструментариев автоматизации тестирования обычно требуют от тестировщиков определенных навыков программирования;
- от бюджета и времени, приемлемых для приобретения и апробации инструментариев автоматизации тестирования, а также для найма и/или обучения тестировщиков.

Примеры тестовых практик, рекомендуемых для проведения тестирования, отражены на рисунке 1 и описаны в 5.9.3.

5.9.3 Примеры тестовых практик

5.9.3.1 Тестирование на основе моделей

С помощью моделей формируются формальные или полуформальные представления требуемого поведения тестируемого объекта. Тестирование на основе моделей позволяет тестировщикам проектировать и реализовывать тесты на различных уровнях абстракции. Языки и нотации тестирования на основе моделей формально могут быть определены синтаксически и в некоторых конкретных случаях семантически. Используя среду поддержки инструментариев, контрольные примеры могут быть быстро сгенерированы и выполнены автоматически. Таким образом тестирование на основе моделей может обеспечить более совершенное тестирование, чем естественные языки и тестирование вручную.

5.9.3.2 Тестирование по сценариям

При тестировании по сценариям тестовые примеры фиксируются (например, документируются в инструментарии управления тестированием или в электронной таблице), после чего могут быть выполнены вручную или с помощью инструментариев автоматизированного тестирования.

Тестирование по сценариям помогает достичь более широкого охвата тестирования. Уровень детализации, требуемый в каждом контрольном примере (например, детализация этапов тестирования и ожидаемых результатов) часто зависит от требований, которые обычно определяют во время планирования тестирования (см. 5.6). Уровень детализации также зависит от того, является ли тестирование автоматизированным (что, как правило, требует более глубокой детализации). Для ручных тестов уровень детализации зависит от знаний, опыта и возможностей тестировщика, включая понимание системы в целом (например: АС, ИВС, ПТК, устройство, СИИ) и всей области тестирования.

5.9.3.3 Исследовательское тестирование

При исследовательском тестировании тесты разрабатываются во взаимодействии тестировщика с элементом тестирования и выполняются путем непосредственных сеансов взаимодействия. Листы сеансов такого взаимодействия часто используются для структурирования самого процесса исследовательского тестирования (например, путем определения основной цели и ограничений по времени для

каждого сеанса тестирования в приложении к элементу тестирования, а при необходимости к подсистеме и/или системе в целом). Эти же листы сеансов используются для записи информации о том, что было протестировано, а также о любом замеченном аномальном поведении. Исследовательские тесты зачастую не полностью описываются в сценариях, так как высокоуровневые сценарии тестирования («замыслы тестирования») часто документируются в листах сеансов, чтобы определить основную цель конкретного сеанса при исследовательском тестировании.

Вместе с тем исследовательское тестирование позволяет проявить творческий подход в проведении необходимого тестирования ПО. Во время исследовательского тестирования цель часто состоит в том, чтобы сфокусироваться на конкретных типах дефектов и ошибок и «охватить» конкретные аспекты требований, спецификаций или исходного кода.

5.9.3.4 Тестирование на основе опыта

Тестирование на основе опыта основывается на знаниях, собранных по итогам предыдущих работ по тестированию, знаниях конкретного ПО, предметной области и системы в целом.

Практики тестирования на основе опыта, как правило, не полагаются на большое количество документации. Эти методы, основанные на опыте, по большей части относятся к тестированию без сценария. Использование таких методов может обеспечить лишь ограниченное соответствие требованиям каких-либо стандартов (например, ГОСТ Р 56921).

5.9.3.5 Параллельное тестирование

При параллельном тестировании альтернативная версия (например, существующее ПО или ПО, разработанное другой группой, или ПО, реализованное с использованием другого языка программирования, либо исполняемый проектный макет ПО) используется в качестве системы для сравнения в целях получения ожидаемых результатов и выполнения сравнения на основе одних и тех же входных данных тестирования.

При исполнении функционально эквивалентного ПО (альтернативной версии) автоматически получаются только ожидаемые результаты.

При использовании совместно с инструментариями для генерации входных данных в целях тестирования этот метод становится основой более производительной технологии проведения автоматизированного тестирования в больших объемах.

5.9.3.6 Тестирование путем сравнения

Тестирование путем сравнения — это метод статистического тестирования, который позволяет тестировщикам определить, какая из двух конкурирующих программ работает лучше. Так тестирование путем сравнения часто используют для оптимизации дизайна пользовательского интерфейса.

Примечание — Например, дизайнер пользовательского интерфейса выдвигает гипотезу о том, что, изменив цвет кнопки «Купить» с текущего красного на синий, можно увеличить продажи. Создается новый вариант интерфейса с синей кнопкой, и эти два интерфейса назначаются разным пользователям. Сравниваются показатели продаж для двух вариантов и, учитывая статистически значимое количество использований, можно определить, была ли гипотеза верна.

5.9.3.7 Тестирование на основе математических методов

Математические методы в тестировании рекомендовано использовать для планирования, разработки, отбора данных и определения входных условий, если требуемое поведение элемента тестирования, пространство ввода или пространство вывода можно описать достаточно подробно, например с помощью формального языка. Формальные языки спецификаций могут быть использованы для поддержки тестирования на основе моделей, а формальный анализ исходного кода может быть использован для выявления потенциальных рисков и информирования о стратегии тестирования.

Из-за большого количества входных данных, которые чаще всего могут быть получены с помощью математических методов, как правило, требуются автоматизированные инструментарии.

5.9.4 Управление тестовыми данными

Необходимо определить требования к настройке и очистке тестовых данных, чтобы обеспечить состояние системы для выполнения и воспроизведения каждого тестового примера.

Примечание — Например, при тестировании программы, связанной с определением кредитного рейтинга для оценки клиента, требуются его банковские данные прошлых периодов. Персональные данные могут потребовать анонимизации или обезличивания для соблюдения правил в отношении конфиденциальности данных.

Настройка, очистка и обеспечение безопасности тестовых данных могут быть выполнены вручную или автоматически с помощью определенных сценариев. Тестовые данные и автоматизированные сценарии должны поддерживаться системой управления их версиями.

5.9.5 Учет тестирования при управлении проектами

Оценка, анализ рисков и планирование мероприятий по тестированию должны быть объединены с мероприятиями по планированию проекта в целом.

Результаты, полученные в ходе мероприятий по детальному тестированию, анализируют и передают руководителю проекта для анализа в контексте проекта. Это может привести к изменениям в плане проекта, обновлению планов проекта и выдаче соответствующих указаний тестировщикам в целях обеспечения соответствия тестирования плану проекта.

5.9.6 Управление конфигурацией

Управление конфигурацией — это набор действий, используемых процессами тестирования на всех трех уровнях для того, чтобы элементы конфигурации были однозначно идентифицированы, изменения этих элементов были зарегистрированы, а их текущее состояние могло быть отражено в отчетах. Элементы конфигурации, представляющие особый интерес для тестировщиков, — это входные параметры и результаты процессов тестирования, они включают в себя всю документацию по тестированию. Система управления конфигурацией должна по возможности позволять повторить тест в любой момент в будущем при тех же условиях. Допустимой практикой является исключение определенных видов тестирования из этого требования о повторяемости при условии, что исключение документировано.

5.9.7 Соблюдение требований по защите информации

В период подготовки, при проведении тестирования, анализе результатов тестирования, при реализации процессов управлении соответствующей информацией с учетом специфики системы должны соблюдаться требования по защите информации (см. ГОСТ Р 56921, ГОСТ Р 56922, ГОСТ Р 56939, ГОСТ Р 58412, ГОСТ Р 59334, ГОСТ Р 59341).

5.10 Документирование

Документацию по тестированию создают в результате выполнения процессов тестирования (см. ГОСТ 34.201, ГОСТ IEC 61508-3, ГОСТ Р 2.601, ГОСТ Р 51583, ГОСТ Р 51904, ГОСТ Р 56922, ГОСТ Р 56939, ГОСТ Р 59795).

5.11 Управление инцидентами

Результаты тестирования необходимо систематически анализировать с тем, чтобы определить, относятся ли выявленные дефекты и ошибки к объекту тестирования или к самим тестам. Результат тестирования, отличающийся от прогнозируемого, рассматривается как инцидент, который должен быть зарегистрирован, расследован, а выявленные дефекты и ошибки исправлены приемлемым образом. Эти работы осуществляются в рамках процесса управления инцидентами (см. ГОСТ Р 59710, ГОСТ Р 59711, ГОСТ Р 59712, а также [13]).

5.12 Использование тестирования при верификации и валидации

Верификация и валидация (аттестация) — это отдельные процессы, которые используют тестирование в качестве одного из основных методов контроля качества, безопасности и эффективности конкретных систем с учетом их специфики. Верификация фокусируется на контроле соответствия элемента тестирования спецификациям, указанным требованиям или другим документам, в то время как главной целью валидации является оценка приемлемости элемента тестирования для удовлетворения потребностей заинтересованных сторон при использовании по назначению. Тестирование (статическое и динамическое) поддерживает как верификацию, так и валидацию (см. рисунок 1). Более подробно рекомендации приведены в ГОСТ Р ИСО/МЭК 12207, ГОСТ Р 57193, ГОСТ Р 59352, ГОСТ Р 59354.

Приложение А
(справочное)

Примеры учета особенностей при тестировании

А.1 Общие положения

Многим конкретным системам свойственны специфические сопутствующие характеристики, с которыми можно связать возможные подходы к тестированию. Ниже приведен ряд примеров, где такая связь может быть продемонстрирована.

А.2 Системы искусственного интеллекта

СИИ существуют в различных формах и используются для решения целого ряда задач (например: поисковые системы, распознавание изображений, понимание и синтез речи). Какую бы форму ни принимали СИИ, они, как правило, сложны, трудны для понимания, а их применение связано со множеством неопределенностей.

Типичные сопутствующие характеристики приведены в А.14.

Для таких систем возможно применение параллельного тестирования (см. 5.9.3.5) и тестирования с анализом метаморфической связи, когда учитывается влияние изменений во входных данных для тестового примера на ожидаемые выходные данные на основе требуемого поведения объекта тестирования.

Различные архитектуры и процесс разработки, связанные с СИИ, означают, что тестировщикам (или специалистам по исследованию данных в роли тестировщиков) требуются специальные навыки (см. ГОСТ Р 53622, ГОСТ Р 59276, ГОСТ Р 59277, ГОСТ Р 59898, ГОСТ Р 70250, ГОСТ Р 70462.1, ГОСТ Р 71199).

А.3 Автономные системы

Автономные системы относятся к широкому спектру областей применения и могут сильно различаться с точки зрения технологии реализации. Автономную систему определяют как систему, которая работает в течение длительных периодов времени независимо от контроля человека. Это определение означает, что автоматические и автоматизированные системы также могут быть автономными системами. Автономные системы можно различить по уровням гибкости, что показано в таблице А.1.

Таблица А.1 — Уровни гибкости в автономных системах

Гибкость (последующие уровни могут включать предыдущие уровни)	Примеры систем
Четкие правила	Простой термостат
Четкие правила с обратной связью	Антиблокировочная тормозная система (ABS)
Фиксированная нейронная сеть, которая не меняет своего поведения в зависимости от накопленного опыта	Система борьбы с пробками
	Робот производственной линии (с использованием обработки изображений на основе машинного обучения)
Система изменяет свое поведение, основываясь на своем прошлом опыте	Самообучающийся термостат
	Финансовая торговая система
	Интеллектуальное здание

Типичные сопутствующие характеристики — это характеристики реального времени (см. А.11).

Один из подходов к тестированию автономности состоит в том, чтобы попытаться вывести систему из ее автономного поведения и запросить вмешательство в неопределенных обстоятельствах. Это является формой негативного тестирования, т. е. такого тестирования, которое направлено на исследование работы приложения в ситуациях, когда с ним выполняют некорректные операции и/или используют данные, потенциально приводящие к ошибкам. Негативное тестирование также может использоваться для того, чтобы попытаться «обмануть» систему, заставив ее думать, что она контролирует ситуацию, когда она должна запросить вмешательство (например, путем создания тестовых сценариев на границе ее операционной оболочки, предлагая применение граничных сценарных значений) (см. ГОСТ Р ИСО 17359, ГОСТ Р 51904, ГОСТ Р МЭК 61508-1, ГОСТ Р МЭК 61508-2, ГОСТ IEC 61508-3, ГОСТ Р МЭК 61508-6, ГОСТ Р МЭК 61508-7).

А.4 Уникальные системы

Уникальные системы создают под конкретный заказ и для определенных целей [т. е. они не являются коммерческими системами (см. А.6)].

Сопутствующие характеристики отсутствуют.

Приемочное тестирование проводят под требования конкретного заказчика.

А.5 Системы замкнутого цикла

Система замкнутого цикла — это система, которая пытается достичь желаемого состояния, используя обратную связь (примером такой системы является контроллер нагрева, который будет управлять нагревом для поддержания желаемой температуры на входе, а также контролировать текущую температуру для достижения своей цели).

Типичные сопутствующие характеристики — это характеристики реального времени (см. А.7, А.11, А.14).

При тестировании системы замкнутого цикла тестировщик должен учитывать, как ранее поданные тестовые сигналы генерируют обратную связь, которая действует как вход в тестируемую систему. Такое тестирование отличается от тестирования системы с разомкнутым контуром, где все входы напрямую контролирует тестировщик.

Пока система замкнутого цикла не будет готова к тестированию в полноценной рабочей среде, тестирование системы замкнутого цикла является специфическим, поскольку необходимы модели установки и среды для имитации обратной связи с системой на основе ее выходов. Модель установки имитирует результаты воздействия системы на управляемую установку (например, двигатель, коробку передач, тормоза), а модель среды имитирует результаты воздействия системы управления на окружающую среду (см. ГОСТ Р ИСО 17359, ГОСТ Р 51904, ГОСТ Р МЭК 61508-1, ГОСТ Р МЭК 61508-2, ГОСТ IEC 61508-3, ГОСТ Р МЭК 61508-6, ГОСТ Р МЭК 61508-7).

А.6 Готовый к использованию коммерческий продукт

Примерами являются системы, предназначенные для продажи многим клиентам (например, коробочное ПО вместо разработанного на заказ).

Сопутствующие характеристики отсутствуют.

Понимание операционной среды неизвестных покупателей может быть затруднено, поэтому часто используется альфа- и бета-тестирование.

А.7 Встраиваемые системы

Физически встроенная программная система, которая обычно представляет собой управляющую часть более крупной системы, такой, например, как контроллер манипулятора робота или система авионики на самолете или СИИ на автомобильном транспорте. Встраиваемые системы разрабатывают для конкретной цели, и их часто оптимизируют для этой цели с использованием специальных защищенных процессоров и оборудования с меньшим энергопотреблением, чем компьютеры общего назначения.

Характеристики реального времени часто ассоциируются со встроенными системами (см. А.11).

Встроенное ПО в какой-то момент будет интегрировано и протестировано на специализированном целевом оборудовании для этой встроенной системы.

Некоторые встроенные системы будут работать в неблагоприятной среде. В таких случаях может потребоваться использование моделирования для проведения испытаний в экстремальных условиях.

Некоторые встраиваемые системы могут быть дорогостоящими с точки зрения ремонта после запуска в эксплуатацию из-за их рабочей среды (например, когда они являются частью ядерного реактора) или из-за стоимости ремонта многих систем, которые не могут быть доставлены для ремонта (например, контроллер стиральной машины). Такая высокая стоимость ремонта часто влечет за собой необходимость тщательного тестирования.

Поскольку встроенные системы обычно взаимодействуют со специализированным оборудованием, тестирование на основе внедрения точек отказа может использоваться для проверки реакции системы на сбои в этом оборудовании.

Встраиваемые системы часто имеют серьезные ограничения по размеру, весу и стоимости, что означает, что они часто конфигурируются с минимальным объемом памяти, который может поддерживать систему. В таких ситуациях может быть уместно тестирование с утечкой объемов памяти (см. ГОСТ Р ИСО 17359, ГОСТ Р 51904, ГОСТ Р МЭК 61508-1, ГОСТ Р МЭК 61508-2, ГОСТ IEC 61508-3, ГОСТ Р МЭК 61508-6, ГОСТ Р МЭК 61508-7, ГОСТ Р 70250).

А.8 Системы высокого уровня надежности

ПО систем высокого уровня надежности не предназначено для обновления в течение срока службы этих систем (например, зонды для дальнего космоса, подводные системы).

Типичные сопутствующие характеристики — аналогично как для встроенного ПО (см. А.7).

Тестирование систем высокого уровня надежности должно быть более строгим из-за невозможности исправить какие-либо дефекты после запуска системы в эксплуатацию.

Для проверки надежности системы в случае внутренних и внешних сбоев может быть рекомендовано негативное тестирование путем введения искусственных ошибок.

А.9 Интернет вещей (IoT-система)

IoT-система — это масштабируемая система, основанная на интернет-технологии, которая позволяет взаимодействовать широкому спектру устройств (например, датчикам, узлам обработки или агрегации и возможно исполнительным механизмам) с различной степенью вычислительной мощности. Протоколы связи и расстояния могут различаться, при этом большинство IoT-систем обмениваются данными на трех уровнях — между устрой-

ствами, на периферии (облака) и в масштабах облака. IoT-системы могут включать в себя множество устройств разных поставщиков, что делает стандартизацию связи и спецификацию предоставляемых функций особенно важной.

IoT-системы часто классифицируются как потребительский Интернет вещей (например, носимые устройства, подключенные автомобили) и промышленный Интернет вещей (например, интеллектуальная розничная торговля, транспорт).

Частично с IoT-системами иногда ассоциируются характеристики встроенного ПО (см. А.7). Например, IoT-система умного города обычно использует множество встроенных подсистем, которые являются подчиненными для центральной невстроенной системы — см., например, ГОСТ Р 71199.

При наличии нескольких взаимодействующих устройств в одной IoT-системе используется тестирование на совместимость, в том числе для проверки взаимодействия различных устройств от разных поставщиков. Необходимость соблюдения стандартов связи обуславливает важность тестирования интерфейсов.

IoT-системы часто имеют возможность заменять существующие и добавлять новые устройства в систему, для чего может оказаться важным тестирование масштабируемости.

IoT-системы также обычно сопряжены с проблемами безопасности, что приводит к необходимости соответствующего тестирования. Промышленный IoT, как правило, использует много данных, что может привести к общим проблемам с обеспечением конфиденциальности. Это может повлиять на тестирование ПО с учетом требований по защите информации (см. ГОСТ Р 51904, ГОСТ Р 56921, ГОСТ Р 56922, ГОСТ Р 56939, ГОСТ Р 58412, ГОСТ Р 59334, ГОСТ Р 59341, ГОСТ Р МЭК 61508-1, ГОСТ Р МЭК 61508-2, ГОСТ IEC 61508-3, ГОСТ Р МЭК 61508-6, ГОСТ Р МЭК 61508-7, ГОСТ Р 70250).

А.10 Мобильные системы

Мобильные системы, как правило, представляют собой портативные вычислительные устройства без физических соединений для связи или питания (например, с питанием от аккумулятора). Мобильные устройства, как правило, достаточно компактны, чтобы их можно было держать в руках. Примерами мобильных устройств являются смартфоны, медиаплееры, смарт-часы и смарт-карты.

Типичные сопутствующие характеристики — аналогично как для встроенного ПО (см. А.7).

Для мобильных систем часто используют тесты, имитирующие потерю сетевого соединения (например, WiFi, сотовой связи, Bluetooth).

Тесты, основанные на потреблении энергии и отказе аккумулятора, являются обязательными для устройств с автономным питанием.

Тестирование на совместимость помогает подтвердить, что приложения могут работать с различными операционными системами и на различных устройствах.

А.11 Системы для работы в режиме реального времени

Системы для работы в режиме реального времени реагируют на внешние события в течение ограниченного промежутка времени (например, система предотвращения столкновений автомобилей). Системы для работы в режиме реального времени часто используют параллелизм для достижения высокой производительности в реальном времени.

Сопутствующие характеристики отсутствуют.

Специальное тестирование необходимо для проверки на наличие сбоев из-за несовершенства проектирования или реализации параллелизма. Тестирование производительности обычно используется для подтверждения соблюдения ограничений при функционировании ПО в режиме реального времени (см. ГОСТ Р МЭК 61508-1, ГОСТ Р МЭК 61508-2, ГОСТ IEC 61508-3, ГОСТ Р МЭК 61508-6, ГОСТ Р МЭК 61508-7, ГОСТ Р 70250).

А.12 Регулируемые системы

В некоторых областях строго необходимо обеспечение соответствия систем конкретным требованиям. Эти требования часто документируют как часть какого-либо нормативного документа. Правила могут быть введены правительством либо может существовать отраслевое саморегулирование. Системы могут находиться под регулированием в целях достижения определенного уровня безопасности (см. А.13) или для обеспечения соблюдения требований законодательства, например в сфере промышленной безопасности [1] — [11].

Сопутствующие характеристики подлежат регулированию.

Часто для проверки соблюдения стандартов проводят аудит.

Нормативные акты часто выступают как источник требований к тестированию ПО (см. ГОСТ Р МЭК 61069-2, ГОСТ Р МЭК 61069-3, ГОСТ Р МЭК 61069-4, ГОСТ Р МЭК 61069-5, ГОСТ Р МЭК 61069-6, ГОСТ Р МЭК 61069-7, ГОСТ Р МЭК 61069-8, ГОСТ Р МЭК 61508-1, ГОСТ Р МЭК 61508-2, ГОСТ IEC 61508-3, ГОСТ Р МЭК 61508-6, ГОСТ Р МЭК 61508-7).

А.13 Системы, связанные с безопасностью

Системы, связанные с безопасностью, могут угрожать жизни, здоровью, имуществу или окружающей среде человека в случае их отказа [1] — [11]. Эти системы, как правило, порождают особую группу рисков, связанных с

обеспечением безопасности. Такие риски часто выявляют на основе анализа безопасности, проводимого на ранних стадиях разработки.

Типичные сопутствующие характеристики подлежат регулированию (см. А.12).

Для проверки противодействия существующим угрозам тестирование может использовать моделирование и прогнозирование рисков. Тестирование с искусственным внедрением ошибок также может быть использовано для проверки обработки ошибок системой (например, для проверки того, что один сбой не приводит к полному отказу системы). Более подробно см. ГОСТ Р 56921, ГОСТ Р 56922, ГОСТ Р 56939, ГОСТ Р 58412, ГОСТ Р 58494, ГОСТ Р 59334, ГОСТ Р 59341, ГОСТ Р 59991.

А.14 Научно-технические системы

ПО для научно-технических систем поддерживает проведение научно-исследовательских и опытно-конструкторских работ в организациях различных академий наук, промышленности, в ВУЗах, научно-исследовательских учреждениях и конструкторских бюро. Типичные системы собирают, управляют и анализируют данные, моделируют физический мир и поддерживают визуализацию результатов.

Научно-техническое программное обеспечение часто нацелено на конечного пользователя-ученого, который использует ПО для проведения научных исследований в своей предметной области. Таким образом, большая часть этого ПО является предметно-ориентированной.

Сопутствующие характеристики отсутствуют.

Тестирование ПО для научно-технических систем, как правило, требует специальных знаний в данной области.

Для сложных научно-технических алгоритмов часто используются исполняемые программные модели для выполнения раннего тестирования.

Оригинальность и сложность реализованных расчетных алгоритмов могут затруднять оценку ожидаемых результатов, поэтому может быть использовано тестирование с поиском по ключевым словам, а также с анализом метаморфической связи, когда учитывается влияние изменений во входных данных для тестовых примеров на ожидаемые выходные результаты.

А.15 Интернет-системы

Веб-приложения основаны на клиент-серверной архитектуре, где клиент работает в веб-браузере, а серверная часть часто содержит базу данных. Примерами являются системы розничных онлайн-продаж, онлайн-банкинг и ПО для онлайн-офиса.

Типичные сопутствующие характеристики относятся к мобильности — веб-сайты, оптимизированные для использования с мобильных устройств (см. А.10).

Из-за доступности веб-приложения для любого пользователя Интернета могут возникнуть проблемы, связанные с удобством использования приложения людьми с ограниченными возможностями. Как правило, эти проблемы решают с помощью тестирования на доступность.

Тестирование безопасности является обязательным из-за простоты доступа к приложению.

Тестирование нагрузки и времени отклика оказывается важным в случаях, когда веб-сайты одновременно используются большим количеством пользователей.

Приложение Б
(справочное)

Пример возможных участников тестирования

Существует множество названий для различных участников тестирования. Поэтому в приложении приведен пример возможных названий. Реальное лицо, которое может выполнять обязанности участника тестирования, будет нести ответственность за выполнение работ, описанных в разделе 4.

Роль каждого из перечисленных ниже участников тестирования может исполнять более чем один человек:

- архитектор тестов: определяет стратегию тестирования, среду тестирования и план тестирования. Определяет организационные требования к тестированию и помогает обеспечить соответствие этим требованиям;
- менеджер по тестированию: разрабатывает, управляет и помогает обеспечить соответствие процессу управления тестированием, а также планирует и контролирует процесс динамического тестирования;
- проектировщик тестов: разрабатывает артефакты для проектирования тестов и сценарии тестирования;
- разработчик моделей: разрабатывает тестовые модели (вербальные, математические, программные), используемые для создания контрольных примеров в рамках тестирования на основе моделей;
- разработчик инструментариев для автоматизированного тестирования: разрабатывает инструментарии и контрольные примеры/сценарии в рамках среды для автоматизированного тестирования;
- тестировщик: выполняет необходимые действия при проведении тестирования, получает результаты тестирования и завершает процессы, связанные с процессом тестирования;
- аналитик по тестированию: анализирует результаты тестирования на корректность, выявляет дефекты, ошибки, определяет наличие проблем, формулирует выводы и рекомендации;
- специалист по специфическим вопросам тестирования: является тестировщиком в части специфических вопросов. Выполняет действия по тестированию по определенным показателям (например таким, как удобство использования, доступность, надежность, безопасность, производительность конкретной системы).

Библиография

[1] Федеральный закон от 21 декабря 1994 г. № 68-ФЗ «О защите населения и территорий от чрезвычайных ситуаций природного и техногенного характера»

[2] Федеральный закон от 21 июля 1997 г. № 116-ФЗ «О промышленной безопасности опасных производственных объектов»

[3] Федеральный закон от 21 июля 1997 г. № 117-ФЗ «О безопасности гидротехнических сооружений»

[4] Федеральный закон от 10 января 2002 г. № 7-ФЗ «Об охране окружающей среды»

[5] Федеральный закон от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации»

[6] Федеральный закон от 9 февраля 2007 г. № 16-ФЗ «О транспортной безопасности»

[7] Федеральный закон от 22 июля 2008 г. № 123-ФЗ «Технический регламент о требованиях пожарной безопасности»

[8] Федеральный закон от 30 декабря 2009 г. № 384-ФЗ «Технический регламент о безопасности зданий и сооружений»

[9] Федеральный закон от 21 июля 2011 г. № 256-ФЗ «О безопасности объектов топливно-энергетического комплекса»

[10] Федеральный закон от 28 июня 2014 г. № 172-ФЗ «О стратегическом планировании в Российской Федерации»

[11] Федеральный закон от 26 июля 2017 г. № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации»

[12] Постановление Правительства Российской Федерации от 12 февраля 2021 г. № 171 «О Координационном центре Правительства Российской Федерации»

[13] ИСО/МЭК 23531:2020[1)] Системная и программная инженерия. Возможности программных инструментариев для организационного управления инцидентами (Systems and software engineering Capabilities of issue management tools)

[1)] Одновременно с введением в действие настоящего стандарта планируется введение в действие рекомендуемого к использованию ГОСТ Р 71303—2024 «Системная и программная инженерия. Возможности программных инструментариев для организационного управления инцидентами. Общие положения», разработанный с учетом основных положений ИСО/МЭК 23531:2020.

УДК 006.34:004.056:006.354 ОКС 35.080

Ключевые слова: программное обеспечение, система, тестирование программного обеспечения, подход к тестированию, тестовая среда

Редактор *Н.А. Аргунова*
Технический редактор *В.Н. Прусакова*
Корректор *О.В. Лазарева*
Компьютерная верстка *Е.А. Кондрашовой*

Сдано в набор 15.06.2024. Подписано в печать 25.06.2024. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 3,72. Уч.-изд. л. 2,90.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении в ФГБУ «Институт стандартизации»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.

gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru