

# **ПРАКТИЧЕСКАЯ РАБОТА.**

## **ПЛАН ТЕСТИРОВАНИЯ. СИСТЕМЫ УПРАВЛЕНИЯ**

### **ТЕСТИРОВАНИЕМ**

#### **1.1. Цель и задачи практической работы**

**Цель работы:** освоить основные принципы разработки документации на примере «Плана тестирования», опираясь на международные и российские стандарты и познакомиться с различными системами управления тестирования.

Для реализации практической работы необходимо решить следующие **задачи**:

1. Исследовать международные стандарты (например, ISO/IEC/IEEE 29119) для разработки тестовой документации.
2. Проанализировать российские нормативные документы (ГОСТы), применяемые при составлении плана тестирования.
3. Изучить типовую структуру и содержание плана тестирования в соответствии с требованиями как международных, так и российских стандартов.
4. Выделить ключевые разделы документа и их назначение.
5. На основе изученных стандартов и типовых разделов составить проект тест-плана для выбранного программного продукта или модуля.
6. Обеспечить документ полным и логичным изложением всех необходимых компонентов (идентификатор, тестируемые элементы, критерии прохождения тестов, график работ и т.д.).
7. Ознакомиться с различными системами управления тестированием, как отечественными (например, Test IT, ТестОпс), так и международными (TestRail, Zephyr, JIRA с плагинами для тестирования).
8. Провести сравнительный анализ функциональности, удобства использования и возможностей интеграции, выбранных TMS.
9. Практически реализовать разработанный тест-план, используя одну из систем управления тестированием.
10. Настроить рабочее пространство, загрузить тестовые случаи и план, выполнить тестирование с фиксацией результатов.
11. Сформировать итоговый отчёт, включающий описание процесса разработки тест-плана, сравнительный анализ стандартов и TMS, а также выводы и рекомендации по улучшению документации и процессов тестирования.

Решение этих задач позволит не только освоить принципы разработки документации по тестированию, но и получить практический опыт работы с системами управления тестированием, а также обеспечить соответствие документации требованиям международных и российских стандартов.

## 1.2. Теоретический раздел

### 1.2.1. Международные стандарты

Стандарты ISO/IEC/IEEE 29119 — это международные стандарты, регулирующие процесс тестирования программного обеспечения. Они разработаны совместно Международной организацией по стандартизации (ISO), Международной электротехнической комиссией (IEC) и Институтом инженеров по электротехнике и электронике (IEEE). Стандарты обеспечивают формальный подход к тестированию ПО, охватывая данный процесс, документацию и методы.

Элементы ISO/IEC/IEEE 29119:

1. ISO/IEC/IEEE 29119-1:2013 определяет основные термины, концепции и принципы тестирования ПО. Включает определения ключевых понятий, таких как уровни тестирования, стратегии и методы тестирования. Устанавливает основы, на которых строятся остальные части стандарта.

2. ISO/IEC/IEEE 29119-2:2013 описывает процессы тестирования, в частности те, которые применяются при разработке программного обеспечения. Включает жизненный цикл тестирования, начиная с планирования и заканчивая анализом результатов. Регламентирует роли и ответственность участников тестирования. Может использоваться как руководство для внедрения эффективного процесса тестирования в организации.

3. ISO/IEC/IEEE 29119-3:2013 определяет форматы и требования к документации, связанной с тестированием. Включает шаблоны для тест-планов, тестовых спецификаций, отчётов о тестировании и других документов. Помогает организациям formalизовать процесс тестирования и улучшить прозрачность работ.

4. ISO/IEC/IEEE 29119-4:2015 описывает методы тестирования, применимые на разных этапах тестирования ПО. Включает такие методы, как функциональное тестирование, нагружочное тестирование, тестирование безопасности и др. Определяет, как выбирать методы тестирования в зависимости от типа системы и требований.

Где применяются стандарты ISO/IEC/IEEE 29119?

В компаниях, занимающихся разработкой программного обеспечения, для унификации тестовых процессов. В организациях, работающих по стандартам ISO 9001, для обеспечения контроля качества ПО. В сертификационных испытаниях программных продуктов перед выходом на рынок. В критически важных сферах (авиация, медицина, финансы), где тестирование ПО требует строгого соблюдения стандартов.

### ***1.2.2. Российские стандарты***

Среди российских стандартов выделяют следующие документы:

1. ГОСТ 19.101-77 «Единая система программной документации. Правила оформления документации». Этот стандарт определяет общие требования к оформлению всей технической и эксплуатационной документации, в том числе и тест-планов. Он помогает обеспечить единообразие и структурированность документов.

2. ГОСТ Р ИСО/МЭК 9126-93 «Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению». Этот стандарт используется для оценки качества программного обеспечения. Его применение помогает формализовать критерии, по которым оценивается эффективность тестирования, и может служить основой для определения показателей успешности тест-плана.

3. ГОСТ Р 56920-2016 определяет перечень обязательных разделов для документов, связанных с тестированием (например, тест-плана), включая такие элементы, как идентификатор, описание тестируемых компонентов, критерии успешного прохождения тестов, требования к тестовой среде, распределение ответственности, график работ, анализ рисков и др.

Эти ГОСТы часто применяются в совокупности, позволяя создать комплексную документацию, отвечающую как требованиям качества, так и структурным стандартам. В зависимости от конкретных задач и области применения проекта могут использоваться дополнительные методические указания или внутренние стандарты организации.

### ***1.2.3. План тестирования***

План тестирования — это документ, описывающий цели, подходы, ресурсы и график запланированных тестовых активностей. Он определяет объекты тестирования, свойства для тестирования, задания, ответственных за задания, степень

независимости каждого тестировщика, тестовое окружение, метод проектирования тестов, используемые критерии входа и выхода, а также любые риски, требующие планирования на случай чрезвычайных обстоятельств.

Ниже представлена краткая характеристика каждого из разделов плана тестирования.

1. Идентификатор тестового плана. Уникальный номер или код, позволяющий однозначно идентифицировать данный тест-план среди других документов проекта.

2. Ссылки на используемые документы. Перечень документов, на которые ссылается данный тест-план, включая спецификации требований, архитектурные описания, стандарты и руководства.

3. Введение. Общее описание целей и задач тестирования, области охвата, а также краткое изложение содержания документа.

4. Тестируемые элементы. Перечень компонентов или функций системы, которые подлежат тестированию.

5. Проблемы риска тестирования ПП. Идентификация потенциальных рисков, связанных с процессом тестирования программного продукта, и описание мер по их минимизации.

6. Особенности или свойства, подлежащие тестированию. Описание функциональных и нефункциональных характеристик системы, которые будут проверяться в ходе тестирования.

7. Особенности (свойства), не подлежащие тестированию. Перечень функций или характеристик системы, которые не будут тестироваться, с указанием причин таких исключений.

8. Подход. Описание стратегии и методов тестирования, включая используемые техники, инструменты и критерии покрытия.

9. Критерии смоук-тестирования. Условия, при которых система считается готовой для проведения более детального тестирования после первоначальной проверки основных функций.

10. Критерии прохождения тестов. Определение условий, при которых тесты считаются успешно пройденными или проваленными.

11. Критерии приостановки и возобновления работ. Условия, при которых тестирование может быть временно приостановлено, и требования для его последующего возобновления.

12. Тестовая документация. Перечень всех документов, которые будут созданы в процессе тестирования, включая тестовые сценарии, отчеты о дефектах и итоговые отчеты.

13. Основные задачи тестирования. Детальное описание задач, которые необходимо выполнить в рамках процесса тестирования.

14. Необходимый персонал и обучение. Определение ролей и обязанностей членов команды тестирования, а также описание потребностей в обучении или повышении квалификации.

15. Требования среды. Описание аппаратного и программного обеспечения, необходимого для проведения тестирования, включая конфигурации тестовых стендов и среды выполнения.

16. Распределение ответственности. Четкое определение обязанностей и зон ответственности каждого члена команды в процессе тестирования.

17. График работ (календарный план). Подробный план с указанием сроков выполнения каждой задачи, этапов тестирования и контрольных точек.

18. Риски и непредвиденные обстоятельства. Идентификация потенциальных рисков, которые могут повлиять на процесс тестирования, и разработка планов действий на случай их возникновения.

19. Утверждение плана тестирования. Список лиц, ответственных за утверждение данного плана, с указанием дат и подписей.

20. Глоссарий. Список терминов и определений, используемых в документе, для обеспечения общего понимания среди всех участников проекта.

Данная структура плана тестирования помогает систематизировать процесс тестирования и обеспечивает прозрачность и эффективность работ.

#### ***1.2.4. Системы управления тестированием (TMS)***

Системы управления тестированием (Test Management Systems, TMS) представляют собой специализированные программные решения, предназначенные для планирования, организации, исполнения и анализа тестирования программного обеспечения. Они обеспечивают централизованное хранение тестовой документации, автоматизацию создания тест-кейсов, контроль выполнения тестов и формирование отчетности. Применение TMS способствует повышению прозрачности тестовых процессов, ускоряет выявление дефектов и улучшает коммуникацию между участниками проекта.

Например, в крупных проектах, таких как банковские системы или корпоративные ERP-решения, использование TMS позволяет распределять задачи между тестировщиками, автоматически запускать регрессионное тестирование и отслеживать результаты в реальном времени.

Основные функциональные возможности современных систем управления тестированием включают:

1. Создание, хранение и редактирование тест-планов, тест-кейсов, сценариев и отчетов. Это позволяет стандартизировать подход к тестированию и обеспечить согласованность всей документации.

2. Формирование тестовых наборов (Test Suites), циклов тестирования и календарного плана, что помогает оптимально распределить ресурсы и контролировать сроки выполнения работ.

3. Регистрация прохождения тест-кейсов, автоматическое обновление статусов, создание задач по обнаруженным дефектам и синхронизация с системами отслеживания ошибок.

4. Генерацию подробных отчетов, метрик и диаграмм, которые позволяют оценить качество тестирования и выявить узкие места в процессе разработки.

5. Связь с системами контроля версий, инструментами CI/CD и системами управления проектами, что обеспечивает единый информационный поток между всеми участниками жизненного цикла ПО.

Применение TMS предоставляет значимые преимущества, подтверждённые практическим опытом ведущих компаний.

Все данные о тестировании собраны в одном месте, что облегчает доступ для всех членов команды, независимо от их географического расположения.

Использование шаблонов и предопределённых процессов обеспечивает соответствие международным (ISO/IEC/IEEE 29119) и отечественным стандартам (например, ГОСТ Р 56920-2016). Это гарантирует единообразие в подходах к тестированию и способствует высокому качеству выпускаемого продукта.

С помощью аналитических инструментов TMS можно оперативно выявлять проблемные области и принимать корректирующие меры, что снижает риск выхода продукта с критическими дефектами.

Автоматическое выполнение тестов, регистрация результатов и интеграция с системами отслеживания ошибок сокращают время на выполнение рутинных операций, позволяя тестировщикам сосредоточиться на сложных сценариях.

Благодаря отчетности и возможности обмена информацией в реальном времени TMS улучшают взаимодействие между разработчиками, тестировщиками и менеджерами проектов.

Современные методологии разработки, такие как Agile и DevOps, требуют высокой гибкости и оперативности в управлении тестированием. TMS интегрируют тестовые процессы в единый цикл разработки, позволяя быстро реагировать на изменения, обеспечивать коллaborацию, поддерживать непрерывное улучшение ПО.

В ряде международных компаний интеграция ТМС с CI/CD инструментами позволила сократить время на обнаружение и исправление дефектов на 40%, а улучшенная отчетность способствовала оптимизации процесса принятия решений на всех уровнях управления проектом.

### **1.2.5. Примеры систем управления тестированием**

Среди мировых систем управления тестированием можно выделить:

1. TestRail — это интуитивно понятная и гибкая система управления тестированием, позволяющая создавать тестовые планы, кейсы и наборы тестов. Продукт интегрируется с популярными баг-трекерами (например, Jira) и CI/CD-инструментами, что делает его удобным решением для agile-команд.

Официальный сайт: <https://www.gurock.com/testrail/>

2. Zephyr — комплексное решение для управления тестированием, которое предлагает возможности для планирования, выполнения тестов и формирования подробной отчетности. Продукт интегрируется с Jira и другими системами, что облегчает отслеживание дефектов и управление тестовыми циклами.

Официальный сайт: <https://www.getzephyr.com/>

3. qTest — мощная платформа для управления тестированием от компании Tricentis. Она предлагает масштабируемые возможности для планирования, управления тестовыми сценариями, интеграции с CI/CD и анализа результатов тестирования, что делает её популярной среди крупных организаций, работающих в Agile-среде.

Официальный сайт: <https://www.tricentis.com/products/qtest/>

**Примечание:** Зарубежные приложения с большой долей вероятности без использования специального программного обеспечения будут недоступны пользователям на территории Российской Федерации. Представлены в практикуме для примера.

Среди российских систем управления тестированием можно выделить:

1. Test IT — отечественная система управления тестированием, разработанная для поддержки создания и ведения тестовой документации, планирования тестовых активностей и интеграции с системами отслеживания ошибок. Продукт ориентирован на повышение эффективности тестирования в российских компаниях.

Официальный сайт: <https://testit.ru/>

2. ТестОпс — российская ТМС, позволяющая организовать полный цикл тестирования от планирования и создания тест-кейсов до их исполнения и

формирования аналитических отчетов. Система поддерживает интеграцию с другими инструментами разработки и управления проектами, что упрощает совместную работу команды.

Официальный сайт: <https://testops.ru/>

### **1.3. Описание работы**

Практическая работа выполняется индивидуально.

#### ***1.3.1. Часть 1. Разработка плана тестирования***

Для выполнения данного шага необходимо взять приложение, разработанное в практической работе №3, и подробно описать для него каждый пункт плана тестирования.

Обязательные пункты плана тестирования:

1. Идентификатор тестового плана.
2. Ссылки на используемые документы.
3. Введение.
4. Тестируемые элементы.
5. Проблемы риска тестирования ПП.
6. Особенности или свойства, подлежащие тестированию.
7. Особенности (свойства), не подлежащие тестированию.
8. Подход.
9. Критерии смоук-тестирования.
10. Критерии прохождения тестов.
11. Критерии приостановки и возобновления работ.
12. Тестовая документация.
13. Основные задачи тестирования.
14. Необходимый персонал и обучение.
15. Требования среды.
16. Распределение ответственности.
17. График работ (календарный план).
18. Риски и непредвиденные обстоятельства.
19. Утверждение плана тестирования.
20. Глоссарий.

### **1.3.2. Часть 2. Изучение концепции TMS**

Для реализации второй части практической работы необходимо выполнить ряд задач:

1. Провести анализ представленных на российском рынке системы управления тестированием. Выбрать одну из них для более глубокого изучения и знакомства с функционалом.
2. Выбрать приложение для проведения тестирования (может быть использован программный продукт, разработанный как в рамках данной дисциплины, так и из любой другой).
3. Разработать не менее пяти тестовых случаев для выбранного модуля или функции приложения.
4. Определить шаги выполнения, ожидаемые результаты и приоритеты для каждого теста.
5. Зафиксировать результаты выполнения тестов, отмечая успешные и неуспешные случаи.
6. Настроить интеграцию TMS с выбранной системой отслеживания дефектов.
7. При обнаружении дефектов в процессе тестирования создавать соответствующие задачи в системе отслеживания дефектов напрямую из TMS.
8. Сформировать отчёты о проделанном тестировании, используя возможности TMS.
9. Проанализировать полученные данные и подготовить рекомендации по улучшению процесса тестирования.

## **1.4. Итоговый отчёт**

По результатам индивидуальной работы отчёт должен содержать:

1. Титульный лист, включающий в себя наименование работы, автора, дату выполнения, название учебной дисциплины.
2. Документирование каждого этапа работы.
3. Описание теоретических основ, практических примеров, результатов тестирования и анализа.
4. Заключение, включающее общую оценку качества программного продукта и документации, выводы о проделанной работе.
5. Приложения (при необходимости): дополнительные материалы (коды, логи тестирования, скриншоты и т. д.).

Отчёт должен быть оформлен в соответствии с ГОСТ 19.401-78 и ГОСТ 34.602-2020. Требования включают стандарты на титульный лист, использование единообразного шрифта, оформление таблиц и диаграмм, наличие нумерации страниц.

Итоговая оценка работы будет зависеть от полноты отчёта, качества выполнения задания, соответствия оформления стандартам и презентации результатов на защите.