https://qarocks.ru/what-is-a-test-plan/

Что такое тест-план и как его создать?

[Оставьте комментарий](https://qarocks.ru/what-is-a-test-plan/#respond) / [Тестовая документация](https://qarocks.ru/category/testdocumentation/) / От [anastasia](https://qarocks.ru/author/anastasiya/" \o "Смотреть все записи от anastasia)

**Содержание:**

* [Что такое тест-план?](https://qarocks.ru/what-is-a-test-plan/#what-is-a-test-plan)
* [В чем важность тест-плана?](https://qarocks.ru/what-is-a-test-plan/#importance-of-test-plan)
* [Как написать тест-план?](https://qarocks.ru/what-is-a-test-plan/#how-to-write-a-test-plan)
  + [Анализ продукта](https://qarocks.ru/what-is-a-test-plan/#analyze-the-product)
  + [Разработка стратегии тестирования](https://qarocks.ru/what-is-a-test-plan/#design-the-test-strategy)
  + [Определение объема тестирования](https://qarocks.ru/what-is-a-test-plan/#define-the-test-scope)
  + [Определение видов тестирования](https://qarocks.ru/what-is-a-test-plan/#define-test-types)
  + [Документирование рисков](https://qarocks.ru/what-is-a-test-plan/#risk-documentation)
  + [Создание тестовой логистики](https://qarocks.ru/what-is-a-test-plan/#create-test-logistics)
  + [Определение цели тестирования](https://qarocks.ru/what-is-a-test-plan/#define-test-objectives)
  + [Определение критериев тестирования](https://qarocks.ru/what-is-a-test-plan/#define-test-criteria)
* [Планирование ресурсов](https://qarocks.ru/what-is-a-test-plan/#resource-planning)
* [Планирование тестовой среды](https://qarocks.ru/what-is-a-test-plan/#test-environment-planning)
* [Оценка затрат](https://qarocks.ru/what-is-a-test-plan/#test-estimation)
* [Результаты тестирования](https://qarocks.ru/what-is-a-test-plan/#test-deliverables)

**Что такое тест-план?**

**Тест-план** – это подробный документ, описывающий стратегию, цели, результаты и ресурсы, необходимые для проведения тестирования ПО. Тест-план помогает определить усилия, необходимые для проверки качества тестируемого приложения. Он также служит основой для тестирования ПО в рамках определенного процесса, который тщательно контролируется и управляется тест-менеджером.

**В чем важность тест-плана?**

Составление тест-плана имеет множество преимуществ:

* Помогает разработчикам, бизнес-менеджерам, заказчикам, понять детали тестирования.
* Тест-план – это как свод правил, которым необходимо следовать в тестировании ПО.
* Оценка затрат, объем и стратегия тестирования документируются в тест-плане.
* Тест-план может быть повторно использован в других проектах.

**Как написать тест-план?**

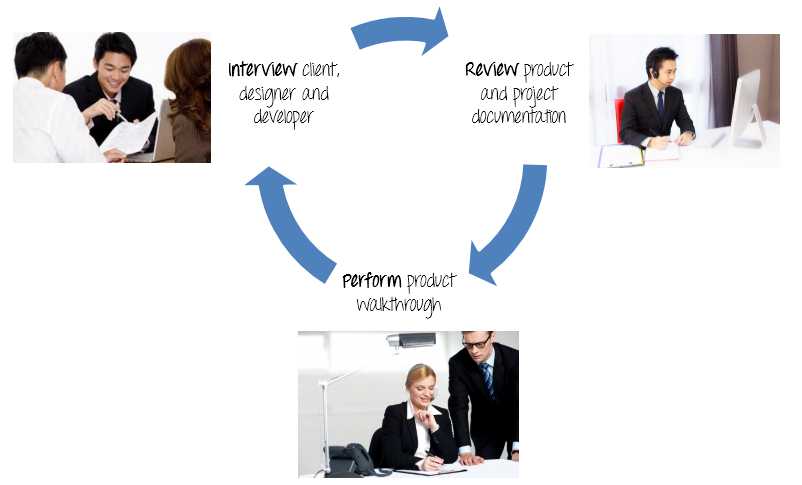
Вы уже знаете, что составление тест-плана является наиболее важной задачей процесса управления тестированием. Чтобы создать этот документ в соответствии с требованиями стандарта [IEEE 829](https://standards.ieee.org/ieee/829/3787/), необходимо выполнить следующие шаги:

1. Проанализировать продукт
2. Разработать стратегию тестирования
3. Определить цели тестирования
4. Определение критерии тестирования
5. Запланировать ресурсы
6. Настроить тестовую среду
7. Распределить задачи
8. Определить результаты тестирования
9. **Шаг 1. Анализ продукта**

Как можно протестировать продукт, не имея о нем никакой информации? Ответ: невозможно. Прежде чем тестировать продукт, его необходимо тщательно изучить.

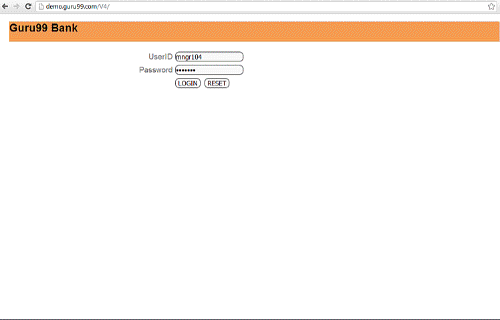
Например, мы тестируем банковский сайт. Для этого сперва необходимо изучить клиентов и конечных пользователей, чтобы узнать их потребности и ожидания от приложения.

* Кто будет пользоваться сайтом?
* Для чего он нужен?
* Как он будет работать?
* Какие программные/аппаратные средства использует продукт?  
  Для анализа сайта можно использовать следующий подход:



Теперь применим полученные знания к реальному продукту:

Проанализируем банковский сайт [http://demo.guru99.com/V](http://demo.guru99.com/V4)[4](https://demo.guru99.com/V4/)[.](http://demo.guru99.com/V4)



Вам следует ознакомиться с этим сайтом, а также изучить документацию. Это поможет вам понять все возможности сайта, а также то, как им пользоваться. Если вам что-то неясно, вы можете задать свои вопросы заказчику, разработчикам, дизайнеру, чтобы получить дополнительную информацию.

**Шаг 2. Разработка стратегии тестирования**

Разработка [стратегии тестирования](https://qarocks.ru/testing-strategy/) (Test Strategy) является важнейшим этапом составления тест-плана. Test Strategy – это высокоуровневый документ, который обычно разрабатывается тест-менеджером. В этом документе определяются:

* Цели тестирования и средства их достижения.
* Усилия и затраты на тестирование.

Вернемся к вашему проекту: вам необходимо разработать стратегию тестирования для проверки банковского сайта. Для этого необходимо выполнить следующие шаги:

**Шаг 2.1 Определение объема тестирования**

Перед началом любой тестовой деятельности необходимо определить объем тестирования. Для этого необходимо хорошо продумать следующее:

* Компоненты системы, которые будут протестированы (аппаратное и программное обеспечение, промежуточное ПО и т.д.).
* Компоненты системы, которые не будут тестироваться.

Определение объема тестирования очень важно для всех заинтересованных сторон. **Точное определение объема поможет вам:**

* Дать всем уверенную и точную информацию о проводимом тестировании.
* Все участники проекта будут иметь четкое представление о том, что тестируется, а что нет.

**Для определения объема проекта необходимо учесть следующее:**

* Точные требования заказчика
* Бюджет проекта
* Особенности продукта
* Навыки вашей команды тестировщиков.

**Теперь необходимо четко определить, что тестировать, а что нет:**

* В соответствии с требованиями к программному обеспечению, проект Guru99 Bank ориентирован только на тестирование функционала и UI сайта.
* [Нефункциональное тестирование](https://qarocks.ru/non-functional-testing/), такое как [стресс-тестирование](https://qarocks.ru/stress-testirovanie/), [тестирование производительности](https://qarocks.ru/when-start-load-testing/) или [базы данных](https://qarocks.ru/database-testing-process/) не будет проведено (это может быть связано с ограничениями времени, ресурсов или просто со стратегическим решением команды проекта.).

**Сценарий проблемы:**

Заказчик хочет, чтобы вы протестировали API. Но бюджет проекта не позволяет это сделать. Что же делать в таком случае?

Нужно убедить заказчика в том, что [тестирование API](https://qarocks.ru/postman-testirovanie-api/) – это дополнительная работа, которая потребует значительных ресурсов. Приведите ему данные, подтверждающие ваши факты.

**Шаг 2.2 Определение видов тестирования**

Каждый [вид тестирования](https://qarocks.ru/types-of-software-testing/) предназначен для выявления определенного типа ошибок в продукте. Но все виды тестирования направлены на достижение одной общей цели – как можно раньше обнаружить дефекты в ПО.

Наиболее распространенные виды тестирования описаны на следующем рисунке:

Существует множество видов тестирования для проверки программного продукта. Вашей команде не хватит сил, чтобы справиться со всеми видами тестирования. Поэтому тест-менеджер должен определить приоритетные виды тестирования исходя из требований к вашему проекту.

**Шаг 2.3 Документирование рисков**

Риск – это возможное событие в будущем, которое может привести к определенным проблемам.

В тест-плане все риски должны быть задокументированы.

|  |  |
| --- | --- |
| Риск | Снижение рисков |
| У команды QA нет необходимых навыков для тестирования сайтов. | Запланируйте учебный курс для повышения квалификации своих сотрудников. |
| График проекта слишком жесткий; трудно завершить проект в срок. | Определите приоритетные виды тестирования для вашего проекта. |
| Тест-менеджер обладает низкими навыками управления. | Запланировать тренинг по лидерству для менеджера. |
| Отсутствие сотрудничества между членами команды негативно сказывается на производительности ваших сотрудников. | Поощряйте каждого члена команды в выполнении его задачи и вдохновляйте их на большие достижения. |
| Неправильная оценка бюджета и перерасходы. | Предварительно определите объем работ, тщательно запланируйте проект и постоянно контролируйте прогресс. |

**Шаг 2.4 Создание тестовой логистики**

В разделе “Тестовая логистика” тест-менеджер должен ответить на следующие вопросы:

* Кто будет проводить тестирование?
* Когда следует проводить тестирование?

**Кто будет проводить тестирование?**

Вы можете не знать точных имен тестировщиков, которые будут проводить тестирование, но вид тестировщика важно определить.

Чтобы выбрать подходящего участника для выполнения поставленной задачи, необходимо учесть, соответствует ли его квалификация требованиям задачи, а также оценить бюджет проекта. Неправильный выбор сотрудника для выполнения задачи может привести к неудаче или задержке проекта.

Для эффективного проведения тестирования ПО лучше всего подходят тестировщики, обладающие следующими качествами:

* Способность понимать требования заказчика
* Стремление к качеству
* Внимание к деталям
* Хорошее сотрудничество со всеми членами команды

**Когда следует проводить тестирование?**

Тестирование должно быть согласовано с разработкой.

К нему можно приступать, когда у вас будут все необходимые элементы, как показано на следующем рисунке:

**Шаг 3. Определение цели тестирования**

Цель тестирования – найти как можно больше дефектов в программном обеспечении; убедиться, что тестируемое приложение не содержит ошибок перед релизом.

Для определения целей тестирования необходимо выполнить следующие 2 шага:

1. Перечислить все аспекты ПО (функциональность, производительность, графический интерфейс), которые необходимо протестировать.
2. Определить цель или задачи тестирования на основе вышеперечисленных характеристик.

Давайте применим эти шаги для определения цели тестирования сайта Guru99 Bank:

Вы можете выбрать метод “сверху вниз” (Top-Down) для поиска функций сайта, которые необходимо протестировать. В этом подходе тестируемое приложение разбивается на компоненты и подкомпоненты.

Вы можете создать Mind Map для поиска функций сайта следующим образом:

На этом рисунке показаны все функции, которыми обладает сайт Guru99.

Исходя из вышеперечисленных функций, можно определить цель тестирования сайта Guru99:

* Проверить, что функциональность сайта Guru99 (счет, депозит…) работает так, как ожидается, без каких-либо ошибок и недочетов в реальной бизнес-среде.
* Проверить, что UI удовлетворяет потребностям клиента.
* Проверить удобство использования сайта. Удобны ли функциональные возможности сайта для пользователя или нет?

**Шаг 4. Определение критериев тестирования**

Критерии тестирования (Test Criteria) – это стандарт или правило, по которому оцениваются результаты тестирования. Существует 2 типа критериев тестирования:

**Критерии приостановки (Suspension Criteria)**

Определите критерии приостановки тестов. Если они будут выполнены, текущий цикл тестирования будет приостановлен до их устранения.

**Пример тест-плана**: Если члены вашей команды сообщают, что 40% тест-кейсов не работают, то следует приостановить тестирование до тех пор, пока команда разработчиков не исправит все баги.

**Критерии выхода (Exit Criteria)**

Это критерии, свидетельствующие об успешном завершении этапа тестирования. Критерии выхода – это целевые результаты тестирования, которые необходимы для перехода к следующей фазе разработки.

Для определения критериев завершения тестирования используют коэффициент выполнения (run rate) и прохождения тестов(pass rate):

* Run rate – это отношение количества выполненных тест-кейсов к общему количеству. Например, необходимо выполнить 120 тест-кейсов, но тестировщик выполнил только 100, поэтому коэффициент выполнения равен 100/120 = 0,83 (83%).
* Pass rate – это отношение числа пройденных тест-кейсов к выполненным. Например, из 100 выполненных тест-кейсов только 80 прошли, поэтому коэффициент прохождения равен 80/100 = 0,8 (80%).

**Шаг 5. Планирование ресурсов**

Планирование ресурсов – это подробное описание всех видов ресурсов, необходимых для выполнения задач проекта. Ресурсами могут быть люди, оборудование и материалы, необходимые для успешной реализации проекта.

Планирование ресурсов является важным фактором планирования тестирования, поскольку оно помогает определить количество ресурсов (сотрудников, оборудования…), которые будут задействованы в проекте.

**Человеческие ресурсы**

В следующей таблице представлены различные члены команды тестирования:

|  |  |  |
| --- | --- | --- |
| № | Участник | Задачи |
| 1 | Тест-менеджер | Управление всем проектом. Определение необходимых ресурсов для проекта. |
| 2 | Тестировщик | Выявление и описание подходящих техник/инструментов/архитектуры автоматизации тестирования. Проверка и оценка подхода к тестированию. Выполнение тестов, запись результатов, отчет о дефектах. |
| 3 | Разработчик в области тестирования | Написание тест-кейсов, тестового набора и т.д. |
| 4 | Администратор тестирования | Создание и обслуживание тестовой среды. |
| 5 | Члены SQA (Software Quality Assurance) | Ответственность за обеспечение качества приложения. Проверка соответствия процесса тестирования установленным требованиям. |

**Системные ресурсы**

Для тестирования веб-приложения необходимо спланировать ресурсы в соответствии со следующей таблицей:

|  |  |  |
| --- | --- | --- |
| № | Ресурсы | Описание |
| 1 | Сервер | Установка тестируемого веб-приложения. Включает в себя отдельный веб-сервер, сервер баз данных и сервер приложений. |
| 2 | Инструмент тестирования | Инструмент тестирования предназначен для автоматизации тестирования, имитации работы пользователя, генерации результатов тестов. Существует множество инструментов тестирования, которые можно использовать, например Selenium, QTP… и т.д. |
| 3 | Сеть | Вам нужна сеть (LAN и Интернет), чтобы имитировать реальные условия работы приложения. |
| 4 | Компьютер | ПК, который пользователи используют для подключения к веб-серверу. |

**Шаг 6 Планирование тестовой среды**

**Что такое тестовая среда?**

[Тестовая среда](https://qarocks.ru/test-environment-setup/) – это совокупность программных и аппаратных средств, с помощью которых команда тестировщиков будет выполнять тесты.

**Как настроить тестовую среду?**

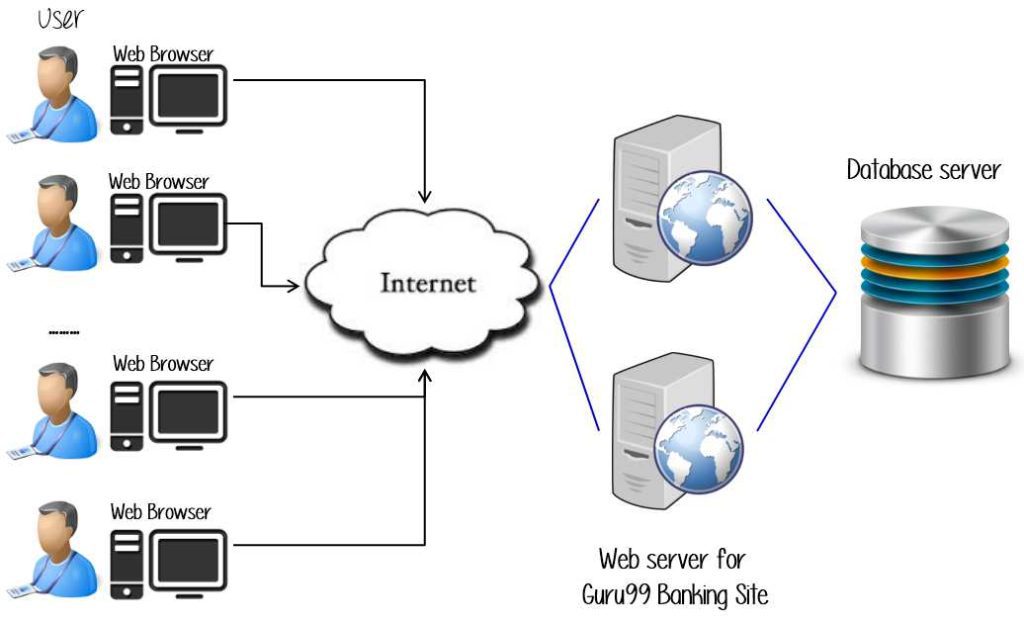
Вернемся к вашему проекту, как настроить тестовое окружение для банковского сайта?

Чтобы решить эту задачу, необходимо тесное сотрудничество между командой тестирования и командой разработки.

Для того, чтобы получить четкое представление о тестируемом веб-приложении, следует задать разработчику следующие вопросы:

* Какое максимальное количество пользовательских подключений данный веб-сайт может обрабатывать одновременно?
* Какие характеристики аппаратного и программного обеспечения требуются для установки этого веб-сайта?
* Нужны ли на компьютере пользователя какие-то особые настройки для просмотра сайта?

На следующем рисунке описана тестовая среда банковского сайта guru99.com:



**Шаг 7 Оценка затрат**

На данном этапе вы должны разбить весь проект на небольшие задачи и оценить усилия и затраты для каждой задачи, как показано ниже:

|  |  |  |
| --- | --- | --- |
| Задача | Участники | Оценка усилий |
| Создание спецификации теста | Разработчик тестов | 170 человеко-часов |
| Выполнение тестов | Тестировщик, администратор тестирования | 80 человеко-часов |
| Отчет о тестировании | Тестировщик | 10 человеко-часов |
| Тестовая поставка |  | 20 человеко-часов |
| Всего |  | 280 человеко-часов |

Затем составляется график выполнения этих задач.

Составив надежный график проведения тестирования, тест-менеджер может использовать его как инструмент для контроля проекта, управления превышением затрат.

Для этого ему необходимы такие данные, как:

* **Рабочие дни и крайний срок для реализации проекта:** Рабочие дни, сроки выполнения проекта, доступность ресурсов – факторы, влияющие на график.
* **Оценка проекта:**Основываясь на оценке проекта, тест-менеджер знает, сколько времени потребуется для завершения проекта. Таким образом, он может составить соответствующий график проекта.
* **Риски проекта:** Понимание рисков помогает тест-менеджеру добавить в график проекта дополнительное время, чтобы устранить риски.

Предположим, что начальник хочет завершить проект Guru99 за один месяц, вы уже оценили усилия для каждой задачи. Вы можете составить график тестирования следующим образом:

**Шаг 8 Результаты тестирования**

Результаты тестирования (Test Deliverables) – это список всех документов, инструментов и других компонентов, которые должны быть разработаны и поддерживаться в процессе тестирования.

На каждой фазе жизненного цикла разработки программного обеспечения существуют свои результаты тестирования.

**До начала фазы тестирования:**

* Тест-план.
* Документы с разработанными тест-кейсами.
* Спецификации проектирования теста.

**Во время тестирования:**

* Тестовые сценарии
* Симуляторы.
* Тестовые данные.
* Логи.

**После завершения цикла тестирования:**

* Отчет о дефектах.
* Примечания к релизу.