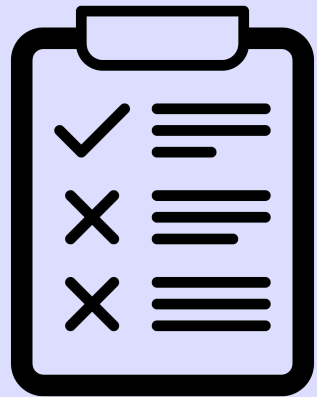


ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ

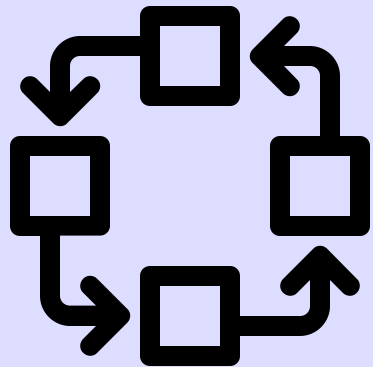




Тестирование

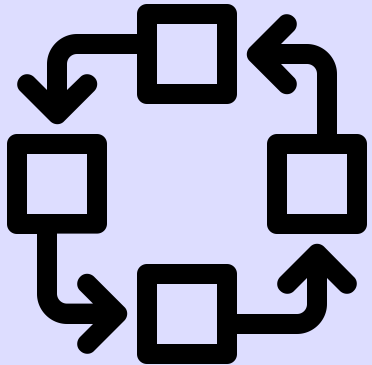
производительности проводится для обеспечения бесперебойной работы и снижения стоимости владения ПО на всех этапах жизненного цикла решения

ЭТАПЫ



1. Анализ системы и подбор требований
2. Подготовка стратегии
3. Настройка генератора нагрузки
4. Проведение мониторинга серверов и генератора нагрузки
5. Подготовка тестовых данных

ЭТАПЫ



6. Разработка нагрузочных скриптов
7. Предварительные запуски тестов
8. Проведение тестирования
9. Анализ результатов и подготовка отчёта

ЭТАП 1 – АНАЛИЗ СИСТЕМЫ И ПОДБОР ТРЕБОВАНИЙ

Анализ системы включает изучение ее свойств, особенностей и режима работы. С его помощью инженеры:

- воспроизводят максимально объективные шаблоны поведения пользователей и профиля нагрузки
- определяют необходимое количество тестовых данных
- выявляют потенциально уязвимые места ПО
- настраивают способы мониторинга системы

ЭТАП 1 – АНАЛИЗ СИСТЕМЫ И ПОДБОР ТРЕБОВАНИЙ

Требования – это критерии, которым система соответствует с технической точки зрения.

Таковыми *критериями* будут следующие измеряемые **метрики**:

Метрика	Определение
Количество пользователей	Программа, которая при выполнении запросов к приложению действует как реальный пользователь.
Время отклика	Время, затраченное системой на выполнение запроса пользователя.
Запросы в секунду	Измерение количества запросов, отправляемых на целевой сервер.
Транзакции в секунду	Измерение количества транзакций, отправляемых на целевой сервер.

Метрика	Определение
Процент ошибок	Процент ошибок от общего числа ответов в единицу времени.
Процессор	Процентное соотношение времени, которое процессор тратит на выполнение рабочего потока.
Оперативная память	Количество МВ в физической памяти, свободное для всех программ и процессов.
Жесткий диск	Информация о процентах использования дискового пространства.

ЭТАП 2 – ПОДГОТОВКА СТРАТЕГИИ

Стратегия разрабатывается на основе детального *анализа* ПО и описывает расширенный подход к **тестированию производительности**.

Анализ включает всю собранную информацию о системе, целях тестирования, требованиях к ПО, *конфигурации тестового стенда*, мониторинге, сценариях поведения пользователей, *профиле и модели нагрузки*, инструментах тестирования производительности.

ЭТАП 2 – ПОДГОТОВКА СТРАТЕГИИ

Конфигурация тестового стенда

Для получения достоверных результатов проверка производительности проводится на отдельном окружении с параметрами и конфигурацией, которые максимально приближены к характеристикам реальной системы.

Части тестового стенда	Характеристики
Аппаратное обеспечение	<ul style="list-style-type: none">▪ центральный процессор (количество ядер, тип, частота)▪ оперативная память (объём, тип)▪ жёсткий диск (тип, скорость)
Программное обеспечение	<ul style="list-style-type: none">▪ ОС (версия и пакеты обновления)▪ сервер приложения (версия и патчи)▪ СУБД (версия и тип)
Сетевое оборудование	<ul style="list-style-type: none">▪ топология сети▪ пропускная способность▪ протокол передачи данных
Приложение	<ul style="list-style-type: none">▪ архитектура▪ БД (структура, данные)▪ ПО, необходимое для работы приложения

ЭТАП 2 – ПОДГОТОВКА СТРАТЕГИИ

Разработка профиля и модели нагрузки

Во время проведения тестирования производительности собирается статистика по использованию ПО, которая нужна для создания **профиля нагрузки** – модели поведения пользователей.

ЭТАП 2 – ПОДГОТОВКА СТРАТЕГИИ

Разработка профиля и модели нагрузки

Модель нагрузки для одного и того же теста бывает быть разной. Например, можно использовать модель нагрузки «+1 пользователь каждые 5 секунд» либо добавить всех пользователей сразу.

ЭТАП 2 – ПОДГОТОВКА СТРАТЕГИИ

Выбор инструмента для нагрузочного тестирования

Использование инструментов нагрузочного тестирования позволяет записать процесс обмена данными между сервером и браузером.

ЭТАП 3 – НАСТРОЙКА ГЕНЕРАТОРА НАГРУЗКИ

Для качественного проведения тестирования производительности инструмент устанавливается на **генератор нагрузки** – виртуальную или физическую машину, расположенную максимально близко к серверу(-ам) приложения. Это снижает искажения при подаче нагрузки, вызванные задержками сети.

ЭТАП 4 – ПРОВЕДЕНИЕ МОНИТОРИНГА СЕРВЕРОВ И ГЕНЕРАТОРА НАГРУЗКИ

Информация, полученная при мониторинге всех серверов и ресурсов системы, используется при анализе результатов.

Мониторинг можно проводить с помощью специальных **утилит** для отслеживания потребления аппаратных ресурсов ПО.

ЭТАП 5 – ПОДГОТОВКА ТЕСТОВЫХ ДАННЫХ

Как правило, подходящий метод устанавливается на этапе анализа особенностей ПО, когда совместно со стороной разработки обсуждаются типы тестовых данных и их структура.

ЭТАП 5 – ПОДГОТОВКА ТЕСТОВЫХ ДАННЫХ

4 способа подготовки данных:

- Код

С помощью скриптов на различных языках программирования можно создавать пользователей, пароли для пользователей и другие необходимые значения.

- SQL-запросы

Один из способов заполнения базы данных – SQL-запросы.

ЭТАП 5 – ПОДГОТОВКА ТЕСТОВЫХ ДАННЫХ

4 способа подготовки данных:

- API-запросы

Если необходимо наполнить базу товарами, информацией о пользователях, то используются API-запросы.

ЭТАП 5 – ПОДГОТОВКА ТЕСТОВЫХ ДАННЫХ

4 способа подготовки данных:

- Интерфейс

Можно создать скрипт, который будет проходить шаги реального пользователя при регистрации. В таком случае есть смысл использовать встроенные функции нагрузочных инструментов, позволяющие провести рандомизацию значений по конкретному алгоритму.

ЭТАП 6 – РАЗРАБОТКА НАГРУЗОЧНЫХ СКРИПТОВ

Процесс разработки скриптов состоит из трех шагов:

- Изучение сценариев
- Создание тест-кейсов
- Отладка сценариев

ЭТАП 7 – ПРЕДВАРИТЕЛЬНЫЕ ЗАПУСКИ ТЕСТОВ

Предварительные запуски тестов нужны для проверки корректности работы подготовленных нагрузочных скриптов, а также для поиска оптимальной модели нагрузки на систему.

Они позволяют определить хватит ли заложенных мощностей генератора для проведения полноценных тестов.

ЭТАП 8 – ПРОВЕДЕНИЕ ТЕСТИРОВАНИЯ

На данном этапе выполняются следующие виды проверок:

1. **СТРЕСС-ТЕСТ** проходит с постепенно увеличивающейся на сервер нагрузкой и возрастает до тех пор, пока не будет достигнут один из критериев остановки теста:

- превышение требуемых значений отклика в несколько раз
- достижение критического уровня использования аппаратных ресурсов (ЦП > 80%, память > 90%)
- сбой системного ПО.

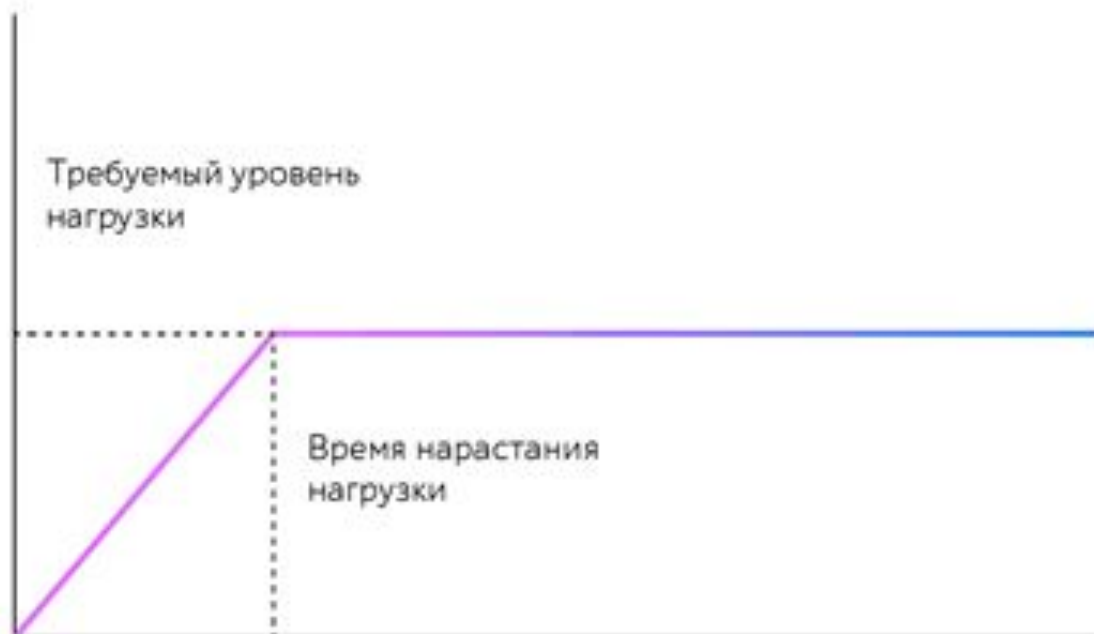
**Виртуальные
пользователи**



**Продолжительность
теста**

2. НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ проводится в течение длительного промежутка времени (4-8 ч.). Если в результате стресс-теста система не выдержала целевую нагрузку, то проверка проходит под нагрузкой 80% от результата максимальной производительности, полученной при проведении стресс-теста.

Виртуальные
пользователи



Продолжительность
нагрузки ~ 4-8 часов

Продолжительность
теста

3. ПРОВЕРКА СТАБИЛЬНОСТИ проводится с ожидаемым уровнем нагрузки при длительном многочасовом тестировании

Виртуальные пользователи

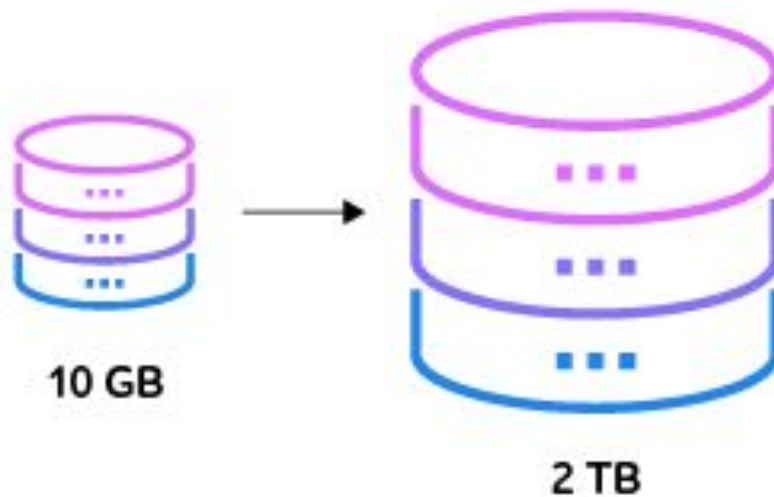


Продолжительность нагрузки ~ 7 дней

Продолжительность теста

4. ОБЪЕМНОЕ ТЕСТИРОВАНИЕ используется для оценки производительности работы ПО при увеличении данных, которые хранятся в БД приложения

Такая же форма нагрузки, как и для нагрузочного теста



5. ПРОВЕРКА МАСШТАБИРУЕМОСТИ позволяет оценить способность решения увеличивать производительность пропорционально добавлению аппаратных ресурсов системы.



6. ПРИ МОДУЛЬНОМ ТЕСТИРОВАНИИ нагрузка подаётся на отдельные компоненты ПО.

7. ТЕСТ «ЧАСА ПИК» проводится для того, чтобы проверить реакцию системы как в момент наибольшей загруженности, так и непосредственно после снижения нагрузки.

8. ТЕСТИРОВАНИЕ КОНФИГУРАЦИИ оценивает работу программного продукта при разных конфигурациях окружения.

ЭТАП 9 – АНАЛИЗ РЕЗУЛЬТАТОВ И ПОДГОТОВКА ОТЧЁТА

На основе предыдущих этапов создаётся документ, который описывает не только результаты проведенных тестов, но и ход выполнения каждого из них.