

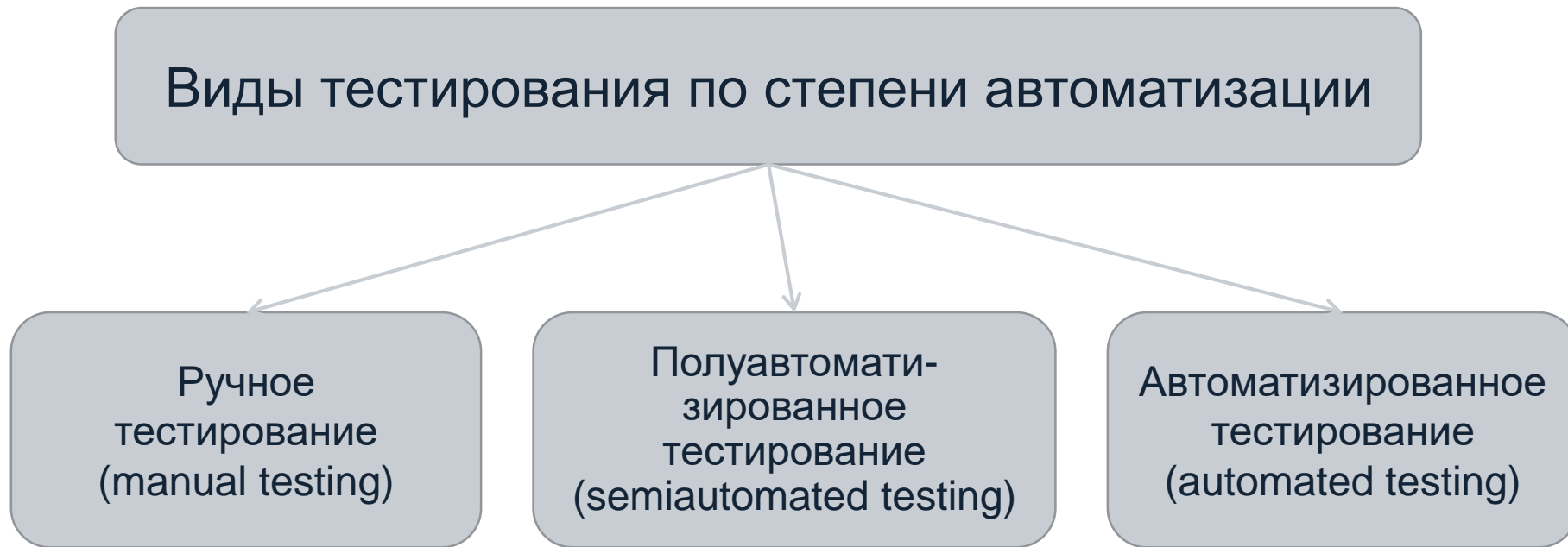
Инструментальные средства тестирования ПО

Sumy Educational Center
Software Quality Assurance (QA)

Вы узнаете о следующем

- Автоматизация тестирования
 - Виды тестирования по степени автоматизации
 - Понятие автоматизации тестирования
 - Инструмент для автоматизированного тестирования
 - Варианты использования автотестов
 - Преимущества и недостатки автоматизации
 - Уровни автоматизации тестирования
 - Техники и подходы в разработке GUI автотестов
 - Инструменты, поддерживающие жизненный цикл тестирования
 - Selenium
- Нагрузочное тестирование
 - Нагрузочное тестирование: понятие, виды
 - Цели нагрузочного тестирования
 - Инструменты для нагрузочного тестирования
 - Примеры инструментов, используемых в компании

Виды тестирования по степени автоматизации



Понятие автоматизации тестирования

- **Автоматизация** – это комплекс мероприятий, направленный на повышение производительности труда человека посредством замены части этого труда работой машин.
- **Автоматизированное тестирование программного обеспечения** - это процесс верификации ПО, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, выполняются автоматически при помощи инструментов для автоматизированного тестирования
- **Автотест (автоматизированный тест)** – это скрипт, имитирующий взаимодействия пользователя с приложением, цель которого – локализация ошибок в работе программного обеспечения.



Инструмент для автоматизированного тестирования

- **Инструмент для автоматизированного тестирования (Automation Test Tool)** - это программное обеспечение, посредством которого специалист по автоматизированному тестированию осуществляет создание, отладку, выполнение и анализ результатов прогона тест скриптов.

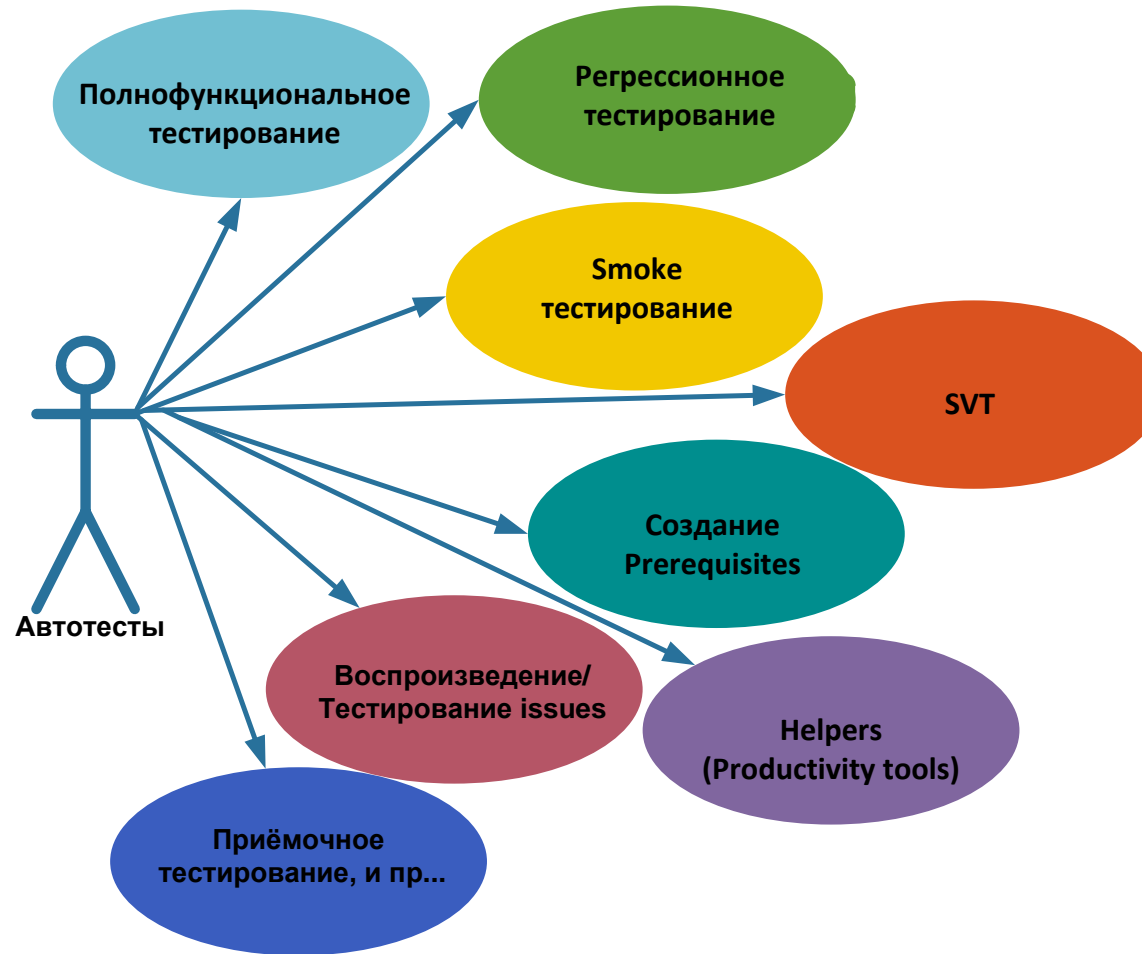


**Инструменты
функционального
тестирования**



**Инструменты
нагрузочного
тестирования**

Варианты использования автотестов



Плюсы и минусы автоматизации тестирования



Выполнение существующих регрессионных тестов

Более частое выполнение большого количества тестов

Выполнение тестов, которые невозможно или сложно выполнить вручную

Постоянство и повторяемость тестов

Освобождение времени на более важные и интересные задачи

Быстрый вывод продукта на рынок

Повышающаяся уверенность в качестве продукта



Полная идентичность проводимых запусков

Затраты на разработку и поддержку

Для разработки автотестов необходимы специальные навыки

Стоимость дополнительных библиотек и инструментов

Пропуск мелких ошибок

Уровни автоматизации тестирования

Unit-тестирование – покрытие кода тестами для проверки работы его методов

Тестирование API – тестирование прикладного программного интерфейса, предоставляемого приложением

Тестирование CLI – тестирование работы приложения через командную строку

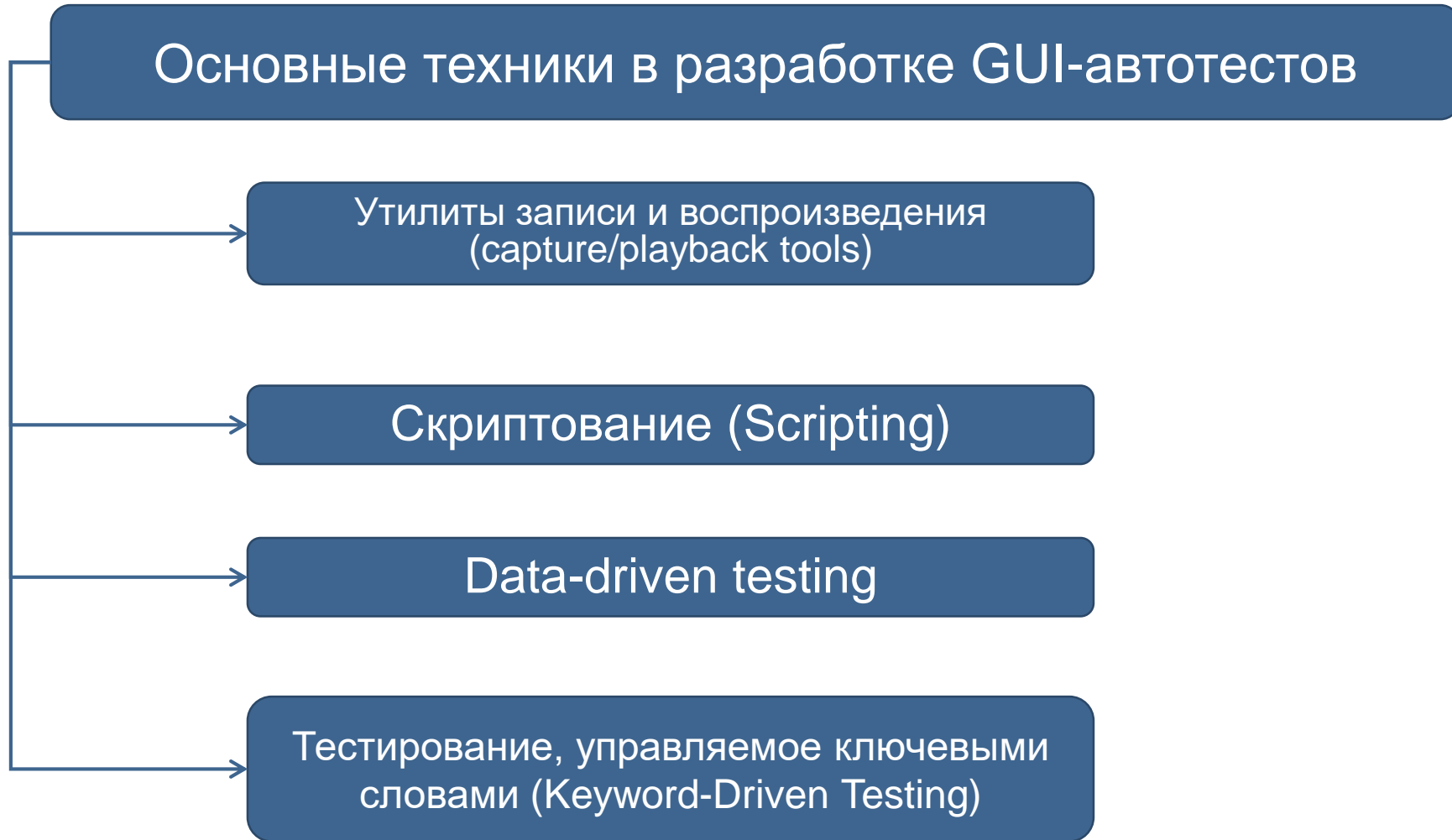
Тестирование GUI – тестирование графической оболочки приложения (web, desktop приложения)

Тестирование интеграций с внешними системами – работа приложения в связке с реальными системами

Что нужно автоматизировать?

Что нужно автоматизировать	Что НЕ нужно автоматизировать
Часто повторяющиеся тесты	Единичные тесты
Трудно выполнимые вручную операции	Часто меняющийся функционал
Длинные end2end сценарии, Workflows	Алгоритмически сложные проверки
Работа с большими объемами данных	

Техники и подходы в разработке GUI-автотестов

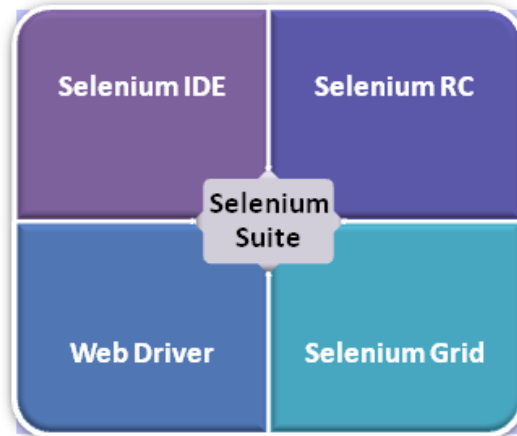


Инструменты, поддерживающие жизненный цикл тестирования

Фаза жизненного цикла	Тип инструмента
Фаза анализа бизнес-процесса	Инструменты моделирования бизнес-логики, отслеживания дефектов, генераторы документов
Фаза определения требований	Инструменты управления требованиями, верификаторы требований
Фаза анализа и проектирования	Инструменты проектирования баз данных и приложений, структурные диаграммы, блок-схемы
Фаза программирования	Средства проверки синтаксиса/отладчики, инструменты обнаружения утечек памяти и ошибок времени выполнения, тестирования исходного кода, средства статического и динамического анализа, инструменты модульного тестирования
Инструменты для проведения измерений	Средства анализа покрытия кода инструментами, измерения удобства применения
Фаза тестирования	Инструменты управлением тестирования, функционального тестирования GUI-приложений, нагрузочного тестирования и тестирования производительности
Другие инструменты	Генераторы тестовых данных, средства генерации заглушек, инструменты эмуляции

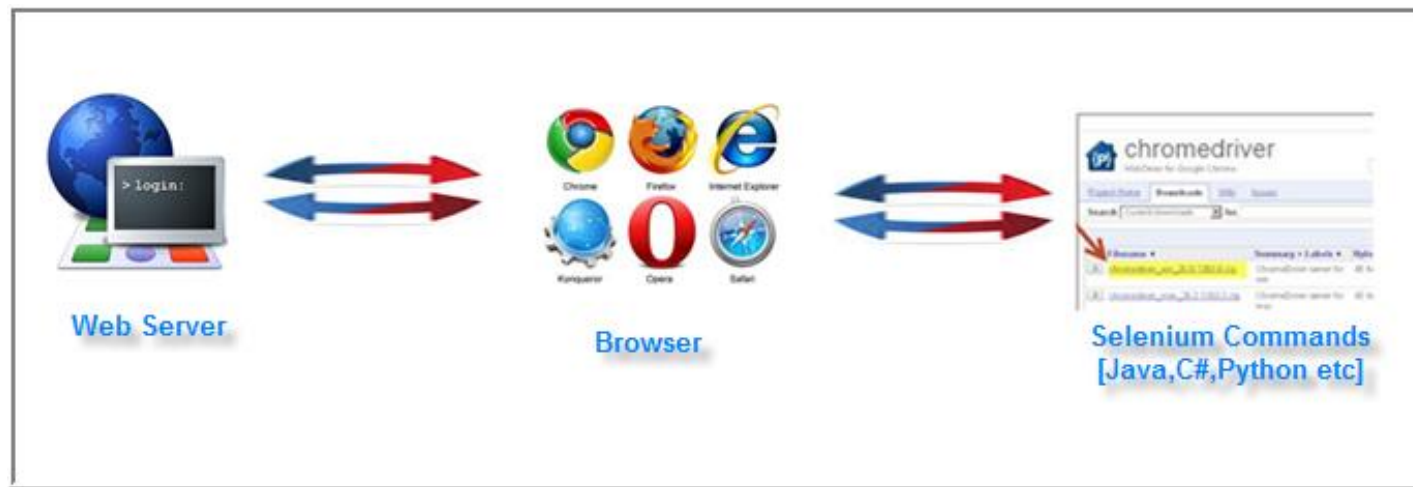
Семейство инструментов Selenium

- ✓ **Selenium IDE** – расширение для браузера Firefox, Записывает и воспроизводит действия пользователя.
- ✓ **Selenium RC** – работа с browser'ом посредством JavaScript-инъекций
- ✓ **Selenium WebDriver** – работа с browser'ом через оговоренный спецификацией интерфейс (<http://www.w3.org/TR/webdriver/>)
- ✓ **Selenium Grid** – функционал, позволяющий запускать тесты на разны окружениях в несколько потоков.

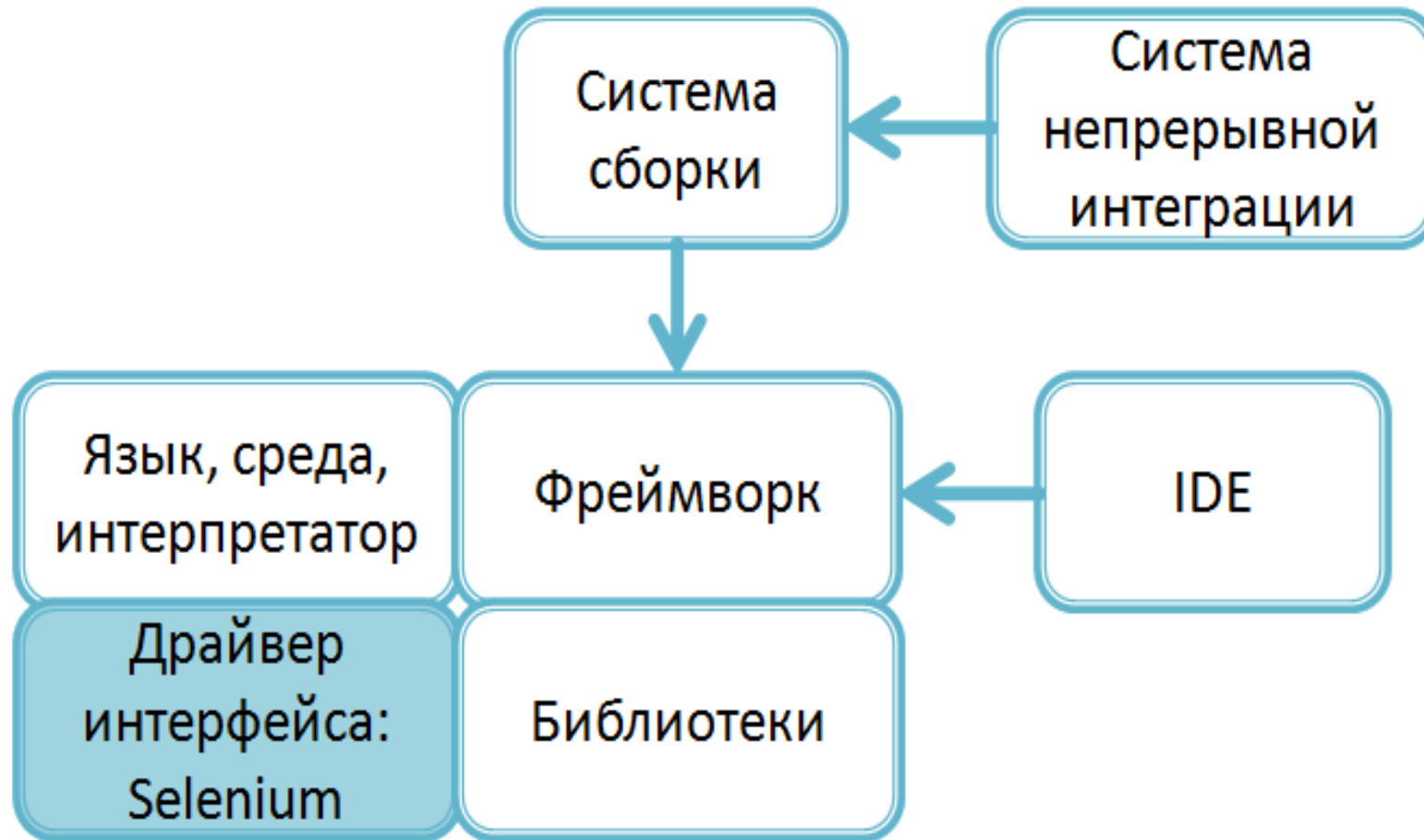


Принцип работы Selenium

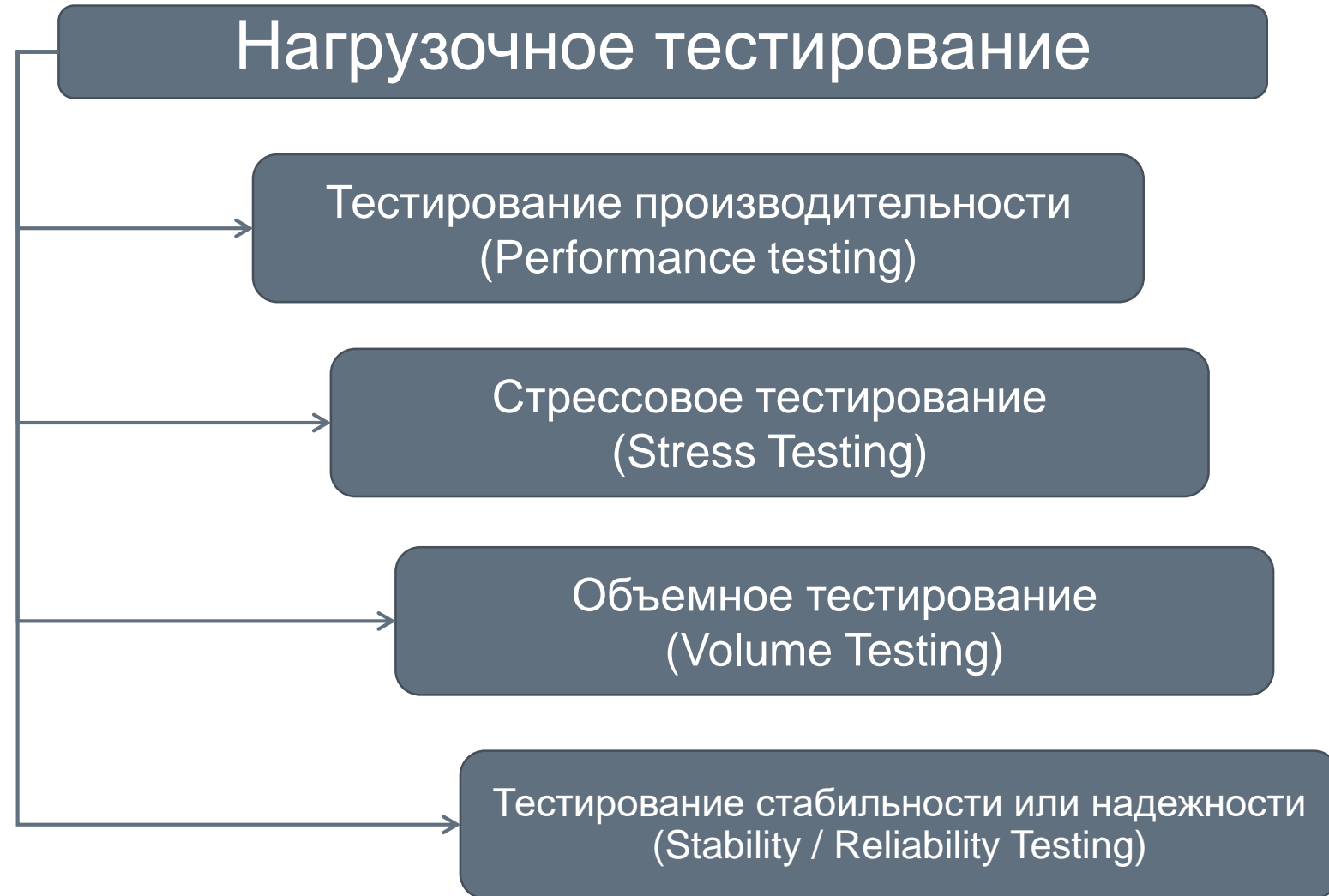
- ✓ **Код** взаимодействует с **драйвером браузера**
- ✓ **Драйвер браузера** взаимодействует с **браузером**
- ✓ **Браузер** взаимодействует с **web-сервером**



Типовая структура решения автоматизации



Виды нагрузочного тестирования

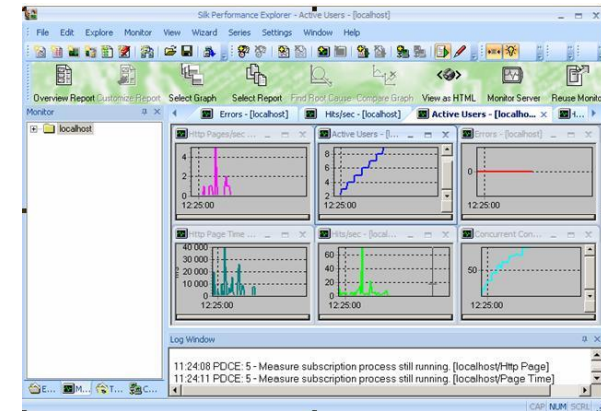
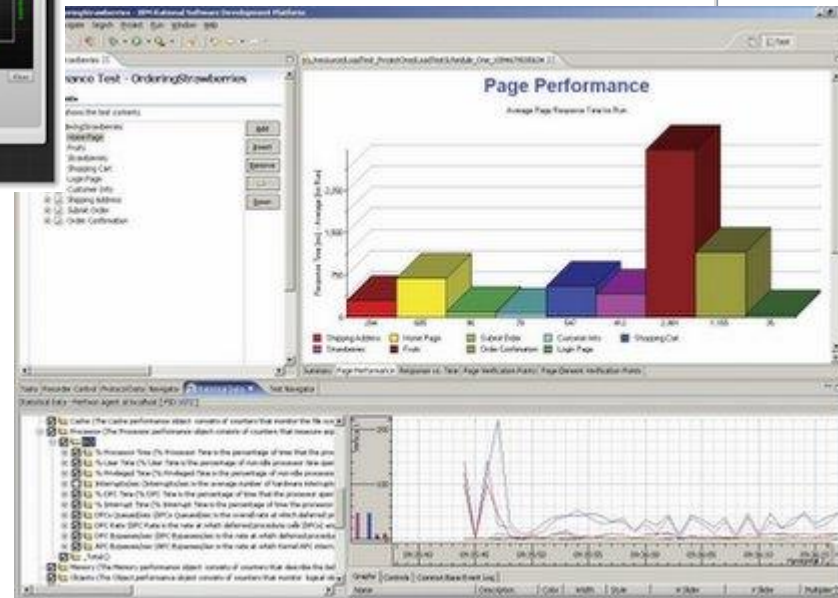
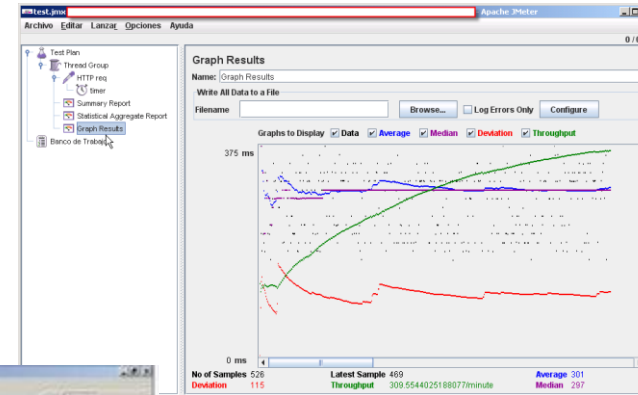


Цели нагрузочного тестирования

- ✓ Оценка производительности и работоспособности приложения на этапе разработки и передачи в эксплуатацию
- ✓ Оценка производительности и работоспособности приложения на этапе выпуска новых релизов, патч-сетов
- ✓ Оптимизация производительности приложения, включая настройки серверов и оптимизацию кода
- ✓ Подбор соответствующей для данного приложения аппаратной (программной) платформы и конфигурации сервера



Инструменты для нагрузочного тестирования



SoapUI

SoapUI - это инструмент для ручного и автоматизированного тестирования SOA-приложений.

Использование:

1. Helpers для ручного тестирования
2. Автотесты для smoke-тестирования (не всегда оптимально)
3. Стабы
4. Нагрузочные автотесты



Не злоупотребляйте использованием Groovy Script! SoapUI не предоставляет полноценной среды для отладки и разработки.

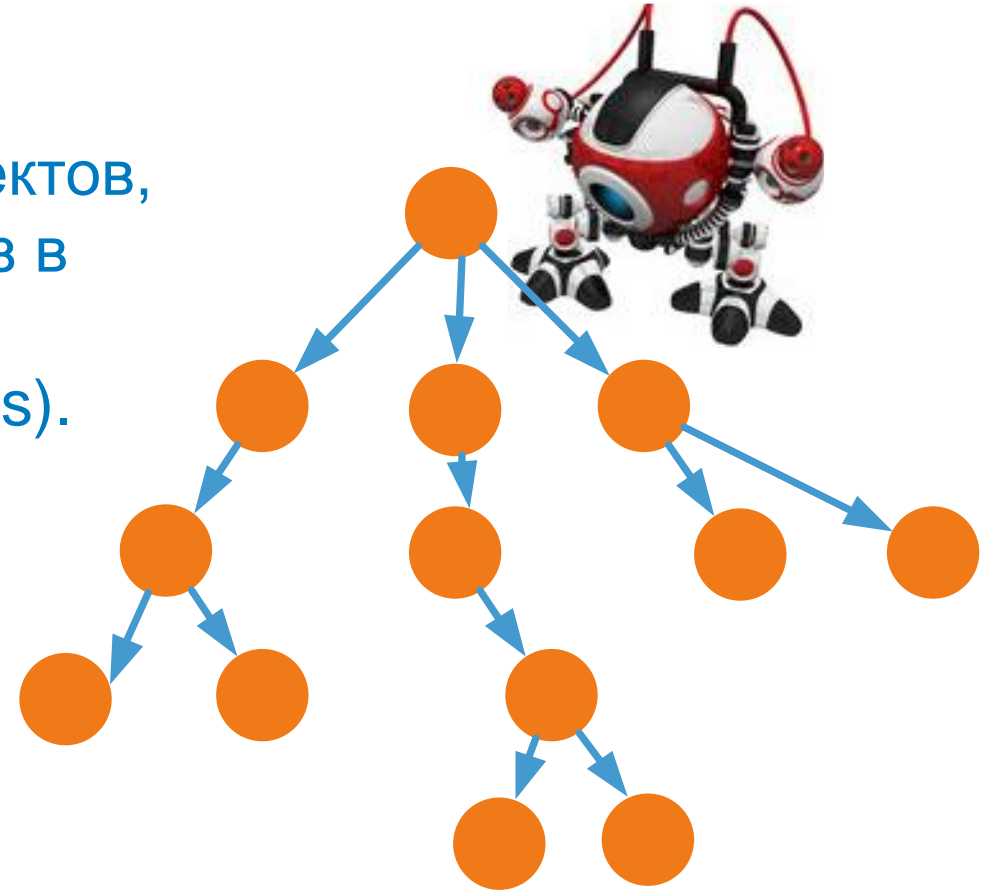
WebCrawler

Out-of-the-box инструмент для рекурсивного обхода иерархии объектов, начиная от корневого элемента вниз в целях обнаружения «битых» гиперэлементов (страниц Diagnostics).

Назначение: Smoke UI тестирование, замеры скорости открытия страниц

Рекомендации к применению:

- начальные стадии проекта, когда объектная модель не стабильна
- После обновления продуктовых компонентов, обновления версии Netcracker



Security Matrix Validator

Проверка Effective Grants объектов за короткий промежуток времени.

Users Group ▶ ▼ Object Types	All Users	Read-only users	Security Administrators	Production Control Errors Resolving	Mass Market Users	Service Fulfillment	Catalogs Management	FF Management	FF Errors Resolving
ROW -	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
All - 0	Pass	Fail	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Project - 7	Pass	Fail	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Circuit Path Element - 9	Pass	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Device Connector - 197	Pass	N/A	N/A	N/A	Fail	N/A	N/A	N/A	N/A
Site - 300	Pass	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Device - 301	Pass	N/A	N/A	N/A	Fail	N/A	N/A	N/A	N/A
Circuit Folder - 801	Pass	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
hub - 888	Pass	N/A	N/A	N/A	Fail	N/A	N/A	N/A	N/A
Ethernet Hub - 2664	Pass	N/A	N/A	N/A	Fail	N/A	N/A	N/A	N/A
Token ring Hub - 3552	Pass	N/A	N/A	N/A	Fail	N/A	N/A	N/A	N/A
FDDI Hub - 4440	Pass	N/A	N/A	N/A	Fail	N/A	N/A	N/A	N/A
Rack - 10025	Pass	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
WAN Switches - 13320	Pass	N/A	N/A	N/A	Fail	N/A	N/A	N/A	N/A
Router - 14208	Pass	N/A	N/A	N/A	Fail	N/A	N/A	N/A	N/A
NLAN Path -	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
100-BaseT PE -	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Report -	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Phone Number -	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Control Container -	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Inventory -	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Subsystem -	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Keyword Driven Testing Tool: написание тест кейсов.

1	2	3	5	6	7
Name	Run	Description	Action	Value	Type
Keyword		Sheet - Validation Subflows	TC1. Changes creations		
TC1. Changes creations	yes	1. Making changes to the server configuration	TC1.1. Making changes to the server configuration		
TC1. Changes creations	yes	2. Operations with status CR	TC1.2. Operations with status CR		
TC1. Changes creations	yes	3. Check work listeners	TC1.3 Check work listeners		
Keyword		Sheet - Validation Subflows	TC2. Export creations		
TC2. Export creations	yes	1. Export CRs to the server configuration	TC2.1. Validate export on source server		
TC2. Export creations	yes	2. Export CRs to the server configuration with any statuses CR	TC2.2. Export CRs to the server configuration with any statuses CR		
Keyword		Sheet - Validation Subflows	TC4. OOB Validation And Restrictions rules		
TC4. OOB Validation And Restrictions rules	yes	1.1.1 Navigation to Documents>Solution Configuration Management Project>tab Domain>OOB SCM Domain and login to server	NavigateToServer	\${current.source.server.alias}	
TC4. OOB Validation And Restrictions rules	yes	1.2 Click button [Import] and upload zip1	Onlineimport	Rule	File
TC4. OOB Validation And Restrictions rules	yes	1.3 Check report	TrackerForRule	Modify	Activity
TC4. OOB Validation And Restrictions rules	yes	2.1 Click button [Import] and upload zip2	Onlineimport	Rule	File
TC4. OOB Validation And Restrictions rules	yes	2.2 Check report	TrackerForRule	Modify	Activity
TC4. OOB Validation And Restrictions rules	yes	3.1 Click button [Import] and upload zip3	Onlineimport	Rule	File
TC4. OOB Validation And Restrictions rules	yes	3.2 Check report	TrackerForRule	Modify	Activity
Keyword		Sheet - Validation Subflows	TC3. Import creations		
TC3. Import creations	yes	1. Imports CRs			
TC3. Import creations	yes	2. Import CR with delete-actions			



Пример файла конфигурации

Пример отчета

The screenshot displays the Netcracker test execution interface. On the left, a tree view shows the test structure for 'SCM Test part1', including steps like '1. Making changes to the server configuration', '2. Operations with status CR', and '3. Check work listeners'. A red error icon is visible next to step 1. On the right, a detailed view of a failed step is shown with the following information:

- Title:** Disable CR
- Description:** Button [Disable] doesn't exist cuz CR was disabled or closed
- Status:** (Error icon)
- Time:** 17:08:27.072

Below this, a screenshot of the application interface shows a 'CR1' (Change Request) that is 'Disabled'. The application title is 'http://qaapp088.netcracker.com:6890/common/uobject.jsp?tab= CR+Details'.

Q&A

Thank You

