

# **Основные принципы построения автоматического теста при помощи QTP**

# Введение

Целью данного документа является изложение подхода к автоматизации тестирования с использованием QTP, а также некоторых хороших практик построения тестов. Данный подход не претендует на какую-то уникальность или на какую-то особенную эффективность и удобство реализации равно как и на истину в последней инстанции. Это один из способов организации тестов для приложений сложнее калькулятора или блокнота. Преимуществами данного подхода являются гибкость и относительная универсальность (независимость от типа тестируемого приложения).

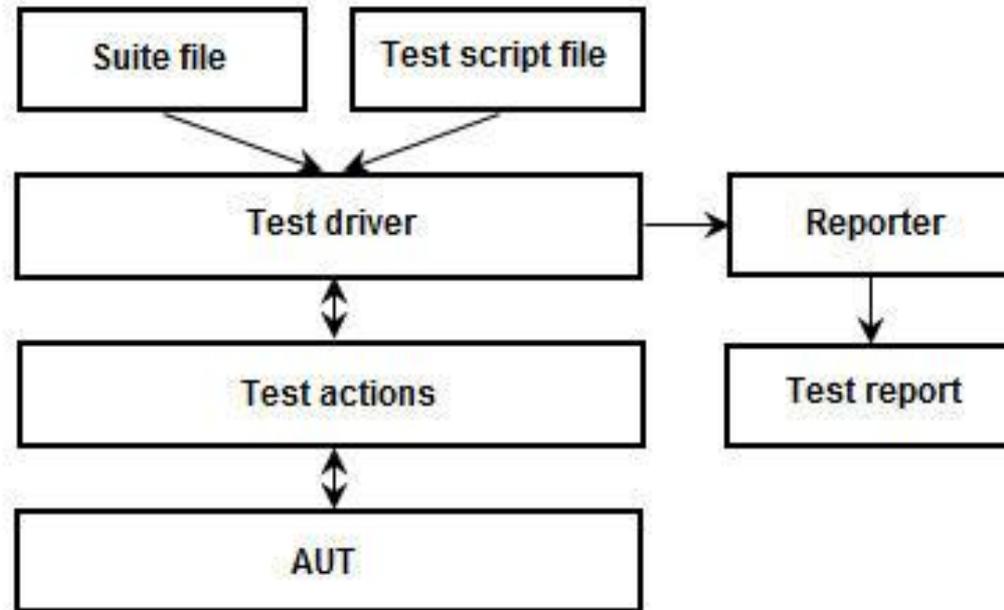
# Структура теста

Структурно тест состоит из следующих компонентов:

- Suite file – excel файлы, содержащий ссылки на файлы с тестовыми сценариями
- Test script file – excel файлы, содержащие описание тестовых сценариев
- Test driver – модуль, отвечающий за взаимодействие с внешними файлами, содержащими тестовые данные, вызов необходимых test actions и подготовку данных для отчета о тестировании
- Test actions – модули, реализующие взаимодействие с AUT.
- Reporter – модуль реализующий формирование отчета о тестировании
- Test report – отчет о тестировании

# Взаимосвязь компонентов

Взаимосвязь компонентов отражена на рисунке



# Компоненты теста:

## Описание

Данный файл предназначен для группировки тестовых сценариев по каким-либо признакам. Он определяет последовательность исполнения тестов и является входной точкой для автоматического скрипта.

Место расположения файла - TestData\`<file name>` на одном уровне с папкой автоматического теста.

Порядок следования колонок в файле не регламентируется

## Структура Suite файла:

Название колонки	Описание	Присутствие обязательно	Заполнение обязательно	Комментарий
No	Номер тестового сценария	Нет	Нет	
Script	Имя файла тестового сценария без расширения	Да	Да	
Execute	Выполнять сценарий?	Да	Нет	Y – сценарий выполняется Другие значения – сценарий не выполняется
Comment	Комментарий	Нет	Нет	

## Test script file

Данные файлы содержат последовательность вызова test actions и данные для них. Кроме перечисленных в таблице колонок в файле могут содержаться любые другие колонки, в зависимости от параметров вызываемых действий.

## Структура Test script файла:

Название	Описание	Присутствие	Заполнение	Комментарий
No	Номер действия	Нет	Нет	
Action	Имя test action	Да	Да	Значение данного поля должно соответствовать имени action в автоматическом скрипте. В противном случае действие выполнено не будет, а в отчет о тестировании будет добавлена информация о вызове неизвестного action.
Execute	Выполнять действие?	Да	Нет	Y – действие выполняется Другие значения – сценарий не выполняется
ReverseMode or Negative	Флаг, означающий что отрицательный результат выполнения необходимо трактовать как положительный	Нет	Нет	Если в поле установлено значение Y, то для соответствующего действия производится инверсия результата

## Test driver file

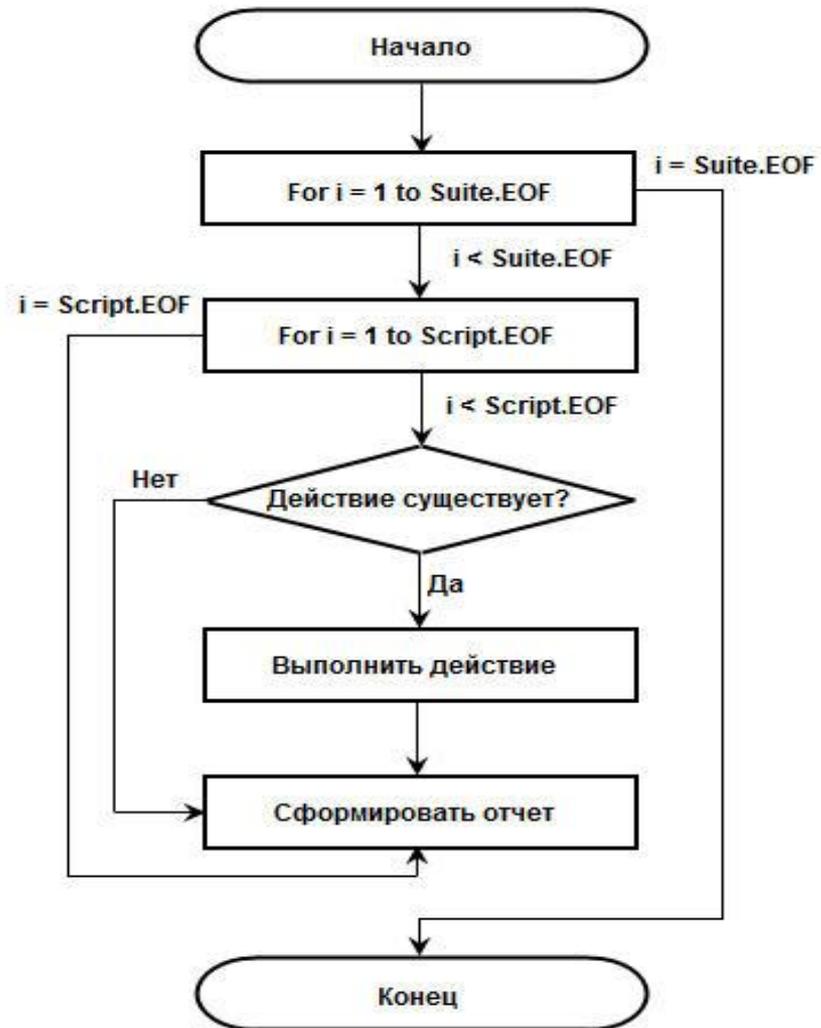
Данный модуль реализуется в виде Action. Он является единственным action в test flow qtp. Все остальные действия вызываются из данного action.

Входными данными для данного action является имя Suite file.

Модуль считывает данные из test script file, формирует объект Dictionary с параметрами и организует последовательный вызов действий, описанных в Test script file, и передает результаты вызовов в reporter для генерации отчета о тестировании.

Передача данных в actions организуется через параметр “data” объекта Environment.

# Алгоритм работы:



## Test actions

Данные модули реализуются в виде Action qtp и обеспечивают выполнение тестовых действий. Данные передаются в action через параметр data объекта Environment.

Отдельный test action реализует законченный шаг теста (открыть форму, создать элемент, проверить элемент). Каждый test action в начале работы должен проверить состояние UI и привести его к исходному для данного test action (например, открыть необходимую форму, если она не открыта). На выходе каждый test action должен сгенерировать объект Dictionary.

Каждый test action должен заканчиваться инструкцией ExitAction. Test Actions должны взаимодействовать с AUT посредством объектов в Object Repository(OR). Допускается обращение к UI с использованием descriptive programming, но предпочтение должно отдаваться использованию OR.

## Элементы объекта Dictionary

Наименование	Описание	Обязателен	Комментарий
State	Результат выполнения действия	Да	True – выполнено без ошибок False – выполнено с ошибками
Description	Описание	Да	Описание результата (например описание ошибки). В случае если State = false, заполнение данного элемента обязательно.
Act	Фактический результат	Нет	Строка, содержащая фактический результат
Exp	Ожидаемый результат	Нет	Строка, содержащая ожидаемый результат

# **Хорошие практики проектирования тестовых действий (test actions)**

# Атомарность

Тестовое действие должно проектироваться таким образом, чтобы по возможности выполнять атомарное действие (т.е. если есть возможность разделить код на два тестовых действия - лучше делить). При этом под действием понимается не клик мышкой по объекту, а законченное действие с т.з. бизнес-процесса (например: создание заказа, проверка данных заказа, отмена заказа).

# АВТОНОМНОСТЬ

Перед выполнением необходимых действий, тестовое действие должно проверять соответствие состояния AUT исходному состоянию и, при необходимости приводить состояние к исходному. Например, перед созданием заказа тестовое действие должно проверять загружена ли необходимая страница и, если нет, то пытаться ее открыть.

# Формирование результата

При выводе результата проверки в отчет необходимо ограничиваться не только констатацией (pass/fail) но и выводить ожидаемый и фактический результат.

# Описание объектов UI

При описании объектов UI в OR необходимо придерживаться следующих принципов:

- Если можно обойтись без использования Smart Identification, то лучше обходиться без него.
- Если можно сгруппировать объекты внутри иерархии (По умолчанию QTR записывает все объекты страницы плоским списком), то лучше группировать.
- Если есть элементы, повторяющиеся на всех (или большинстве) страницах, то лучше вынести такие объекты на generic страницу.

# Работа со сложными объектами UI

Для работы со сложными объектами UI (таблицы, меню, комбобоксы) желательно создавать отдельные классы (в отдельной functional library) и всю внутреннюю логику работы реализовывать именно там.

# Стабильность теста

При разработке авто теста следует помнить, что авто тест – это программа предназначенная для тестирования другой программы (AUT). Т.е. при разработке авто теста следует учитывать вероятность ошибок в AUT. Это означает, что при выполнении авто тест должен быть стабильнее AUT и «падать» он должен только в крайнем случае (желательно перед этим записав причину в отчет).

# Список полезной литературы

## VB Script:

- <http://www.w3schools.com/Vbscript/default.asp>
- [http://www.askit.ru/custom/progr\\_admin/progr\\_admin\\_plan.htm](http://www.askit.ru/custom/progr_admin/progr_admin_plan.htm) (с п.2 по п.4)
- VB Script Reference из хелпа по QTP

## QTP:

- Quick Test Professional Tutorial
- QuickTest User's Guide

## Для ознакомления с DOM структурой

- <http://www.w3.org/DOM/>
- [http://msdn.microsoft.com/en-us/library/ms764730\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms764730(VS.85).aspx)

# Ваши вопросы