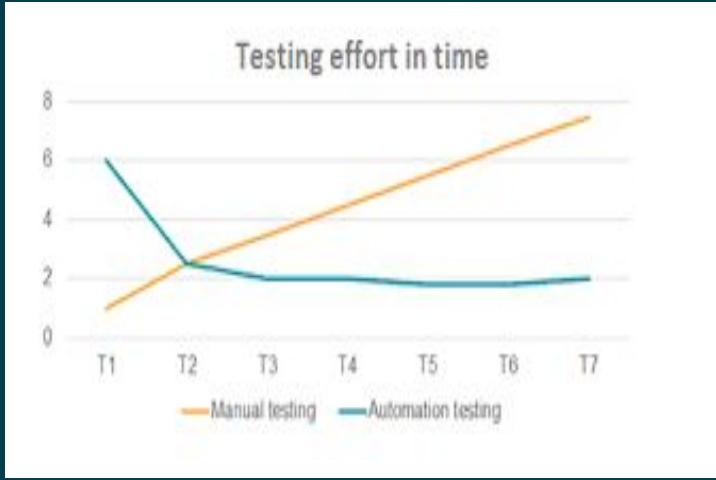
API Automation

- By Usha Sai

/thoughtworks

Why to automate....

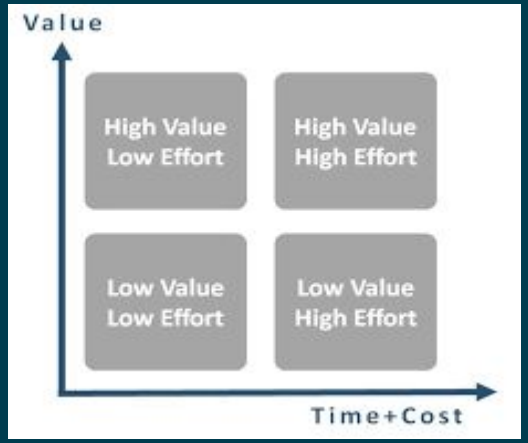
- Reduce human error
- Improves efficiency
- Reduce feedback cycle
- Get continuous feedbacks

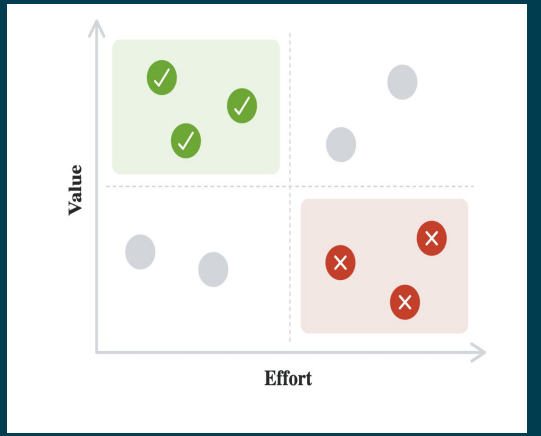


/thoughtworks

How to automate....

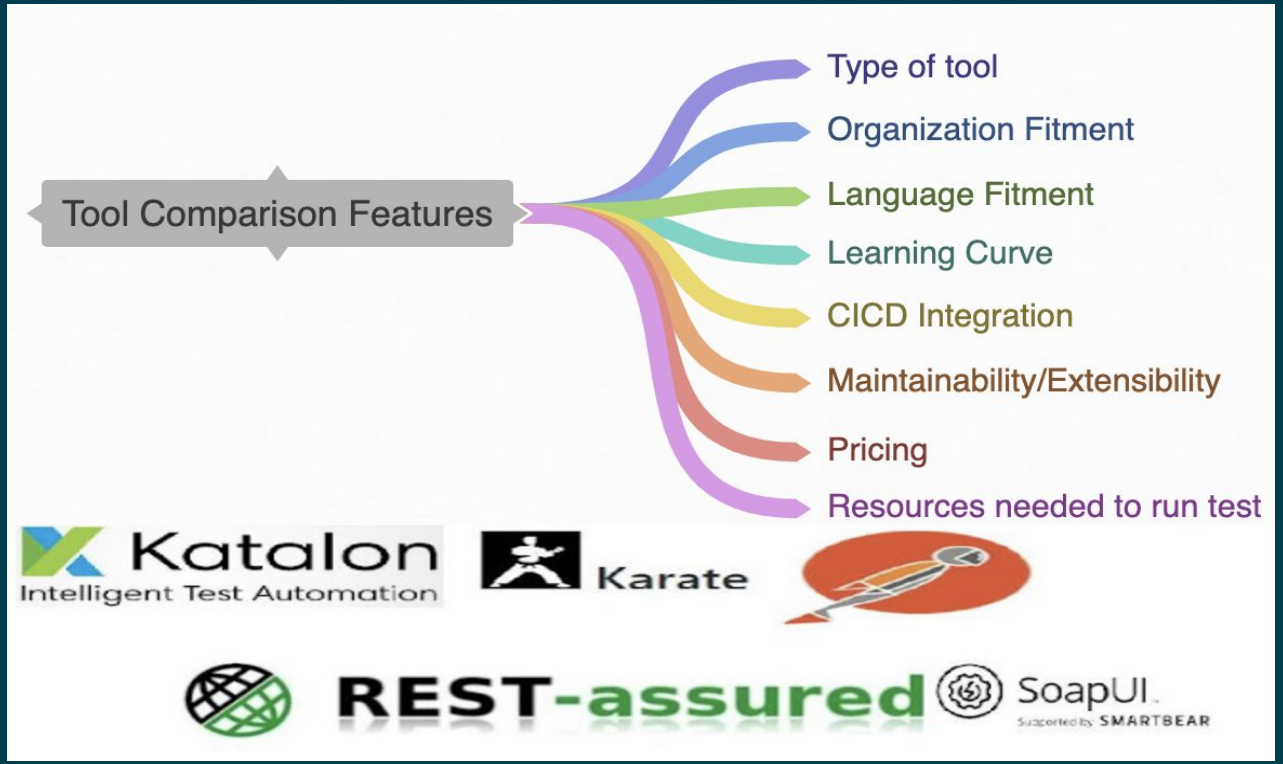
- Identify the tests to be automated
- Determine best tool for automation





/thoughtworks

Picking right tool...



/thoughtworks

Rest Assured....

- Java DSL for writing tests for restful API's
- Runs on top of JUnit or TestNG
- Makes tests powerful but easy to read and maintain
- Open source

/thoughtworks

REST Assured features...

- Supports all http methods (get, post, put...)
- Supports gherkin syntax (given, when, then)
- For assertions we can use hamcrest matchers
- For selecting elements from response we can use JSONPath and XML path

S.N.	Method and Description
1	**GET** The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.
2	**HEAD** Same as GET, but transfers the status line and header section only.
3	**POST** A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.
4	**PUT** Replaces all current representations of the target resource with the uploaded content.
5	**DELETE** Removes all current representations of the target resource given by a URI.

/thoughtworks

Configuring rest assured...

Maven -

```
<!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured</artifactId>
    <version>4.4.0</version>
    <scope>test</scope>
</dependency>
```

/thoughtworks

API to be automated...

- BART API - https://api.bart.gov/docs/overview/index.aspx
- Provides station data
- Clone - https://github.com/UshaSaiC/API-Automation

/thoughtworks

Chapter - 1 : Pre-req : Give them the repo to be cloned… Status code validation (write the code during season) [Exercise - etd end point verify 200]

Chapter - 2 : validating response body, content type and logging request and response body (write the response body validations code test here during session)
`root.routes.route[0].name` ---> JSON path
JSONPath is path expression for JSON object
Ily.. XMLPath - XML and XPath - HTML
Rest assured uses gpath notation of JSONPath query language

`equalTo` ---> hamcrest matchers
Expresses expectations in readable format [Exercise - same end point one more body assertion]

Chapter - 3 : Parameterizing tests
2 types of params -
Path Params ---> after api part, its /route/routes/name (3 path params are here route, routes and name)
Query Params ---> after api part, its /?page=3 (page is key name for param and 3 is its value)
In automation, data provider is used

Chapter - 4 : Optimizing code
Request specification is used to reuse request specification
Response specification is used to reuse response related checks
Extraction of response body and then doing assertions(extraction is used when once generated authentication token, must be used in consequent steps)

/thoughtworks

Chapter - 5 : XML // optional this chapter
Rest assure detects the xml responses by content type.. No additional configuration is required

Chapter - 6 : Deserialization and serialization
Get single and multiple users - deserialization
Post i.e create a user - serialization

Chapter - 7 : exception handling, timeout, reporting // optional this chapter
Command - `mvn surefire-report:report -Dtest=SlowResponseAPI`

/thoughtworks

Hamcrest Matchers

- `allOf` - matches if all matchers match (short circuits)
- `anyOf` - matches if any matchers match (short circuits)
- `not` - matches if the wrapped matcher doesn't match and vice
- `equalTo` - test object equality using the equals method
- `is` - decorator for equalTo to improve readability
- `hasToString` - test Object.toString
- `instanceOf`, `isCompatibleType` - test type
- `notNullValue`, `nullValue` - test for null
- `sameInstance` - test object identity
- `hasEntry`, `hasKey`, `hasValue` - test a map contains an entry, key or value
- `hasItem`, `hasItems` - test a collection contains elements
- `hasItemInArray` - test an array contains an element
- `hasProperty` - checks if a Java Bean has a certain property can also check the value of this property
- `closeTo` - test floating point values are close to a given value
- `greaterThan`, `greaterThanOrEqualTo`, `lessThan`, `lessThanOrEqualTo`
- `equalToIgnoringCase` - test string equality ignoring case
- `equalToIgnoringWhiteSpace` - test string equality ignoring differences in runs of whitespace
- `containsString`, `endsWith`, `startsWith` - test string matching

/thoughtworks

Source - http://hamcrest.org/JavaHamcrest/javadoc/2.1/

Plain Old Java Objects (POJOs)

Straightforward java classes with some properties and methods to modify those properties

/thoughtworks

Serialization and Deserialization

Deserialization and Serialization

Converting a java object to xml or json api response is called serialization

Converting api response which is in json or xml or any other format to instance of Java Object

/thoughtworks

/thoughtworks