Go School(Ngee Ann Polytechnic)

Project
API-T-DROID
(API Automated Testing Droid)
Enabling A 'Continuous Testing Model' Organization

Chan Jiunn Hwa
26th April 2021
Version 0.90(Alpha)

Table of Contents

1   Executive Summary

This project(API-DROIT, suggested pronounciation is Epic-Driod) is an adroit API testing service.

The continual growth of microservices has required such services to be tested in order to conform to expectations of functionality, reliability, performance, and security of the application. And in conjunction, the rise of DevOps and Agile methodology means that testing departments cannot remain as that 'other deparment' but needs to integrate and automate. Test automation benefits the quality process through speed, wider and deeper test coverage, consistency, significant cost savings and productivity gains, all resulting in faster time to market with a better delivery and product.

On the other hand, with increasing adoption Agile and DevOps methodologies, low level of automation of test activities would become the critical bottleneck in delivery.

There is therefore an opportunity to help enable automated testing, and in this project automated API testing.

1.1 Enabling Capability Maturation Towards A Continuous Testing Model Organization

Ironically, QA/Testing has remained perhaps the most manual, but testing cannot remain to be manual, or this chasm will have implications for competiveness. Like DevOps with its move towards CI/CD, it is envisioned that automated testing will help the organization to become CI/CD and CT(ie. Continual Testing) as well.

By CT(Continual Testing) it means the organization will no longer see Testing as a discrete, a before Release Activity, but as a continuous cycle, that is in sync with the entire product pipeline.

Hence, it means that CT organizations needs to integrate Testing right on project inception, throiugh the development stage, and into post-production as well. For instance, Regression Testing, Performance Testing, Security Testing, etc should be continued after Release, especially if it is an API service with a public front.

In addition, instead of dedicated testing teams, more modern methodologies may see sense in embedding testers into development teams, especially those running on fast iteration cycles.

Ultimately, it is only through integration and automation, that whole-of-organization cycles can iterate faster and better.

2 Background

Many research have shown that the API testing market is expected to triple over the next six years due to a number of factors.

Global API Testing Market to Grow to Over $1 Billion by 2022

September 21, 2017

The Global API Testing Market Size Is Estimated to Grow from USD 447.4 Million in 2017 to USD 1,099.1 Billion by 2022, according to the "API Testing Market by Component (API Testing Software/Tools and API Testing Services), Deployment Type (Cloud Based and On-Premises), Vertical, and Region - Global Forecast to 2022" report from Research and Markets.

The API testing services segment is the fastest growing market segment that helps in driving the API testing market during the forecast period. The growth in the API testing services segment is expected to be driven by the growing significance of partnerships between the software development teams and the Quality Assurance (QA) services or an offshore QA teams. The API testing tools/software segment is expected to hold the largest market share in the API testing software market during the forecast period.

The adoption of API testing solutions has primarily been gaining competitive advantage over the others, to enable the continuous delivery of software development.

## 2.1 Factors Driving Towards API Testing

Generally, with the trend of moving from monoliths to containers and microservices catalyzes API testing.

Also, new development methodologies such as Agile, together with new Technologies has moved Testing to what is called as 'shifting left and right'.

Shifting left allows code to be tested earlier in the life cycle. The practice not only helps catch bugs sooner but also prevents developers from spending time on code that will be found to be erroneous later in the lifecycle. More continuous and rapid testing will increase testing code coverage.

While not applicable to all software types, services are well-positioned to shift right, a practice of testing in production environments. By using live production traffic to test a new service prior to its broader release, teams can determine if the new release will cause problems such as slower response times or deviations away from CPU and memory consumption. Twitter's Diffy, A/B testing, blue-green deployments, canary releases, and feature flags (like LaunchDarkly) can facilitate testing in production by routing a small percentage of live traffic to new instances to test functionality.

Next, according to Google Trends, the interest in 'API Testing' has been growing steadily over the last couple of years.
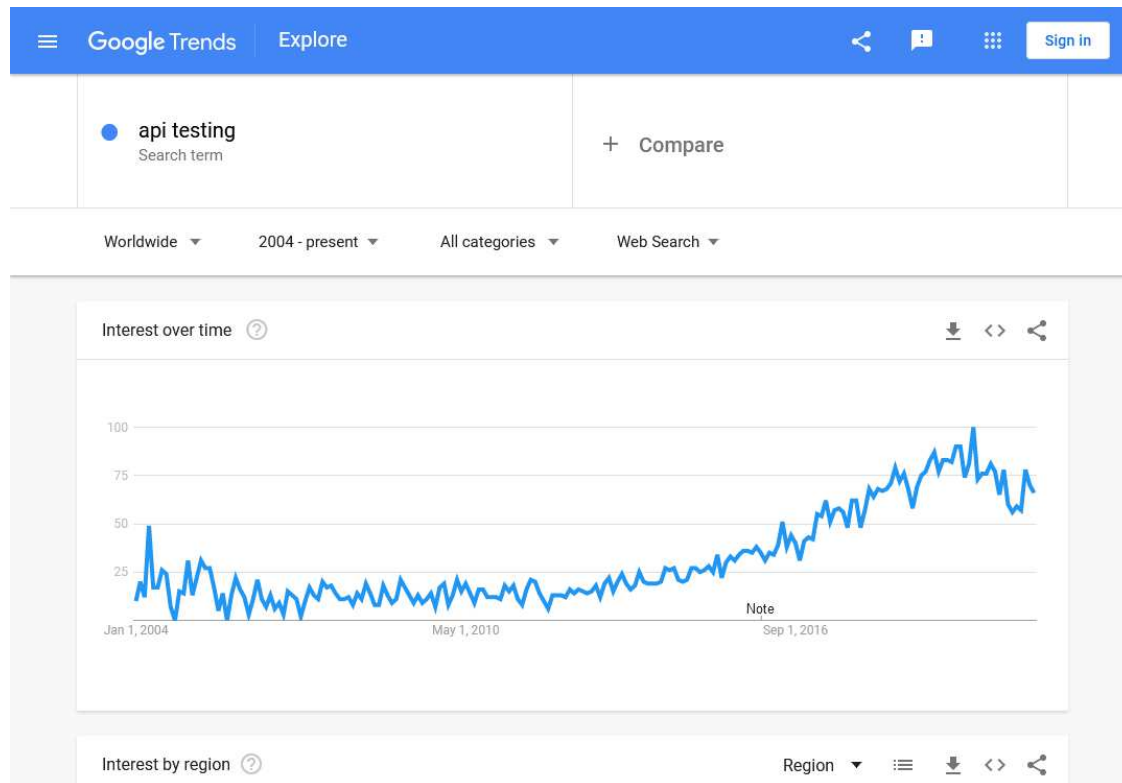
And according to a research by SmartBear over 3,372 software professionals in API testing during 2019, 91% of respondents either currently have, or plan to have a formal API testing process in place in the near future. About 45% of API testers reported that their organization automated more than 50% of test projects.

To sum up, there is defintely an increasing need for API Testing Services. In short, these are driven by:

● The rise of microservices.

● The trend towards test automation

● The rise of DevOps with the mantra to 'Automate Everything'.

2.2 The Testing Industry Environment

Various market research report expects growth in the software testing industry but it remains to be fragmented.

Fragmentation, is also further complicated by new testing startups coming online; in various Verticals and Silos, often with overlapping services. Overall, this results in complications for user adoptions.

2.3 The Testing Tools Landscape

Similarly, just as there are many players in the market, there is also a wide array of testing tools.

Sampling a quick look from the report Top 10 API Testing Tools in 2020 by Dzone, it shows names that are familar, while others remains unherard of.

Part of this fragmentation and proliferation is simply because of the evolving and dynamic landscape, as there is no one-size-fit-all tools. It remains challenging to find the ideal-one-tool that can do all.

Of note is that features of the commercial players such as Postman, Tricentis Tosca, though quite sufficient but the costs of ownership can be high. On the other hand, open-source solutions such as Rest-Assured, Karate DSL, are expensive in terms of requiring skilled manpower and effort.
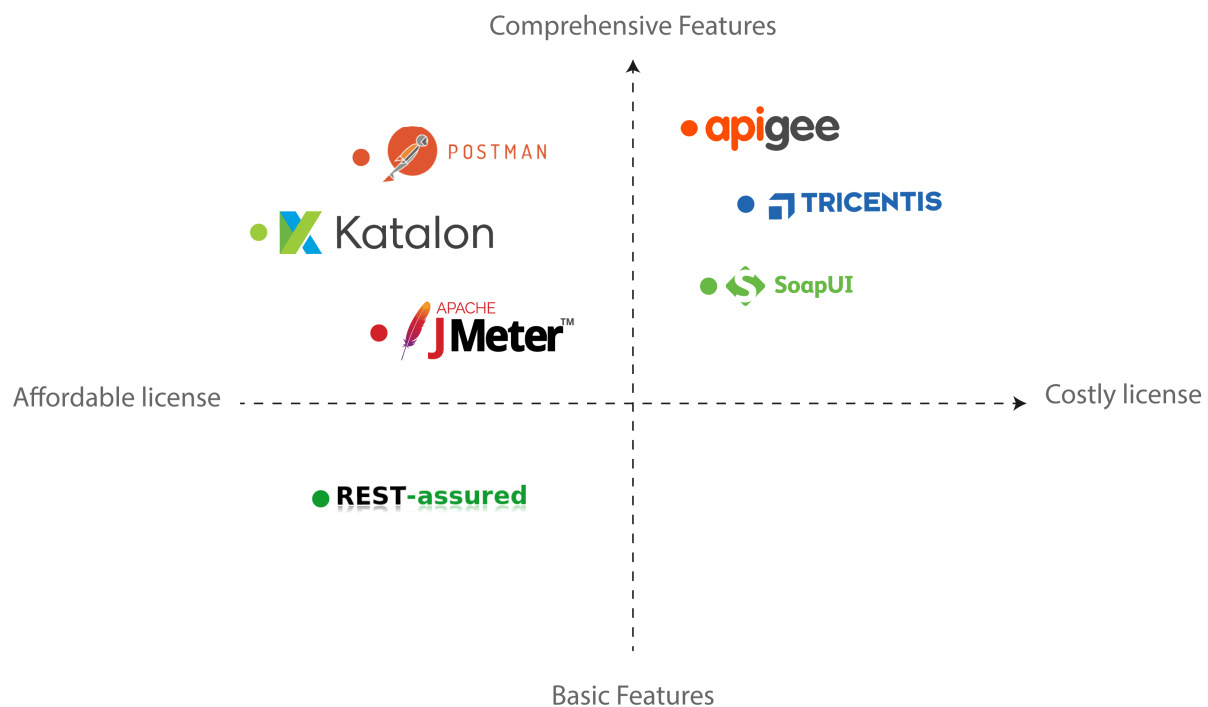
3 Project Proposal

Having framed the background, we next discussed the the source of idea of this project.

In the previous assignment, while building along the REST console API client to test some service API endpoint, it was observed that even with Postman/Curl, it is very laborious, slow, and error-prone to having repetively try out the endpoints.

In an effort to overcome this, I build 2 clients:

● C#/WPF client - a naive Postman clone which was helpful to my aims.

● REST console API client - as required by assignment.

Towards the end of the assignment, while iterating throught the console client, it was turned into a client that allows 'scripted' scenarios via configuration files.

While both proved to be very helpful in improving productivity and process, but the console client with its ability to run automated test thus inspired the desire for further exploration.

Looking back, as an example, a simple test that may have taken 5-10 mins to manually setup, test, record, and tear-down could be runned and completed in less than 10 seconds('whoa' was the 1st word I said then).

Tests could be runned as frequently as required through cron schedules, if needed.

With assignment deadline being a hard constraint, this productivity gain frees up time to focus on development.

Lastly, it allows richer and deeper testing, because different set of endpoint actions could be re-combined or repackaged to form different test scenarios.

.

3.1 Objective

Having expeirenced some of the issues and difficulties of Manual Testing, the objective of APIC-T-DROIT is to deliver a performant, and cost effective automated test client, if not an EPIC-TESTING-DROID.

3.2 Scope

Due to the time constraint, and understanding the fact that the developer does not come from a Test/QA background, the challenge and scope is to develop both a useful and usable scriptable client that is baseline complete.

3.3 Feature Discovery

To circumscribe an initial features set, we will walkthrough the below steps.

To beign, generally speaking these are common categories of testing:

- Validation Testing

- Functional Testing

- UI testing

- Load testing

- Runtime/ Error Detection

- Security testing

- Penetration testing

- Fuzz testing

- Interoperability and Compliance testing

Next, to have a feel of what features that are in demand, we take a look at features of some of the more popular product. For sure, it is not possible to include all their features in this timeline, nor desirable, but some commonly found features will need to be considered.

Each of tool has its own pros and cons; but they are widely used by developers and testers for their API testings.

CHARACTERISTICS	SoapUI	POSTMAN	Katalon
REST API Testing	Yes	Yes	Yes
SOAP API Testing	Yes	No	Yes
Automated assertion generation	Yes	Yes	Yes
Test script reusability	High	None	High
Data-driven support	Yes (*)	JSON, CVS	Excel, JSON, CVS, Datasource, and internal data
BDD Cucumber support	No	No	Yes
Environment handling	Yes (*)	Yes	Yes
Test reports	Yes (*)	Simple JSON, HTML formats	Reports in console, log, HTML and advanced analytic reports
Scripting languages	Groovy	JavaScript	Groovy Java
Web UI Testing	No	No	Yes
Mobile app testing	No	No	Yes
Test execution analytics	No	No	Yes

(*) Only supported in the commercial edition.

Thirdly,the product needs to support the Tester through the Testing Lifecycle. The below is a common depiction:
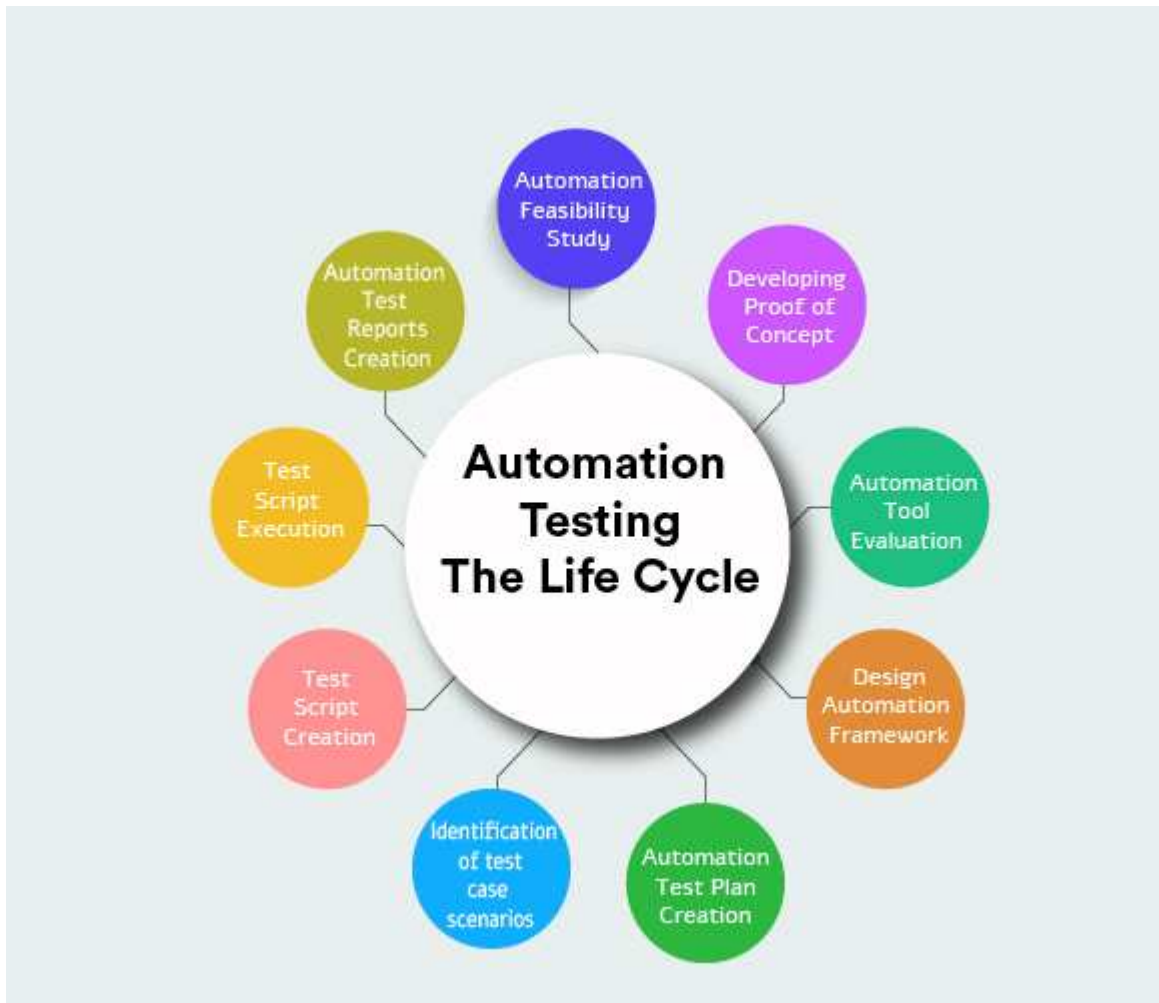
To sum up, though this may(or may not) be 'Mission Impossible', but if the project is undertaken, the specific logical and physical features will be further worked out in Functional and Design Analysis Phase.

At this point, the below could be some ideas, taking into account of the above and adding some product differentiation - with the exact features depending on further analysis, work progress, design flow, and available resources. Of consideration is also important to not just delineate the basic features, but to and allow extensibility in future phases.

In all consideration, some tentative features, detailed functions to be analysed.

- Functional Test
- Load Test
- Security Test
- Expired Links
- Syntax Expansion (WaitSleep, WaitResponse)
- Response Checking for matching(Response OK, Responsetime, etc as pass critieria)
- Reports and Dashboard
- Test Cases Creation and Management

4   Project Plan

A Project Plan helps ensure that work is expended with a disciplined approach and effort is planned and budgeted for each lifecycle activity.

4.1 Deliverables

The following artifacts are to be produced:

- Product
- Testing
- Demonstration
- Documentation

4.2 Product Demonstration

To demonstrate the usefulness of the product, a tentative idea is to demonstrate bugs discovery of some public API(subject to availability):

- Shoppee
- Other public API(to be researched)

4.3 Schedule

The proposed schedule:

TimeLine							26 APR - 14 MAY 2021													
Project APIC-T-DROIT				WK0							WK1								WK2	
		26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14
		M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F
		START					WEEKENDS							WEEKENDS						
CONCEPTUALISATION/RESEARCH/SELECTION(C)																				
Project Planning																				
Literature Review & Research																				
ANALYSIS & DESIGN(AD)																				
System Analysis & Design																				
CODING & DEVELOPMENT																				
Core Engine									CORE											
Report/Others													OTHERS							
TESTING																				
Planning & Design																				
Integration Test																				
REVIEW																				
Final review																				
Final changes and fix																				
DOCUMENTATION																				
Documentation																				
Presentation Preparation																				
SUBMISSION																				

5 REFERENCES

1. Introduction to API Testing https://docs.katalon.com/katalon-studio/docs/introduction_api_testing.html

2. Global API Testing Market to Grow to Over $1 Billion by 2022. https://www.devopsdigest.com/global-api-testing-market-to-grow-to-over-1-billion-by-2022

3. STATE OF TESTING REPORT 2020. https://www.practitest.com/assets/pdf/PT_repo4_20-20.pdf

4.

5. 17 Best Functional & API Automation Testing Tools In 2021 https://theqalead.com/tools/automation-testing-tools/

6. Top 10 API Testing Tools in 2020. https://medium.com/@alicealdaine/top-10-api-testing-tools-rest-soap-services-5395cb03cfa9

7. The 10 API Testing Tools You Can't Live Without in 2021 https://www.testim.io/blog/the-9-api-testing-tools-you-cant-live-without-in-2019-2/

8. What Are Automation Testing Tools? 9 Types & Examples   https://theqalead.com/topics/what-are-automation-testing-tools/

9. Top 3 API Testing Tool Comparison https://huddle.eurostarsoftwaretesting.com/soapui-vs-postman-katalon-studio-top-3-api-testing-tool-comparison/

10. Introducing Redpoint's Software Testing Landscape https://medium.com/memory-leak/introducing-redpoints-software-testing-landscape-3c5615f7eeae

11. Global Software Testing Services Market 2019-2023 | 12% CAGR Projection Over the Next Five Years. https://www.businesswire.com/news/home/20191115005456/en/Global-Software-Testing-Services-Market-2019-2023-12-CAGR-Projection-Over-the-Next-Five-Years-Technavio

12. Why IT Companies should switch to Automation Testing? https://www.appventurez.com/blog/why-should-companies-switch-to-automation-testing/