Browsers Lab Platform

The Blockchain Solution for the Software Automation Industry.

WHITEPAPER v1.2

Disclaimer

The information in this document is subject to change or update without any notice and should not be considered commitment by the Browsers Lab. This document therefore should not be reproduced or copied without the written permission of Browsers Lab. The contents therefore must not be imparted to a third party nor be used for an unauthorized purpose.

Nothing in this document shall be deemed to constitute a prospectus of any sort or a solicitation for investment, nor does it in any way pertain to an offering or a solicitation of offer to buy any securities in any jurisdiction. This document is not composed in accordance with, and is not subject to, laws or regulations of any jurisdiction, which are designed to protect investors.

Browsers Lab Token is not a digital currency, security, commodity, or any other kind of financial instrument and has not been registered under the Securities Act, the Security laws of any state of United States or the securities laws of any other country, including the securities laws of any jurisdiction in which a potential token holder is a resident

Abstract

Browsers Lab Token is regulated by smart contracts on an Ethereum based blockchain platform.

The platform supports the automation for software development by creating and innovating the current platform. The blockchain gives Browsers Lab Token the power to change the world of software development and software quality assurance.

Introduction

The intent of Browsers Lab is to create an innovative and simplistic way of revolutionizing your software development lifecycle (SDLC). Currently, one of the major bottlenecks of software development happen in the Testing part of the SDLC. As more application make their way toward a micro service infrastructure, the bottleneck for testing is bound to increase. At Browsers Lab, we want everyone to be prepared.

“Introducing Browsers Lab, a cloud based automation platform.”

What we provide is the simplicity of creating your test scripts in our cloud based platform. Using our browser grids, you can now execute your test scripts without the need of a computer. Browsers Lab is slowly blurring the lines between a technical and non-technical personnel. At Browsers Lab, we believe that making it easier for anyone to get an insight into automation and execute their organization automation script is key. Browsers Lab not only makes it easier for test execution, but also makes it simplistic for test creation. Functional test is a key aspect of testing, but as the market for automation grows, so does the need for automation. Help us get you prepared for the future by helping build your automation platform. We will provide UI Automation, API Automation and Mobile Application Automation.

Using blockchain technology, Browsers Lab opens their platform to individuals and companies in different markets. You will have the ability to utilize our platform, reward your employees, pay your employees or hire freelancers using our token. As time progresses our platform will help link individuals with a company that is right for them and companies with the right individuals.

Core Objectives

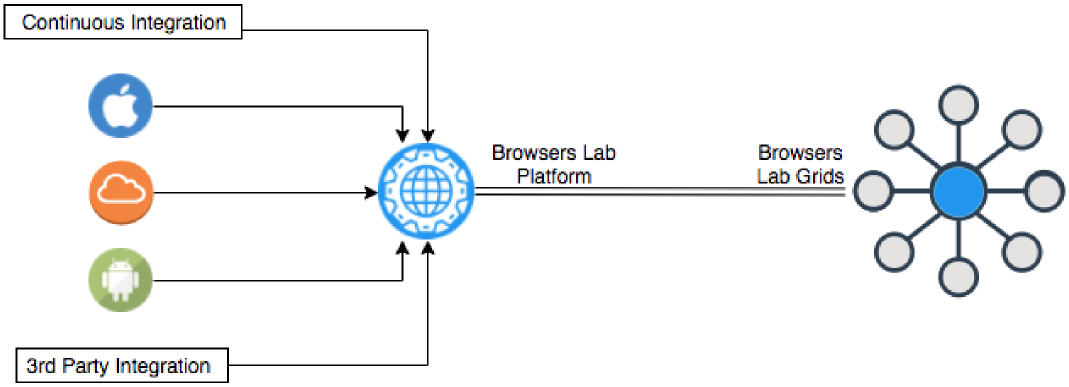
The core objective of Browsers Lab is to be the first to provide cloud based automation with the ability to manage and execute from our platform. Our platform opens up the blockchain for individuals and companies looking for a safe, simple and fast way to improve their software development life cycle.

Our goal is simply to provide a stable platform to build against, the ability to leveraging our platform to help with your development processes without having to spend countless hours with the setup. Our goal is also to provide access to individual and teams to execute test scripts without having any setup. Browsers Lab will also provide a way to execute your tests against individual environments with different parameters via our platform.

Architectural Overview

Browsers Lab frontend is built using Angular with a micro service infrastructure built with Node JS. Our structure will be Cloud-based infrastructure for stability and growth. All of our micro services are containerized therefore at any time our platform sees an influx of users, we can easily scale our platform.

As Browsers Lab grows, so will our support for desktop and mobile applications *(please see roadmap for more information).* Our goal is to have the ability to create your setup for any location without having to worry about if the application is supported on your device. Browsers Lab will provide a desktop application (Windows, Mac and Linux) to have the ability to code your tests directly from the OS of your choosing. Browsers Lab in addition to the desktop application will also provide applications for your Android and iOS devices.



Need for Browsers Lab

Testing a fundamental aspect of all software development. Not testing your application results in extra cost down the line, along with multiple developers tied down to fix the problem. Testing takes countless hours and locks a resource down but also does not guarantee that your application has been fully tested.

Releasing a new feature can break existing functionality and can cause an unforeseen problem. Verifying existing functionality is also time consuming and results in a resource being locked down. If an issue is found during the release of this feature, how can anyone verify that fixing the second issue might not create more.

Executing a full regression is expensive and is usually ignored due to time constraints. Verifying something by an individual also takes time and can be an issue if the resource is unable to repeat the same results. The testing team is more focused with the currently regression rather than focusing on newer features being released. Consequently, the testing documentation also lags behind due to unforeseen issues caught during regression.

On average, only about 52% of developers actually do functional testing on their development work. Developers rarely have time to code automation scripts and hiring automation engineers is expensive. Companies tend to lag behind due to funding issues or problems of time occur.

In an agile environment, companies cannot lock a resource down to do mundane tasks of testing while that resource could be utilized for new features. Companies have to hire many Quality Assurance Analysts and Automation Engineers to tackle these issues which also drives up cost.

Analysis

Software Testing utilizes approximately 40%-50% of total resources, 30% of total effort and 50%-60% of the total cost of software development. The cost benefit analysis of test automation and its impacts on the overall quality and schedule of the software is the central problem . This problem is of great significance in the scenarios of Continuous Integration (CI) where minor changes are incorporated frequently into the software. Optimization of regression testing cycles, in the scenarios of CI is vital because it has a great impact on revenue of the software industry which is about to produce a new release of an old software.

COnstructive COst **MOdel II (COCOMO II**)
MTEi: Manual Testing Effort
TMEi: Automated Test-case Maintenance Effort
ATEi: Automated Testing Effort
TAEi: Test Automation Effort
TTEi: Total Test-team Effort
STE: Combined total Software Testing Effort
KLOC: Kilo of line of code

Impacts of Test Automation on Software's Cost

Measured all the cost in terms of effort which has a unit of person/months

$$TTE_{i\ (without\ automation)} = MTE_i$$

$$STE_{without\ automatuon} = \sum_{i=1}^{k} TTE_{i\ (without\ automation)}$$

$$TAE_i = 2.4(KLOC)^{1.05}$$

$$TME_{i+1} = 0.3(TAE_i)$$

$$TTE_{i\ (with\ automation)} = MTE_i + (TME_i + ATE_i) + TAE_i$$

$$STE_{with\ automation} = \sum_{\mathrm{i=1}}^{\mathrm{k}} TTE_{i\ (with\ automation)}$$

$$Cost\ or\ Effort\ Impact = STE_{without\ automation} - STE_{with\ automation}$$

Impacts of Test Automation on Software Quality

Software quality have defined six main quality characteristics that are: Functionality, Reliability, Usability, Efficiency, Maintainability and Portability. Out of these features Functionality (degree to which program fulfills its requirements), Reliability (extent of failure free operations of software during a specified time slot) and Maintainability (time and effort required to analyze a failure in an operational software, to change it and then test the changed system) prominently affect the quality of the software.

$$Functionality\ F = \frac{number\ of\ SRS\ features\ fulfilled}{total\ number\ of\ features\ required}$$

$$Reliability\ R = \frac{mean\ time\ to\ failure}{mean\ time\ to\ failure + mean\ time\ to\ repair}$$

$$Maintainability\ M = \frac{time\ spend\ on\ testing}{total\ development\ time}$$

Impacts of Test Automation on Software’s Time

The impacts of test automation on the total time to market of the software can be calculated simply by adjusting the gained or lost effort

$$Time\ Impact = 2.4(Effort\ Impact)^{0.38}$$

Results

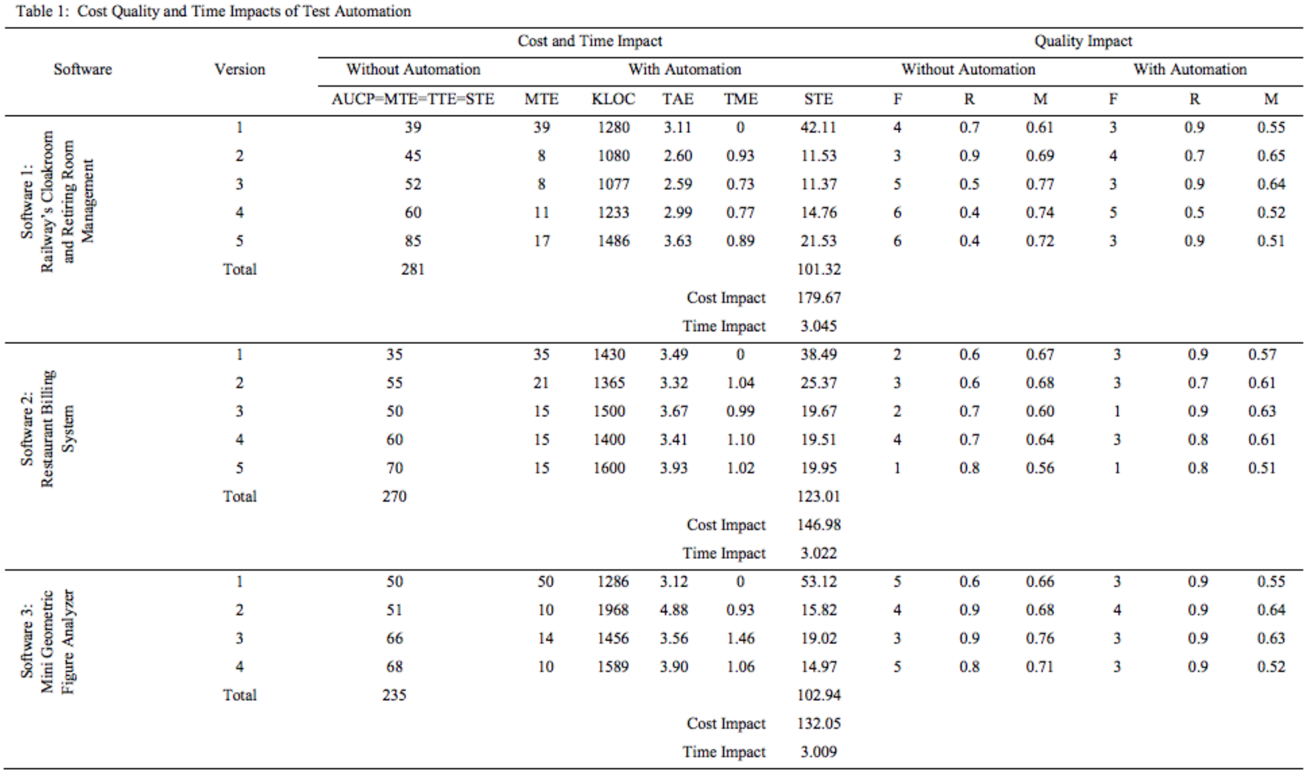


Table 1: Cost Quality and Time Impacts of Test Automation

Software	Version	Cost and Time Impact						Quality Impact					
		Without Automation	With Automation					Without Automation			With Automation		
		AUCP=MTE=TTE=STE	MTE	KLOC	TAE	TME	STE	F	R	M	F	R	M
Software 1: Railway’s Cloakroom and Retiring Room Management	1	39	39	1280	3.11	0	42.11	4	0.7	0.61	3	0.9	0.55
	2	45	8	1080	2.60	0.93	11.53	3	0.9	0.69	4	0.7	0.65
	3	52	8	1077	2.59	0.73	11.37	5	0.5	0.77	3	0.9	0.64
	4	60	11	1233	2.99	0.77	14.76	6	0.4	0.74	5	0.5	0.52
	5	85	17	1486	3.63	0.89	21.53	6	0.4	0.72	3	0.9	0.51
	Total	281					101.32						
						Cost Impact	179.67						
						Time Impact	3.045						
Software 2: Restaurant Billing System	1	35	35	1430	3.49	0	38.49	2	0.6	0.67	3	0.9	0.57
	2	55	21	1365	3.32	1.04	25.37	3	0.6	0.68	3	0.7	0.61
	3	50	15	1500	3.67	0.99	19.67	2	0.7	0.60	1	0.9	0.63
	4	60	15	1400	3.41	1.10	19.51	4	0.7	0.64	3	0.8	0.61
	5	70	15	1600	3.93	1.02	19.95	1	0.8	0.56	1	0.8	0.51
	Total	270					123.01						
						Cost Impact	146.98						
						Time Impact	3.022						
Software 3: Mini Geometric Figure Analyzer	1	50	50	1286	3.12	0	53.12	5	0.6	0.66	3	0.9	0.55
	2	51	10	1968	4.88	0.93	15.82	4	0.9	0.68	4	0.9	0.64
	3	66	14	1456	3.56	1.46	19.02	3	0.9	0.76	3	0.9	0.63
	4	68	10	1589	3.90	1.06	14.97	5	0.8	0.71	3	0.9	0.52
	Total	235					102.94						
						Cost Impact	132.05						
						Time Impact	3.009						

[1]

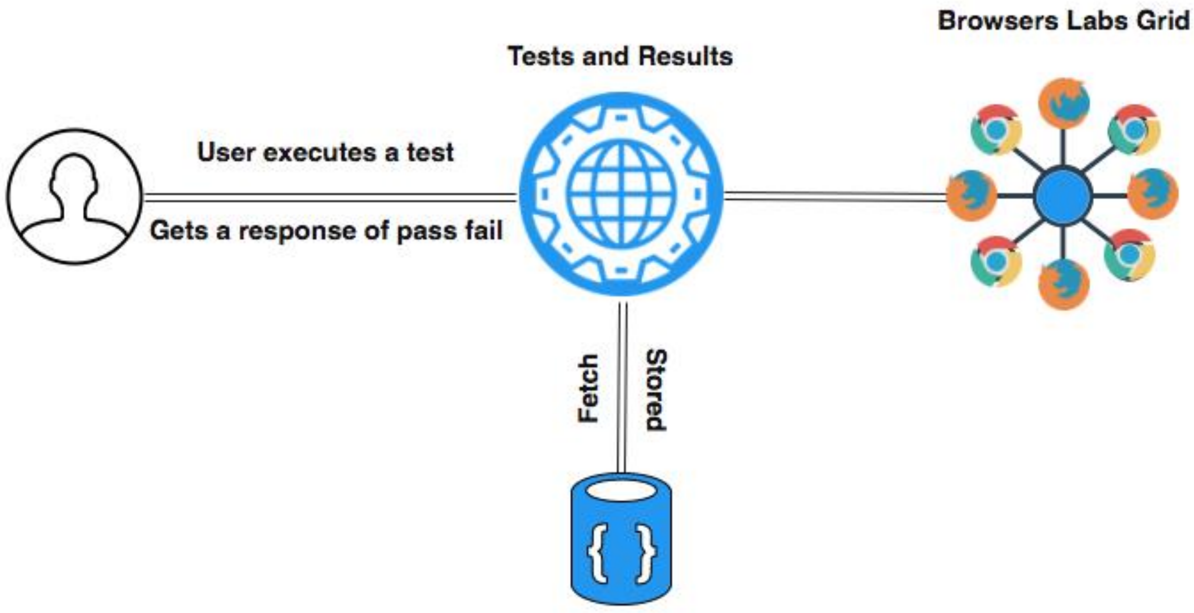
[1] https://www.sciencedirect.com/science/article/pii/S1877050916001277

Conclusion

It is a common observation in all the projects that there is a positive cost and time impact of test automation, and quality is also improved in most of the cases as program is found incorrect fewer numbers of times with automated test cases than with manual testing. The availability increases in all the cases and relative time in testing is also fairly decreased because of test automation. Software testing has a prime importance in software verification and validation. It is important because of two main reasons, first, it assures software quality, and second, nearly 60% of the total software's cost is spent over different types of testing.

How Browsers Lab solves the problem

Browsers Lab provides cloud based infrastructure for writing, storing and running your automated selenium tests. Developers, automation engineers and QA's can write and store their tests on our platform with the ability to execute their tests in Browsers Lab Cloud infrastructure. We give the ability to execute the stored tests from any location and any device.

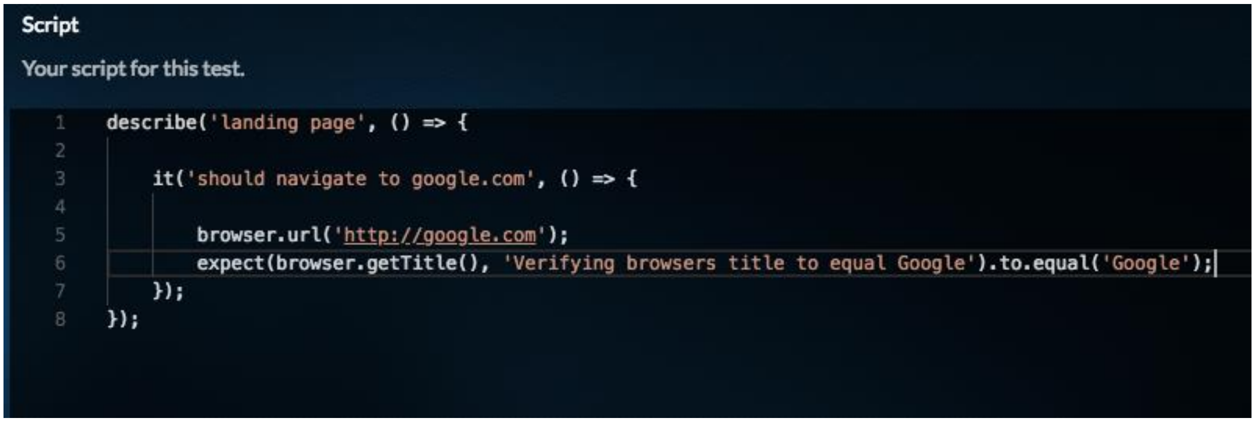


You can execute your tests from any location, for example, before leaving your home, you could execute your test cases and have the result emailed to you before you even get to your office. You could see the health of your environment without actually going through the step process. Browsers Lab makes it possible for you to test on multiple different devices and browsers. You have the ability to verify multiple environments with the same parameters to ensure stability of your company's infrastructure.

Regression checklists can now be automated and executed at any given time without having to worry about locking your resources. Browsers Lab can give you detailed reports of your platform’s stability and intractability. Tests will more reliable and faster when running standardized tests which cannot be skipped or ignored.

Using our platform, you will have the ability to attach your continuous integration and have your tests run as your environment is deploying. Browsers Lab helps you find bugs before they can even be caught my a manual tester.

Writing your test cases in Browsers Lab could not be any easier. It is based on mocha framework that relies on simply describing what you are testing, and what it should do.



Script

Your script for this test.

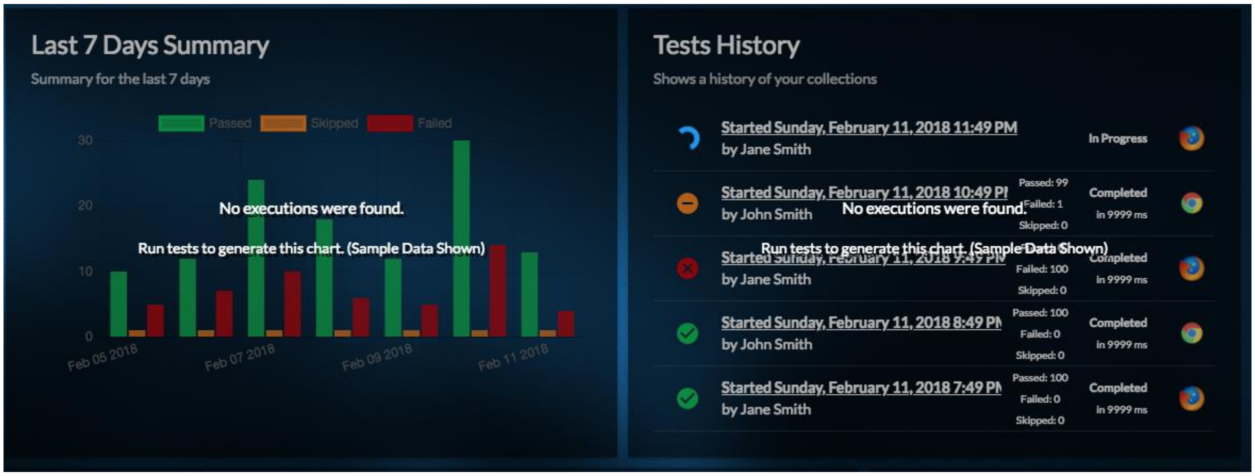
```
describe('landing page', () => {

    it('should navigate to google.com', () => {

        browser.url('http://google.com');
        expect(browser.getTitle(), 'Verifying browsers title to equal Google').to.equal('Google');
    });
});
```

As you can see, this gives the person the ability to type out their scripts directly on our platform and save them in the most simplistic fashion. In the simple script above, it is navigating to Google.com and verifying if the page title is Google.

Browsers Lab gives you detailed reports about your test executions (Image below). Browsers Lab is built to perform all those mundane tasks that no one wishes to do but they must be done. Our platform not only gives you detailed results but allows you to manage your organizations in an individual fashion.



Last 7 Days Summary
Summary for the last 7 days
Passed
Skipped
Failed
30
20
10
0
No executions were found.
Run tests to generate this chart. (Sample Data Shown)
Feb 05 2018
Feb 07 2018
Feb 09 2018
Feb 11 2018

Tests History

Shows a history of your collections

Started Sunday, February 11, 2018 11:49 PM
by Jane Smith
In Progress

Started Sunday, February 11, 2018 10:49 PM
by John Smith
Passed: 99
Failed: 1
Skipped: 0
Completed
in 9999 ms

No executions were found.

Run tests to generate this chart. (Sample Data Shown)

Started Sunday, February 11, 2018 9:49 PM
by Jane Smith
Failed: 100
Skipped: 0
Completed
in 9999 ms

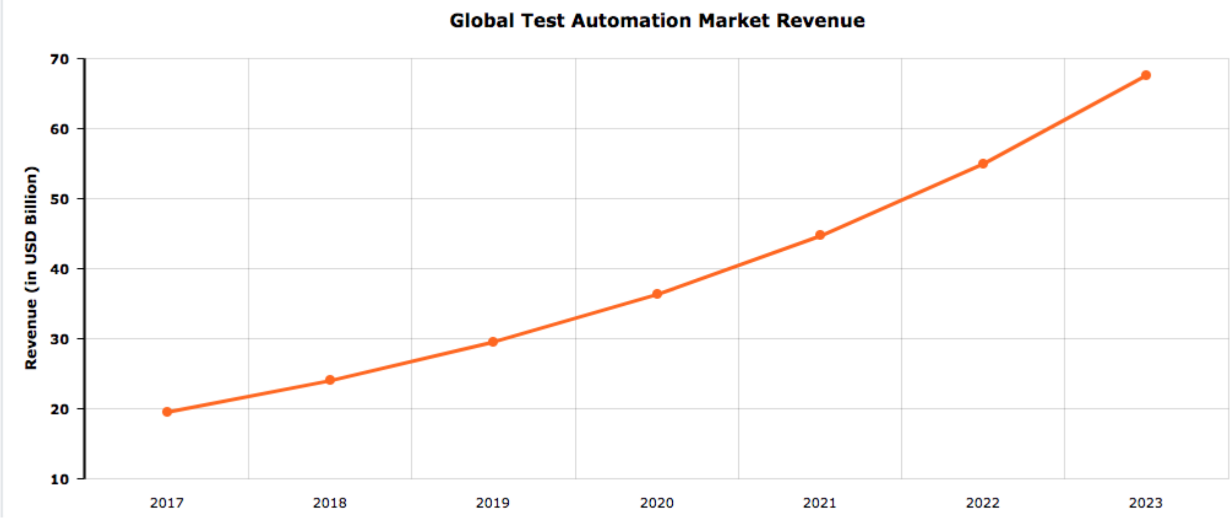
Started Sunday, February 11, 2018 8:49 PM
by John Smith
Passed: 100
Failed: 0
Skipped: 0
Completed
in 9999 ms

Started Sunday, February 11, 2018 7:49 PM
by Jane Smith
Passed: 100
Failed: 0
Skipped: 0
Completed
in 9999 ms

Currently the software automation industry is approximately $18 billion dollars growing at a Compound Annual Growth Rate (CAGR) 23.01%. At the end of 2022, the software automation industry will be approximately $54.98 billion dollars[2]. Another report by Transparency Market Research estimates that the software automation industry is likely to be at

[2] https://www.zionmarketresearch.com/news/test-automation-market

$109.69 billion dollars by the end of 2025[3]. As we can see, software automation is on the rise and has been for a long time. In order to keep up with the current market and current technologies, companies must adopt software automation as a forefront of their commitment to a smoother and safe environment. As we go toward agile methodology, we must adopt automation as a mainstream factor.



[3] https://www.transparencymarketresearch.com/test-automation-market.html

BL Team

Browsers Lab team has numerous years of experience. Each of our team members bring a different perspective to the team. Most of us have experienced the testing bottleneck and have banded together to resolve this issue for everyone.

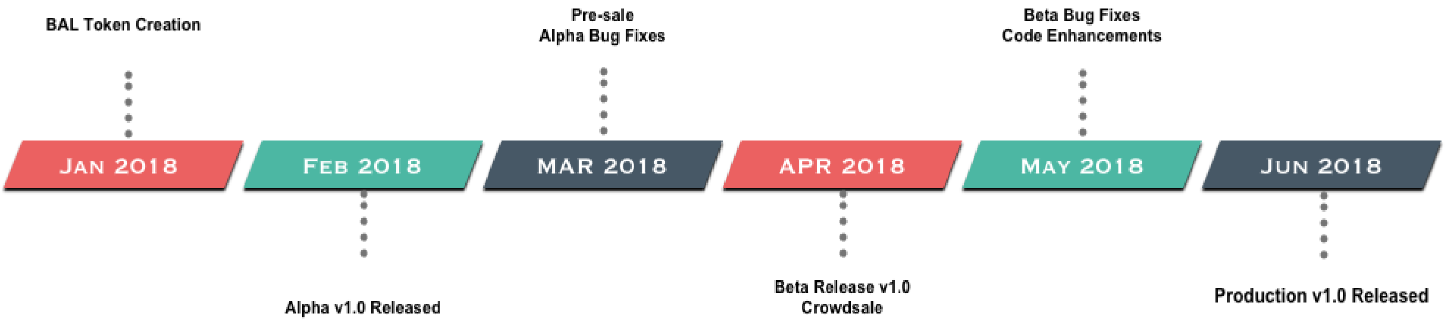
Our team consists of engineers with multiple years of experience and different background skills in many programming languages (Node JS, JavaScript, HTML5, Java, Go, C, C++, C#, etc.). Our data management teams has years of experience with Big Data dealing with multiple technologies (MongoDB, Cassandra, MSSQL, MYSQL, etc.). Some have worked for big named firms that have joined our team to help build a stable and sustainable product. Dealing with issues of software development in years of vast experiences, our team has committed their time to building Browsers Lab.

Browsers Lab team is helping set new standards for Software Development Life Cycle. Regardless of the outcome of this ICO, they are committed to bringing to market a product that will revolutionize the current Software Development Life Cycle. Help Browsers Lab team bring this change to the current structure by funding this ICO.

Road Map

Browsers Lab is current in full development. The alpha version has been deployed for everyone to experience. As time progresses, our platform will be developed to an enterprise standard with high level security.

The release plan is as follows:



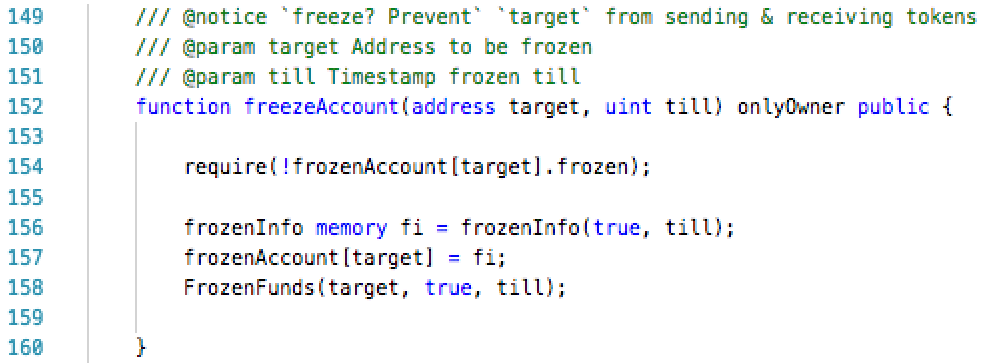
In January 2018, We created the Ethereum Smart Contract for our token (BAL). This is followed by Alpha Version of our platform being released in February 2018. The alpha platform will essentially give you the look and feel of our platform. Everyone will have the ability to view our platform and the development of this platform. March 2018 will be spent fixing and enhancing our platform in alpha stage. This will also be the time when Browsers Lab launches a Pre-sale for BAL Tokens. In April 2018, the Beta Version of the platform will be released. Bugs fixes and code enhancements will follow afterward in May 2018. Production Platform released in June 2018.

Our platform does not stop at June as we plan to introduce newer functionalities for Browsers Lab. We plan to release API Automation soon after our production release. This will lead us to build mobile platforms to make it easier for developers to code. Integrating our Payments Platform will be our next step which will give the ability to pay for services and employees using BAL tokens. As Browsers Lab grows, so will our features and enhancements.

Browsers Labs Token

The BAL token is a standard ERC20 token. This token can be used by multiple trading platforms and wallets. BAL token will be one of the ways to interact with our platform. Our token will allow users to employ our services, pay/reward their employees, hire freelancers, etc.

Browsers Lab smart contract contains all the security and safety provided by Ethereum's blockchain technology. The smart contract meets all of the standards for an ERC-20 token(transfer, transfer from, approve, approveandcall, etc.). We have also added our own functionality to help put our contributors mind at ease. During our sales, some of our tokens will need to be frozen for a set period of time. In order for us to accomplish this, we have created the **Freeze** and **Unfreeze** function with a time limit.



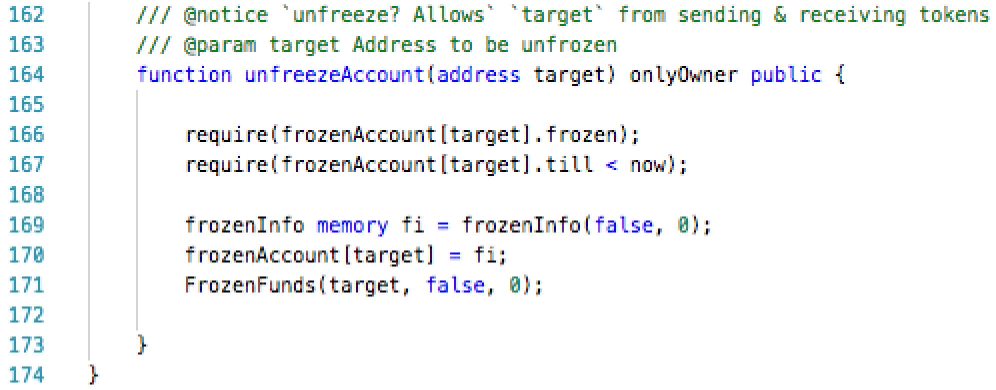
```
/// @notice `freeze? Prevent` `target` from sending & receiving tokens
/// @param target Address to be frozen
/// @param till Timestamp frozen till
function freezeAccount(address target, uint till) onlyOwner public {

    require(!frozenAccount[target].frozen);

    frozenInfo memory fi = frozenInfo(true, till);
    frozenAccount[target] = fi;
    FrozenFunds(target, true, till);

}
```

Freeze Account functionality allows us to take an account ***(address target)*** and a timestamp ***(uint till)*** when the account should be frozen for. That information is then taken and added to the frozen account mapping.



```
    /// @notice `unfreeze? Allows` `target` from sending & receiving tokens
    /// @param target Address to be unfrozen
    function unfreezeAccount(address target) onlyOwner public {

        require(frozenAccount[target].frozen);
        require(frozenAccount[target].till < now);

        frozenInfo memory fi = frozenInfo(false, 0);
        frozenAccount[target] = fi;
        FrozenFunds(target, false, 0);

    }
}
```

Unfreeze account functionality takes an account ***(address target)***, verifies that the account is currently in the frozen state, checks to see if the time limit has reached, and only then does it unlock the account if those conditions are satisfied.

BAL Tokens Smart Contract and Crowd Sale Contract will be publicly available at https://github.com/browserslab/ico-contracts

The ICO

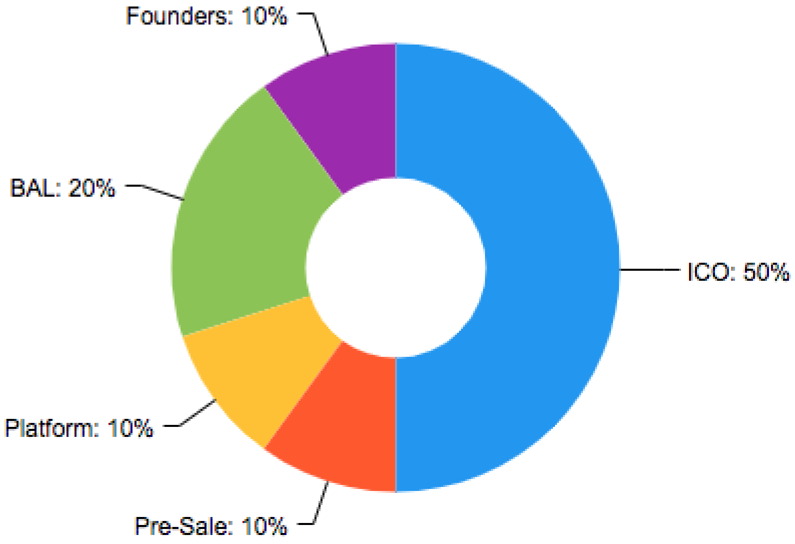
Browsers Lab team are fully committed to get this platform to production. We at Browsers Lab will not set any soft-cap for our ICO. Regardless of the outcome of our ICO, Browsers Lab will be pushed to production.

The pre-ICO will provide an opportunity for loyal contributors to join before the official ICO is launched. The pre-ICO will last 15 days from its start date and will remain open until the end of the 15 days. We will have a hard cap for our token in pre-ICO as they are all bonus based *(more on this later).* The tokens are distributed on first-come-first-served basis. Regarding disbursement, our tokens will be issued the moment the contribution is received. The timeline may be subject to delay due to unforeseen circumstances.

Total of **two billion (2,000,000,000) BAL tokens** will be issued in our smart contract but it will be allocated. Tokens will be allocated as shown below.

Allocation BL

Tokens will be allocated in multiple stages of our platform. BAL Tokens are intended to be allocated as follows:



Our initial pre-ICO will be launched on **March 19th 2018** and the main ICO will follow at **April 23rd 2018**. The maximum crowd sale cap is set to 60% (1,200,000,000 BAL) of our total supply, of which 200,000,000 BAL is for pre-ICO and 1,000,000,000 is for ICO. The accepted contributions tokens will be ETH and BTC.

Breakdown:

- ➔ 10% (200,00,000) Reserved for founders. Founders tokens will be locked for 6 months after the initial contract deployment.
- ➔ 10% (200,000,000) will be sold directly on the platform for consumer of our services. These tokens will be locked for 6 months after the initial contract deployment.
- ➔ 20% (400,000,000) are reserved by the Company to incentivize community, bug bounty, beta testers and partners. These tokens will be locked for 6 months after the initial contract deployment. Afterwards, tokens will be unlocked in 50% increments per 6 months, during the next year to support future steering of the project.
- ➔ 10% (200,000,000) are reserved for pre-ICO. Any unsold tokens will be moved over to the main ICO event.
- ➔ 50% (1,000,000,000) are reserved for our crowd sale.

Pre ICO

An initial pre-ICO will be launched in **March 19th 2018**. The company will be selling 10% (200,000,000) of the total supply at a wholesale price for early contributors. These contributions will be given a bonus. The exchange rate for **1 ETH** will be **40,000 BAL** (**40%** bonus)**.**

The minimum in pre-ICO phase will be set to 0.1 Eth.

For more information about our pre-ICO, contact us at investor@browserslab.com

ICO

The main ICO will start **April 23rd 2018**.

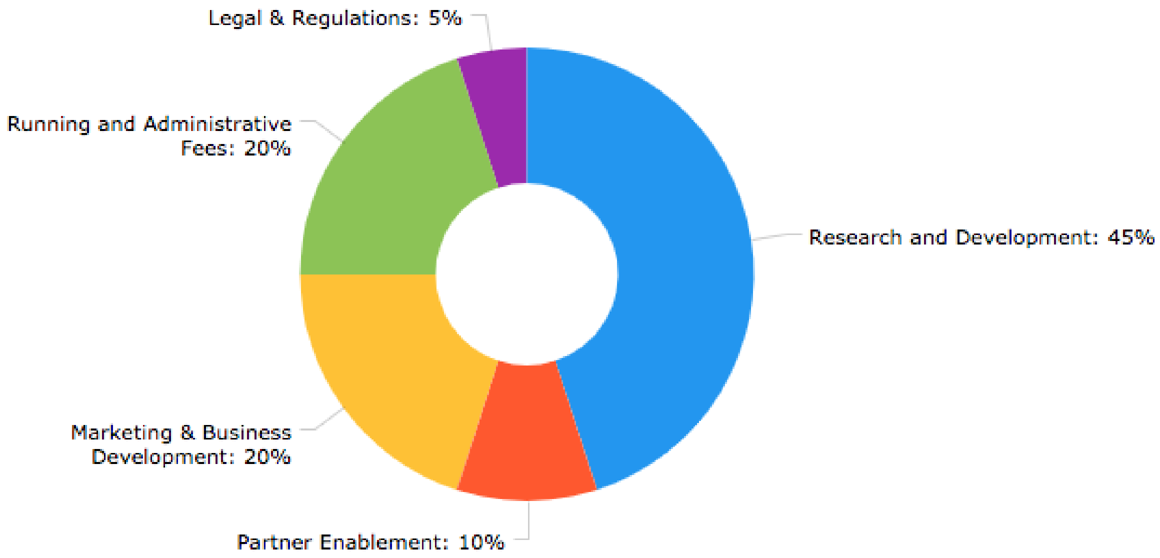
BAL Token bonus schedule will be as follows:

- ➔ Week 1: **32,775 BAL** for **1 ETH** (**15%** bonus)
- ➔ Week 2: **31,350 BAL** for **1 ETH** (**10%** bonus)
- ➔ Week 3: **29,925 BAL** for **1 ETH** (**5%** bonus)
- ➔ Week 4: **28,500 BAL** for **1 ETH** (no bonus)

Usage of fund

This ICO will help us hire talented individuals, set-up our infrastructure, pay for marketing as well as pay for our product and business development so we can be the first platform to offer cloud automation.

The post-ICO breakdown is as follows:



The values and percentages are:

- ➔ 45% will be used for Research and Development
    - ◆ A significant part of our funds will be allocated for infrastructure, R&D, personnel, and development of Browsers Lab
    - ◆ Hiring technical resources and establishing the support infrastructure
- ➔ 10% will be used for Partner Enablement
    - ◆ Develop relationships with consumers.
    - ◆ Helping engage Browsers Lab in joint ventures
- ➔ 20% will be used for Running and Administrative Fees
    - ◆ This allocation will allow us to deal with the operational and administrative expenses of our platform
    - ◆ Providing 24/7/365 technical support
    - ◆ Auxiliary personnel for administrative department
- ➔ 20% will be used for Marketing & Business Development
    - ◆ This allocation will help bring companies to our platform and drive platform adoption.
    - ◆ Developing relationships, marketing plans and implementation of marketing strategies to increase clientele
- ➔ 5% will be used for Legal & Regulations
    - ◆ Establishing foreign legal entities
    - ◆ Setup of our regulatory framework

Project Development Phases

Browsers Lab project is broken down in multiple phases:

- Alpha Release
    - Initial setup of our platform
        - Setup the initial infrastructure for deployment
        - Getting the deployment process setup
        - Automatic builds for our platform
    - Basic User Interface built to showcase the platform
        - Getting the User Interface built
        - Fixing issues with the UI for a stable alpha environment
    - Getting user input in the platform for UI design and ease of use
        - User input for setting priorities for our development team
        - Gathering requirements from user input for further improvements.
    - Modifying the platform for UX/UI
        - Making our platform as user friendly as possible
- Beta Release
    - Add code enhancements for the backend
        - Fix outstanding bugs in our platform
        - Code cleanup
        - Setting a set development and deployment structure
    - Scalability setup for our environment
        - The ability to scale our environments in a matter of minutes
        - Taking the human element out of scalability
    - Continuous integration built into our platform
        - Automatic deployments for our environments
        - Setup code build processes
    - User Inputs for a more user friendly platform
        - User input is very important to Browsers Lab. We believe in making the development on our platform as easy as possible.
- Production
    - Expansion of our grids
        - Setup up multiple grids for the increased number of users
        - Multiple browser support
    - Expansion of our infrastructure
        - Making Browsers Lab a smooth and simple place to manage your automation
        - Ensure performance is not affected and all users can access their platform.
- Post-Production
    - Features
        - Add external tools for building
            - Desktop Application

        - iOS Application
- Payment Platform Integration
    - The ability to use our platform for a payment system for your employees and freelancers
    - Giving thank you bonuses from our platform
- API Integration
    - Allowing users to test their APIs without having to deal with multiple other tools.
    - This will make it easy enough for people to write and execute their tests without the need of tools on their local machines.
- Mobile Automation
    - Mobile development has been an upcoming industry but rarely does it get the support it needs. Our platform will allow you to execute your tests against your application.

As Browsers Lab grows, so will our feature and offerings. We plan to expand in our support for the automation industry by helping provide a stable tool. The plan stated above is subject to change.

Conclusion

Browsers Lab is cloud-based automated testing platform. Browsers Lab allows users to store & execute tests in the cloud on different browser platform while providing a comprehensive test infrastructure for automated desktop and mobile application using Selenium, Appium and JavaScript unit testing frameworks.

Browsers Lab provides infrastructure to store all your test scripts and run them on the same platform. We even make it easy for a non-technical tester to select the tests and environment that has been written by some other team members and run them and view the results. We provide the API automation, which makes it easy for a developer to run their API test. Mobile applications development is on the rise but we rarely see any good testing platforms. We plan to offer our users the ability to run mobile automation against emulators and real devices.

Reference

1. Divya Kumar and K.K. Mishra, *The Impacts of Test Automation on Software's Cost, Quality and Time to Market*(Procedia Computer Science 79 (2016) 8 – 15)
2. https://www.zionmarketresearch.com/news/test-automation-market(*Test Automation Market by Test Type, Global Industry Perspective, Comprehensive Analysis, and Forecast, 2016 - 2022, ZMR-1937)*