MYTHBUSTING:

Every project needs to be automated

The owner of an assurance multinational company, recently hired a QA Automation team to automate their new mobile app. There are two applications, one on Android and one on iOS. The functionality is the same on both applications and the apps are still in development. The client is waiting for a response regarding the implementation of the project. No contract was signed yet.

How would you implement the automation testing process?

We will search on the internet for awesome automation tools and gather data about what tools we need to use and what language. Then we will start creating a project, automate some workflows and boom:

we have a prototype to give to the client

Is that ok?
Is our decision mature enough?

What questions may arise?

- Do we have a process?
- How do we select the right tool?
- Do we need to integrate the project with other tools?
- Do we need to allocate budget to buy software?
- Are the tests maintainable?
- How much time do we need to automate the manual tests?
- Do we need to use a test management tool?
- How will we track defects?
- Do we have enough resources?
- When and how do we start?

How to choose the test framework?

Cucumber
nunit
Protractor
end to end testing for AngularJS
pytest
appium
Ranorex®
Codeception
Se
JUnit 5.2
UI TESTING FOR ANDROID
espresso
TestNG

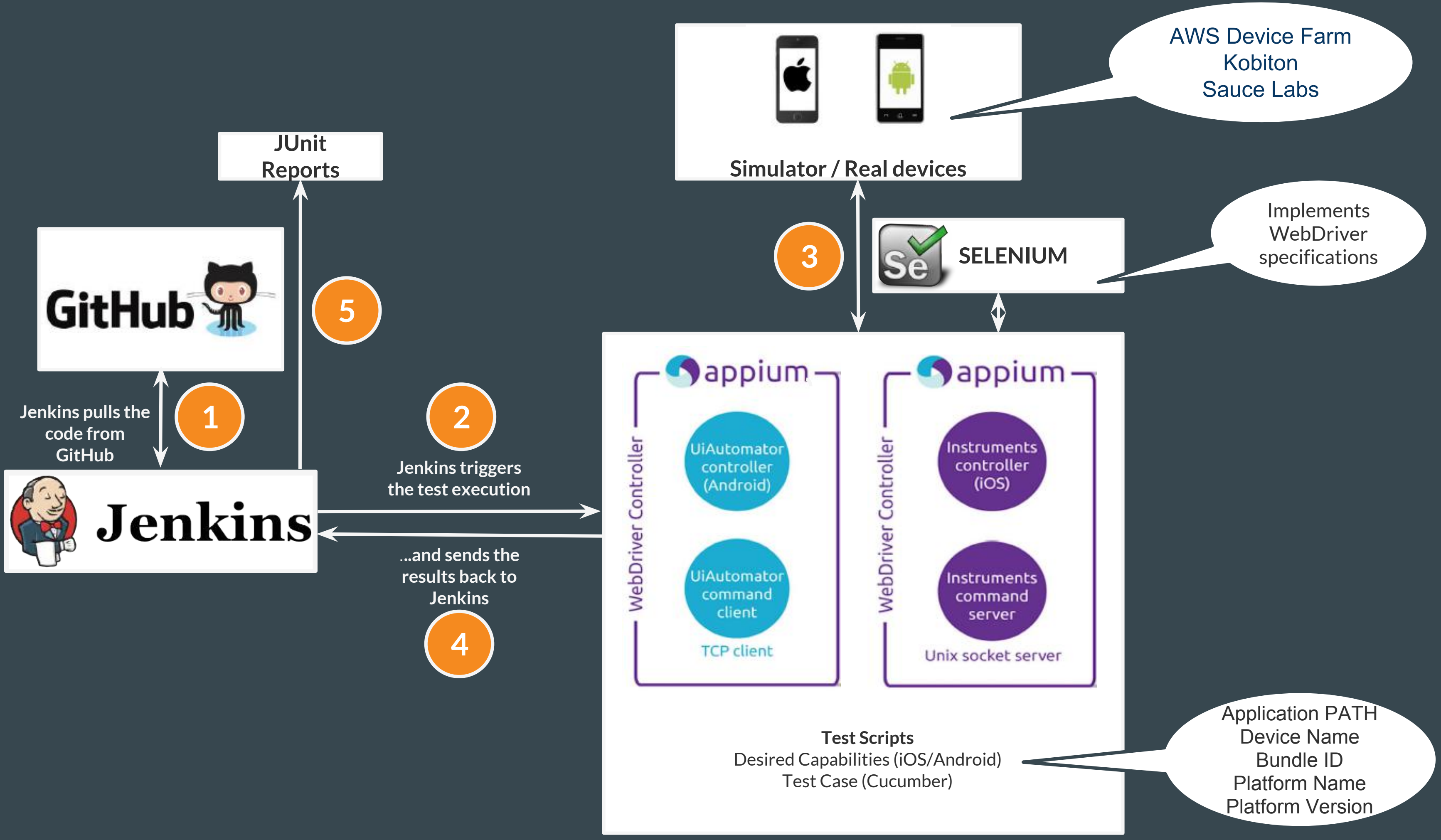
The owner of an assurance multinational company, recently hired a QA Automation team to automate their new mobile app. There are two applications, one on Android and one on iOS. The functionality is the same on both applications and the apps are still in development. The client is waiting for a demo and a response regarding the time and budget that needs to be allocated for the project. No contract was signed yet.

Which to choose? Why?

- To be stable
- XCTest/XCUITest
- Appium
- Calabash
- EarlGrey
- Espresso
- Robotium

How to choose the language?

1. Testers knowledge
2. Project's programming language
3. Programming language community of the testing framework (e.g. Selenium)
    - Java 68%
    - JS 15%
    - Python 11%
    - Ruby 5%
    - C# 1%
4. Run tests in parallel without any pain



JUnit Reports

5

GitHub

Jenkins pulls the code from GitHub

1

Jenkins

2

Jenkins triggers the test execution

...and sends the results back to Jenkins

4

Simulator / Real devices

AWS Device Farm
Kobiton
Sauce Labs

3

SELENIUM

Implements WebDriver specifications

appium

WebDriver Controller

UiAutomator controller (Android)

UiAutomator command client

TCP client

appium

WebDriver Controller

Instruments controller (iOS)

Instruments command server

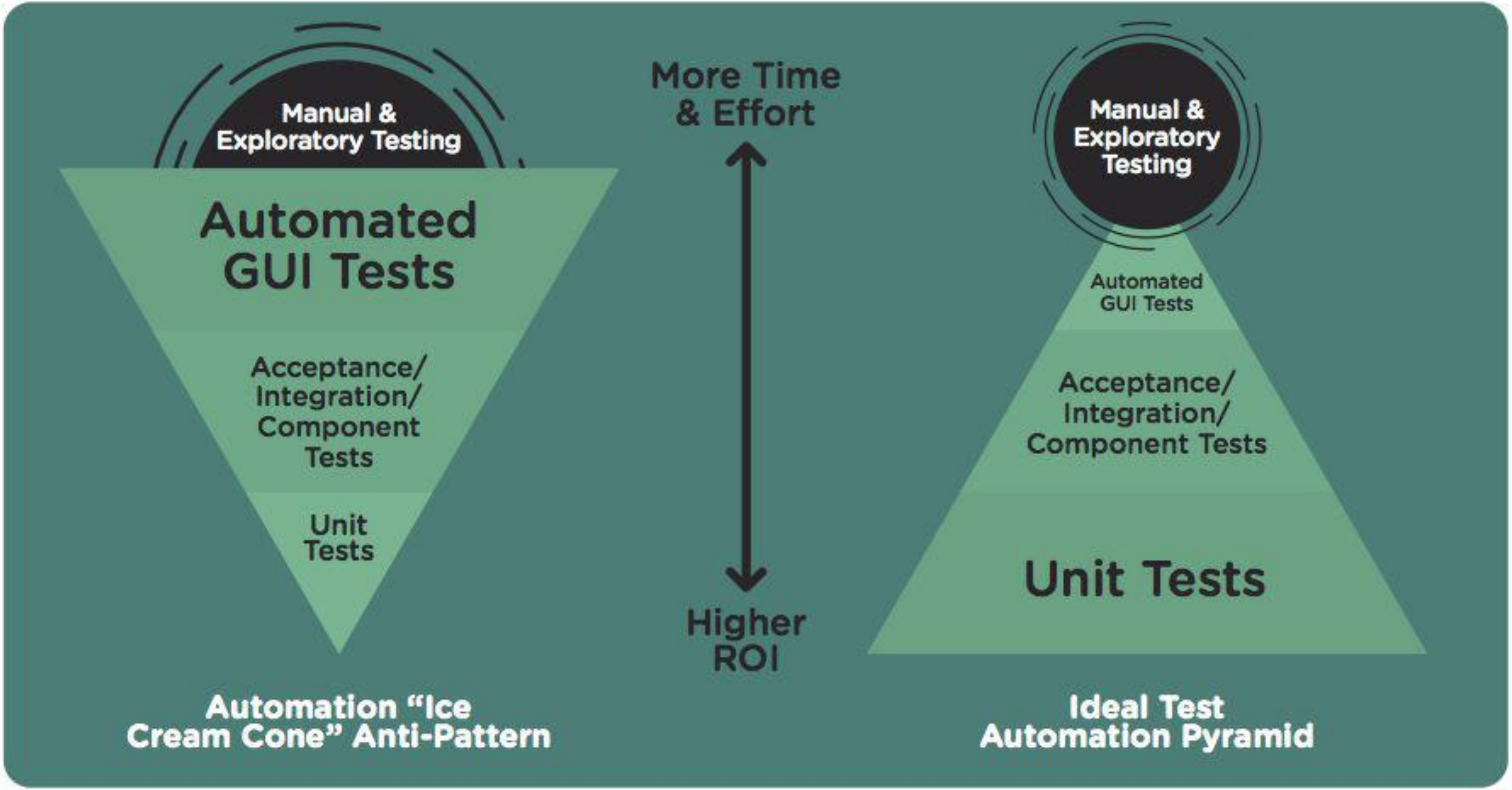
Unix socket server

Test Scripts
Desired Capabilities (iOS/Android)
Test Case (Cucumber)

Application PATH
Device Name
Bundle ID
Platform Name
Platform Version

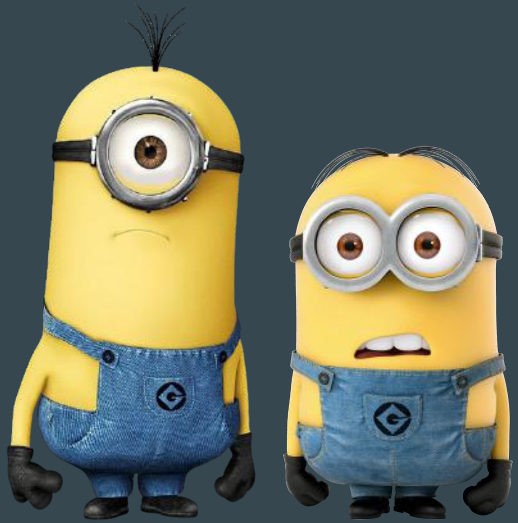
#fact1 think about the test framework, programming language & infrastructure

But there is more than one type of automation testing...


Manual & Exploratory Testing
Automated GUI Tests
Acceptance/ Integration/ Component Tests
Unit Tests
Automation "Ice Cream Cone" Anti-Pattern
More Time & Effort
Higher ROI
Manual & Exploratory Testing
Automated GUI Tests
Acceptance/ Integration/ Component Tests
Unit Tests
Ideal Test Automation Pyramid

#fact2 automation is done at 3 levels

Why do we automate?



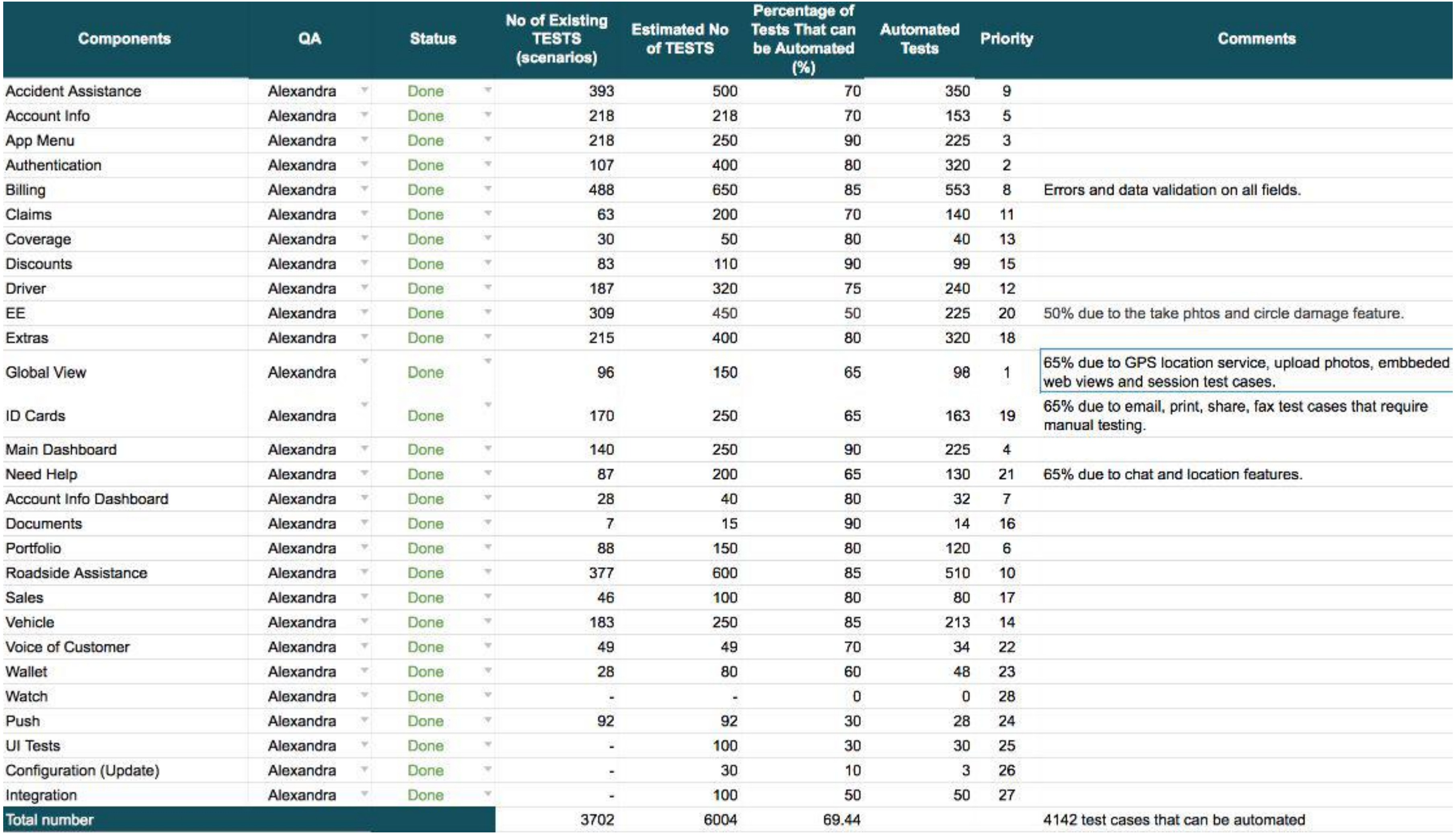
Because we want to...

- automate complex web applications
- quickly run critical & high priority tests
- quickly run tests that need to be executed on multiple configurations
- eliminate the regression tests
- automate tests that are prone to human error
- automate data-driven tests
- increase test coverage
- decrease costs

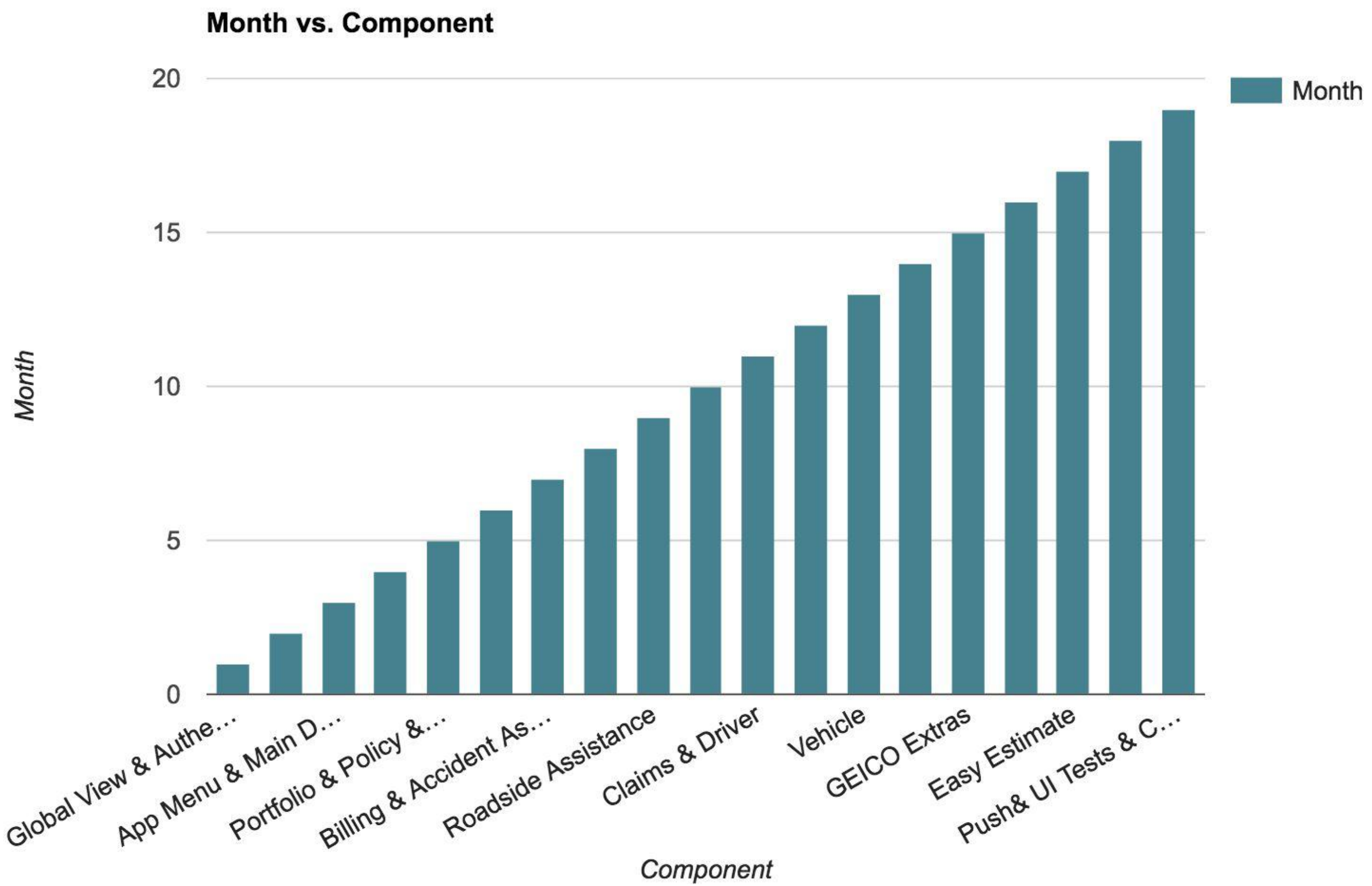
#fact3 get quicker results (save money and time)

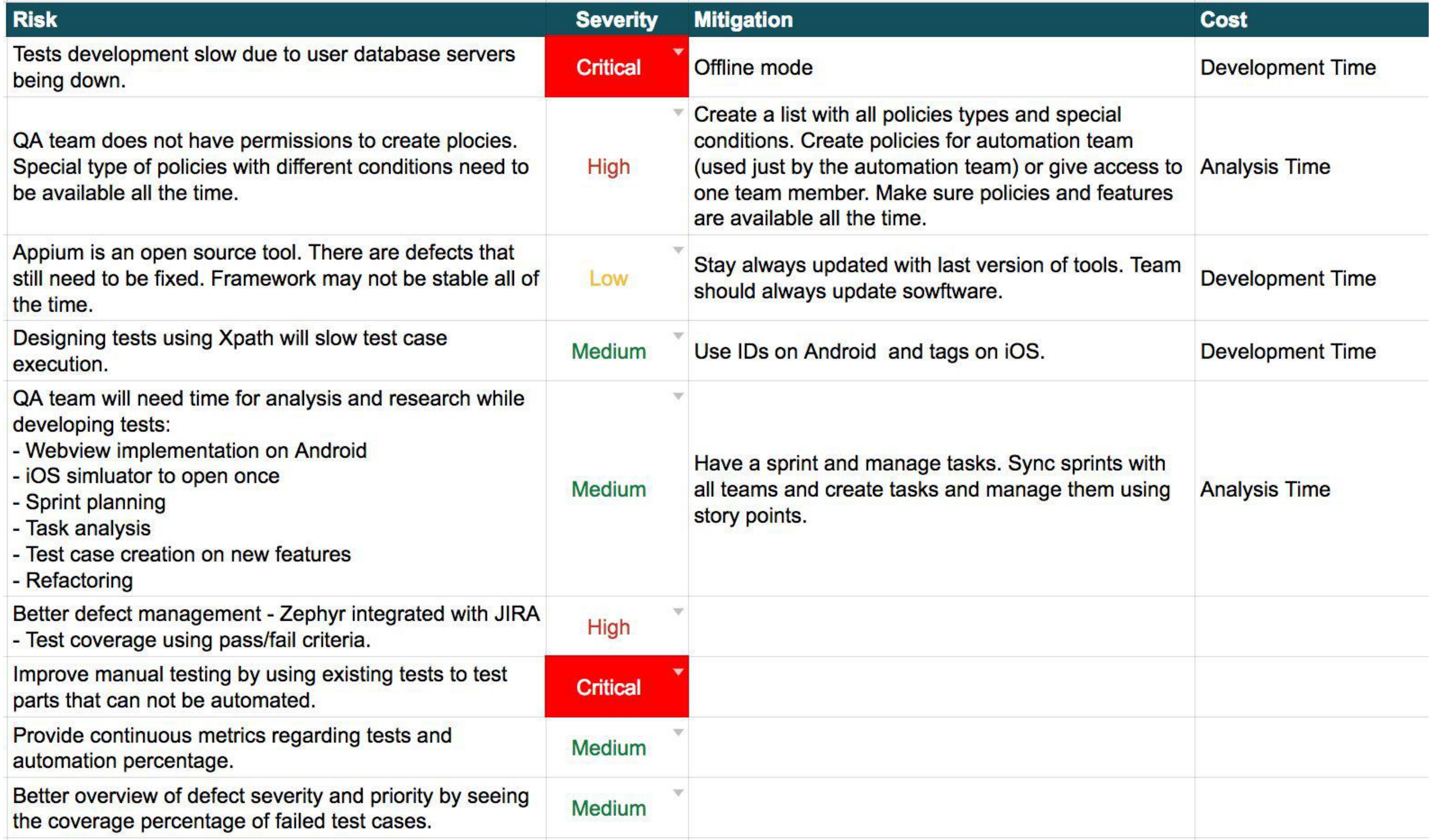
So how do you decide to automate or not?

Success in test automation requires careful planning and design work.

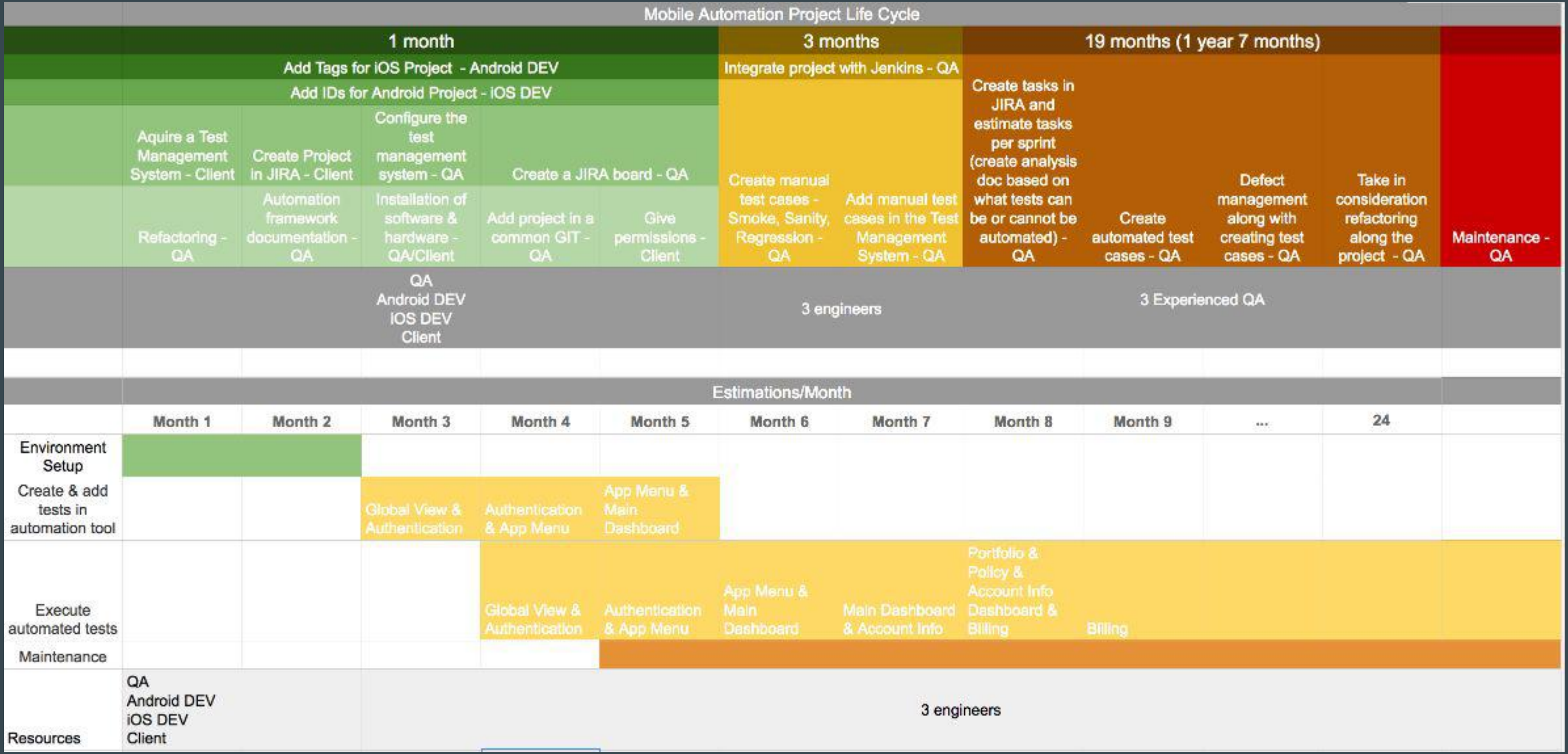


Components	QA	Status	No of Existing TESTS (scenarios)	Estimated No of TESTS	Percentage of Tests That can be Automated (%)	Automated Tests	Priority	Comments
Accident Assistance	Alexandra	Done	393	500	70	350	9	
Account Info	Alexandra	Done	218	218	70	153	5	
App Menu	Alexandra	Done	218	250	90	225	3	
Authentication	Alexandra	Done	107	400	80	320	2	
Billing	Alexandra	Done	488	650	85	553	8	Errors and data validation on all fields.
Claims	Alexandra	Done	63	200	70	140	11	
Coverage	Alexandra	Done	30	50	80	40	13	
Discounts	Alexandra	Done	83	110	90	99	15	
Driver	Alexandra	Done	187	320	75	240	12	
EE	Alexandra	Done	309	450	50	225	20	50% due to the take phtos and circle damage feature.
Extras	Alexandra	Done	215	400	80	320	18	
Global View	Alexandra	Done	96	150	65	98	1	65% due to GPS location service, upload photos, embbeded web views and session test cases.
ID Cards	Alexandra	Done	170	250	65	163	19	65% due to email, print, share, fax test cases that require manual testing.
Main Dashboard	Alexandra	Done	140	250	90	225	4	
Need Help	Alexandra	Done	87	200	65	130	21	65% due to chat and location features.
Account Info Dashboard	Alexandra	Done	28	40	80	32	7	
Documents	Alexandra	Done	7	15	90	14	16	
Portfolio	Alexandra	Done	88	150	80	120	6	
Roadside Assistance	Alexandra	Done	377	600	85	510	10	
Sales	Alexandra	Done	46	100	80	80	17	
Vehicle	Alexandra	Done	183	250	85	213	14	
Voice of Customer	Alexandra	Done	49	49	70	34	22	
Wallet	Alexandra	Done	28	80	60	48	23	
Watch	Alexandra	Done	-	-	0	0	28	
Push	Alexandra	Done	92	92	30	28	24	
UI Tests	Alexandra	Done	-	100	30	30	25	
Configuration (Update)	Alexandra	Done	-	30	10	3	26	
Integration	Alexandra	Done	-	100	50	50	27	
Total number			3702	6004	69.44			4142 test cases that can be automated


Month vs. Component
Month
20
15
10
5
0
Month
Global View & Authe…
App Menu & Main D…
Portfolio & Policy &…
Billing & Accident As…
Roadside Assistance
Claims & Driver
Vehicle
GEICO Extras
Easy Estimate
Push& UI Tests & C…
Component



Risk	Severity	Mitigation	Cost
Tests development slow due to user database servers being down.	Critical	Offline mode	Development Time
QA team does not have permissions to create plocies. Special type of policies with different conditions need to be available all the time.	High	Create a list with all policies types and special conditions. Create policies for automation team (used just by the automation team) or give access to one team member. Make sure policies and features are available all the time.	Analysis Time
Appium is an open source tool. There are defects that still need to be fixed. Framework may not be stable all of the time.	Low	Stay always updated with last version of tools. Team should always update sowftware.	Development Time
Designing tests using Xpath will slow test case execution.	Medium	Use IDs on Android and tags on iOS.	Development Time
QA team will need time for analysis and research while developing tests: - Webview implementation on Android - iOS simluator to open once - Sprint planning - Task analysis - Test case creation on new features - Refactoring	Medium	Have a sprint and manage tasks. Sync sprints with all teams and create tasks and manage them using story points.	Analysis Time
Better defect management - Zephyr integrated with JIRA - Test coverage using pass/fail criteria.	High		
Improve manual testing by using existing tests to test parts that can not be automated.	Critical		
Provide continuous metrics regarding tests and automation percentage.	Medium		
Better overview of defect severity and priority by seeing the coverage percentage of failed test cases.	Medium		



Mobile Automation Project Life Cycle

1 month						3 months		19 months (1 year 7 months)				
Add Tags for iOS Project - Android DEV						Integrate project with Jenkins - QA						
Add IDs for Android Project - iOS DEV												
	Aquire a Test Management System - Client	Create Project in JIRA - Client	Configure the test management system - QA	Create a JIRA board - QA								
	Refactoring - QA	Automation framework documentation - QA	Installation of software & hardware - QA/Client	Add project in a common GIT - QA	Give permissions - Client	Create manual test cases - Smoke, Sanity, Regression - QA	Add manual test cases in the Test Management System - QA	Create tasks in JIRA and estimate tasks per sprint (create analysis doc based on what tests can be or cannot be automated) - QA	Create automated test cases - QA	Defect management along with creating test cases - QA	Take in consideration refactoring along the project - QA	Maintenance - QA
			QA Android DEV iOS DEV Client			3 engineers		3 Experienced QA				

Estimations/Month

	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Month 7	Month 8	Month 9	...	24	
Environment Setup	🟩	🟩										
Create & add tests in automation tool			Global View & Authentication	Authentication & App Menu	App Menu & Main Dashboard							
Execute automated tests				Global View & Authentication	Authentication & App Menu	App Menu & Main Dashboard	Main Dashboard & Account Info	Portfolio & Policy & Account Info Dashboard & Billing	Billing			
Maintenance					🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧
Resources	QA Android DEV iOS DEV Client		3 engineers									

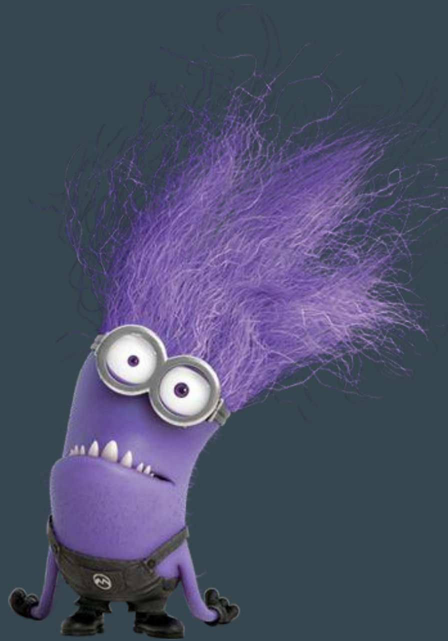
#fact4 start by doing a thorough analysis

Advantages of automation

- Testers get motivated
- Get fast information about defects introduced by changes
- Quick feedback (unit tests/integration tests)
- Frees up the time of the testers
- The development team can contribute

Disadvantages of automation

- False sense of quality => passing tests may not always be correct
- False positive or false negative tests
- Flaky tests => tests which fail or pass for the same config
- Maintenance time and effort => update the packages always
- Slow feedback => automated tests can take a long time to script
- Not many bugs found



#fact5 test automation when done correctly can have many advantages and be very beneficial

Automation in SDLC models

- Waterfall
- V-Model
- Prototype
- Spiral
- Iterative
- Agile
    - EP
    - Scrum
    - Kanban
    - Lean Development
    - Spotify Agile Model

Automation Process

- Communication
- TDD or BDD?
- Link automate tests with manual tests
- How to work on tasks?
- Code review
- Commit (style guide/code conventions PEP 8, Oracle)

Future of automation

- The arise of open source tools
- Big data testing of great importance
- QA as a Service?
- IoT testing
- The third wave: AI (deprecated)

#fact6 future trends can affect the project on the long run

The business behind the project

- Outsourcing or Product company
- Old or new client
- Do you have a contract or not?
- Time vs Budget
- Quality vs Quantity

Think about:

1. Complexity of the system
2. Infrastructure within the company
3. The ideal test automation pyramid
4. Budget
5. Team & knowledge
6. Time
7. Methodology used
8. The automation process
9. Trends
10. Business behind the project

Case 1: An **outsourcing** company is working on a project for a public broadcasting service provider. The project consists in websites that manage the content of the frontend and backend application. All the **infrastructure is within the outsourcing** company. The client is an **old client** with a **contract** that was **signed** for the **next 3 years**. The team is formed of **skilled technical individuals** and the team is working with the **Kanban** methodology.

Case 2: A **product company** is working on a new project consisting of a financial service mobile application. The application needs to be available on both **Android and iOS**. The team is formed of **skilled automation testers**, manual testers, DEVs, BA, product owner, UX designer and a manager. A **prototype** was designed and the management decided to go on with the product. The **business** and **management** are **not technical** individuals. There is **no timeline in finishing the project**, just to **deliver** the **basic functionality** in **10 months**. The team is working with the **Scrum** methodology.

Case 3: An **outsourcing company** receives a **small project** to implement a mobile application both on **Android** and **iOS** for a gym, where people can make appointments for their courses or see their membership. The **client** will **take care of maintenance**. They just need the **application to be ready ASAP**. A **contract was signed** for **6 months**. The team is working with **Scrum** and it is formed by three developers from which one is the team lead and a **QA mid automation engineer**.

MYTH BUSTED!

To all Testers:

It takes super powers...

to do all the things you do.
Thanks for being awesome!

join at Sildo.com with #test2018