



W3

Test Automation

Wednesday, October 23rd, 2019 10:15 AM

Destroying the Horcruxes of Full-Stack Automation

Presented by:

Iryna Suprun

Xandr

Brought to you by:



888-268-8770 · 904-278-0524 - info@techwell.com - <http://www.starcanada.techwell.com/>

Iryna Suprun

Iryna started her career as a software engineer in 2004 in Ukraine, where she was born. She received her master's degree in computer science in 2006, and

敢慧 植獲⁴潰煤梲湯愠 雷 瑩⁹ 獮 匠 浥榆 映 獵 湯
挽浯 獵牴⁹ 整瑳 敲 球浩 捩 瑳浥 摯 摯
拽 愠 瀼懂晴牯潦 桴 潔敍 汰捩 吠 磁慥獲愠
楣 瑩眠 浩 潞 禳 浯 楨柿渠啤愠 癯 敌档 槲 敲瀼獫 桓
瑳整 槲 敬 愠獫牯啤浯 浯 愠柿櫛 畢 挽瑩档牡 愠
獲慥磁 档 物 浯 瑳 慣湮瑯 潞⁰整瑳 獮
禳 漠 瑩 囧 ← 慶据 械据濫敬 景 吠 梲柿簾 牡⁴ 樑 磁
峻 整 械 愠灰楛慣梲湯 潭 柿映 潭潮楛桴捩 業物獮 散 牡档
瑩挽畴敲業牧 梲柿愠灰 h 捌 M 愠 映滿 戢 雷
瑩 眈 惛楫柿 据慶概 峻磁散時 槲 楷 潦獫牯眠 梲 槲 獫 獫桔
袷懂映 懂 捌歡敲戢潞 潦 整 浚整 瀼湧煤 敵煩溲挽
滿瀼峻磁 眠慶惛 癩 牡 瑳整 僂 汰浩 愠正 極 焯 瑳整
捩 湯 灸 据 湯瀼敲敦牲 捌 慰

Destroying the Horcruxes of Full-Stack Automation

The background image is a dark, atmospheric scene. In the center, a wooden bowl sits on a white cloth, with wisps of white smoke rising from it. To the left and right, several lit candles in various containers provide a warm, orange glow. In the foreground, an open book lies flat on a dark surface. The overall mood is mysterious and ritualistic.

Iryna Suprun

Automation Framework

API

Reporting

Tests

Other helping
modules and tools

ACCIO, BUGS!

System Under Test

UI

Real Time
Processing system

API #1

Reporting System

Kafka

API #2

Spark Jobs

Databases



Full-stack Automation Framework is like a magic spell:

- Complex system
- Requires constant investment of time and resources
- Well designed and architected
- Set of tools and language are carefully selected
- Has a lot of features to fit the needs of two or more agile teams that use it
- Works for multiple teams and technologies



Sometimes magic does not work

- Low rate of adoption, people just don't want to use it
- Low ROI
- Number of automated tests is not increasing
- Quality is not improved
- Nobody trusts test results

Is the soul of framework cursed?
We need to find and destroy the Horcruxes!

Automation Framework is a Software

very specific requirements

Body: Functionality

1. Test Runner
2. Test Statements, Classes and Objects
3. Asserts
4. Logging
5. Helpers...

Soul: Quality Attributes

1. Reliability
2. Usability
3. Communication
4. Test Data
5. Portability
6. Reporting
7. Integration

To be alive everything must have a soul !!!

Deadly Nagini's Bite: no trust

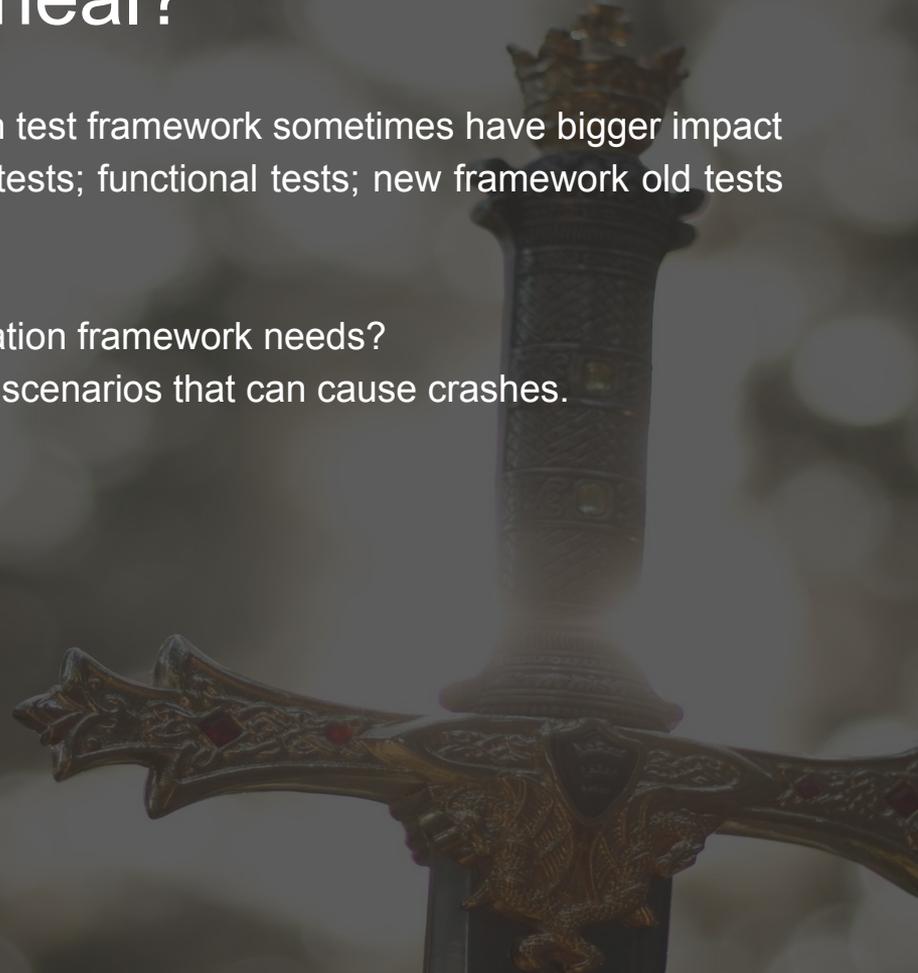
Reliability, Robustness

Test Automation Framework is like a medical software. Its goal is to provide insights on how healthy the system under test is. We all know that flaky tests are bad, but what about framework itself?

- Will you trust the medical device that crashes time to time?
- Will you trust the medical software that was not thoroughly tested?
- Will you trust the medical software that gives you different results after update?
- Will you use technology that was only tested on mice, and never on people?
- Will you use application to monitor your heart if it has bugs?

How to heal?

- Test Framework should be TESTED. Bugs in test framework sometimes have bigger impact on quality than bugs in software code. Unit tests; functional tests; new framework old tests and old code
- Code Reviews
- Are you sure that CI/CD is what your automation framework needs?
- Invest into robustness. Think about possible scenarios that can cause crashes.
- Handle Exceptions!



Salazar Slytherin's Locket: I hate it!

When you build automation framework did you think about users? Does your framework act like Salazar Slytherin's Locket and brings the worst emotion?

- I need to spend two hours to install it, for this I need to download 10 dependencies and install very old version of MySQL. Installation instruction has 20 steps.
- I need two weeks to learn how to add new tests.
- I cannot run test where I want when I want.
- I cannot easily select tests I want to run
- It is slow!
- Updating tests is the pain in the ...
- Errors that give me no insights what I did wrong

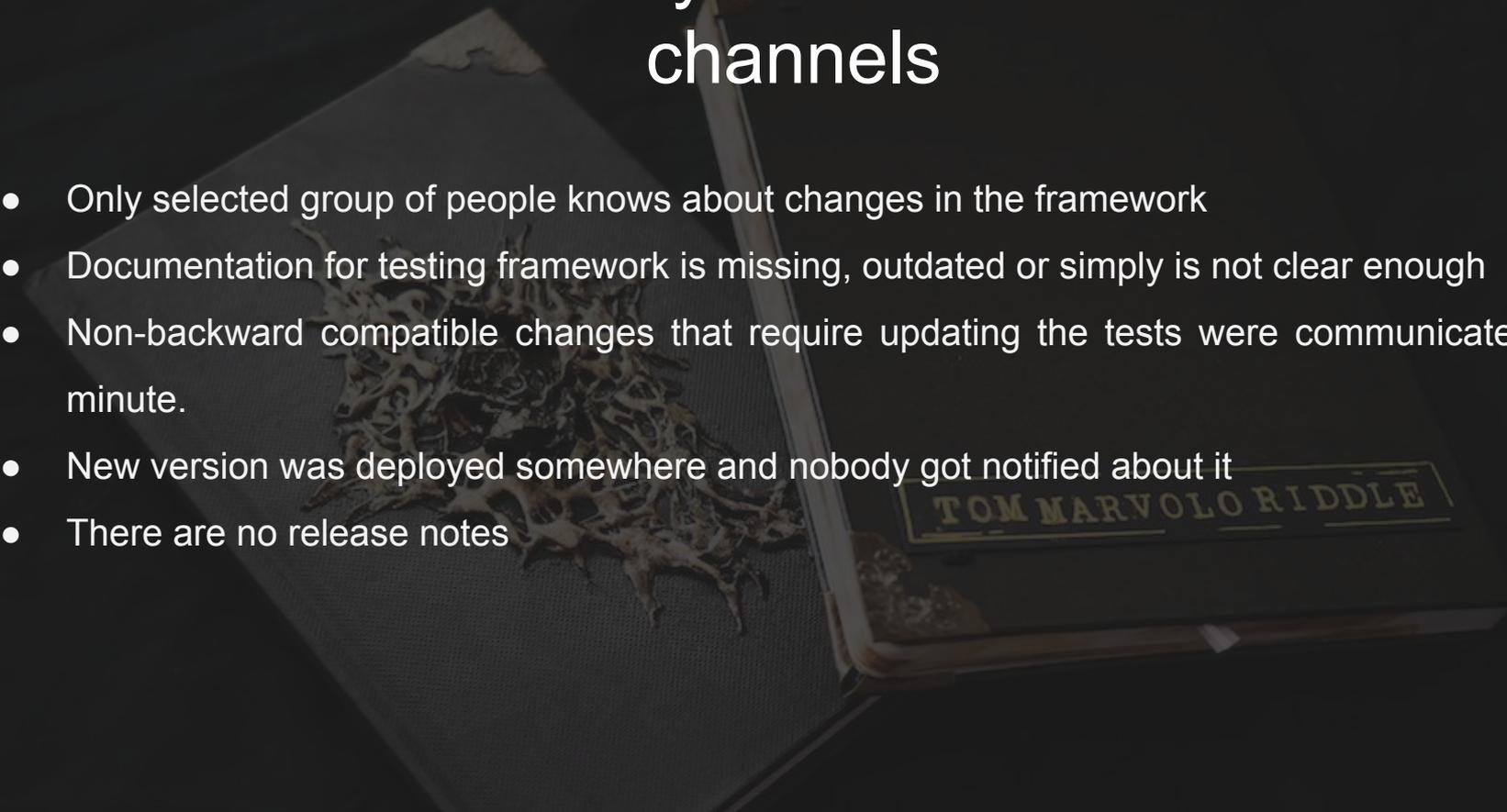


Salazar Slytherin's Locket: take it off

- Installation and initial setup should be as easy as possible. Take care about dependencies. Use containers technologies if needed.
- Simplify adding new tests. Use templates, examples.
- Where possible use tools and languages available on the market, not internal ones.
- Make sure that if user makes a mistake the error messages are crystal clear and provide full information about what had happened and how to fix it.
- Easy to pick and choose any subset of tests to execute (tip - use tags)
- Updating tests should be easy: use code design patterns, make sure that there is no code copy-paste, follow clean code practices
- Invest in speed: parallelize, find bottlenecks

Tom Riddle's Diary: broken communication channels

- Only selected group of people knows about changes in the framework
- Documentation for testing framework is missing, outdated or simply is not clear enough
- Non-backward compatible changes that require updating the tests were communicated last minute.
- New version was deployed somewhere and nobody got notified about it
- There are no release notes

A dark, textured diary with a gold-colored title label that reads "TOM MARVOLO RIDDLE". The diary is open, showing a dark, textured cover on the left and a page with a large, intricate, golden, web-like pattern on the right. The background is dark and slightly blurred.

TOM MARVOLO RIDDLE

Tom Riddle's Diary: be clear

- It is a software! So follow proper release cycle. Prepare Release Notes
- Communicate all non-backward compatible changes
- Invest in user documentation!
- Proper communication of environments upgrade

Marvolo Gaunt's Ring - powerful curse of test data

- Generating test data takes forever
- System's under test UI is used to generate test data
- Automation execution corrupts data in database
- It is not possible to distinguish generated data for different tests runs
- Test data is created manually
- Scripts use data in preset database and assume that it exists and it's good
- Data cleanup is not automated

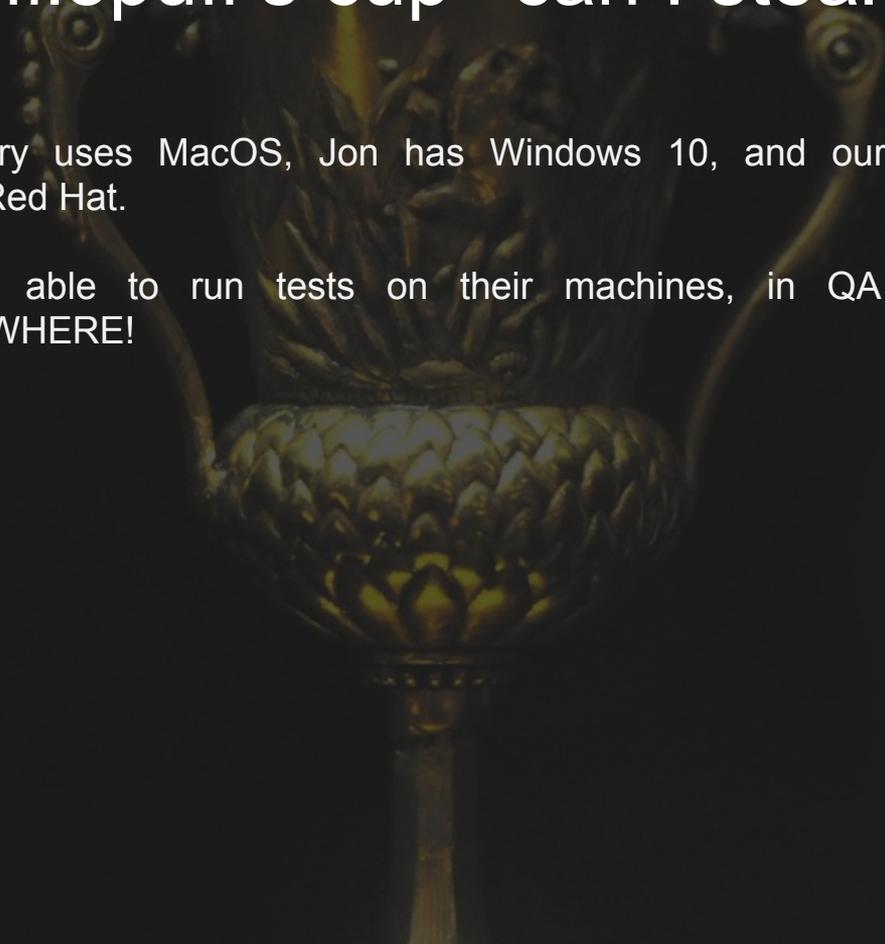
Marvolo Gaunt's Ring - break the curse

- Framework provides tooling for test data creation.
- Do not use UI to create test data
- Do not assume that some data exists
- Make sure that framework provides ability to distinguish between data created for different test runs (unique ids, time stamps, build number)
- Do cleanup but only when needed.
- Make sure that tests are not corrupting existing data (if it's required by test - restore it)

Hufflepuff's cup - can I steal it?

Jack uses Fedora, Mary uses MacOS, Jon has Windows 10, and our test environments are AWS/google cloud with Red Hat.

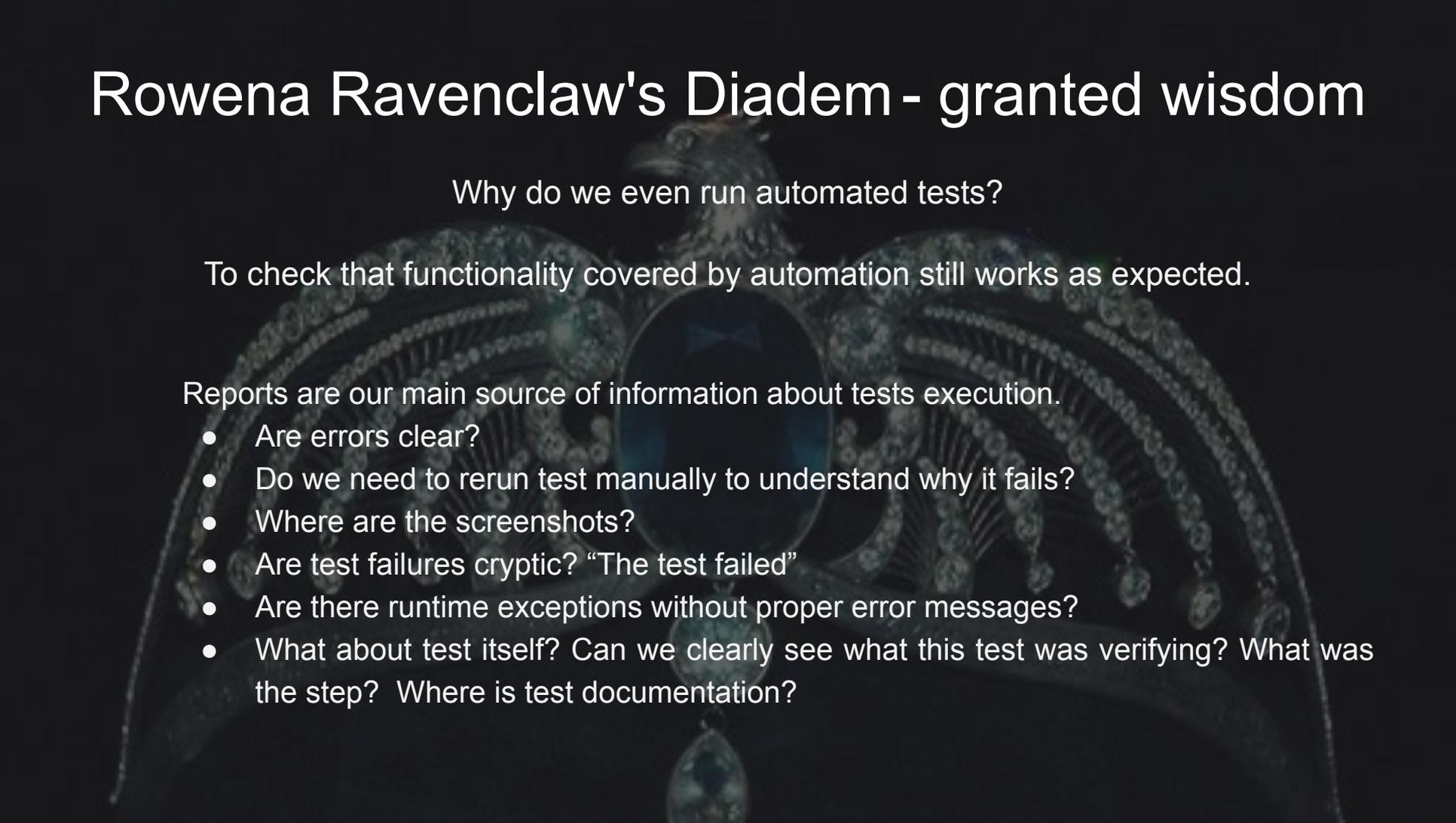
- Users should be able to run tests on their machines, in QA environment, in build pipeline....EVERYWHERE!



Hufflepuff's cup - can I steal it?

- Developers and Testers should be able to run the framework on Linux/Windows/MacOS. It might require taking care about file location, installation process
- Test it on different systems
- Use containerized solutions

Rowena Ravenclaw's Diadem - granted wisdom



Why do we even run automated tests?

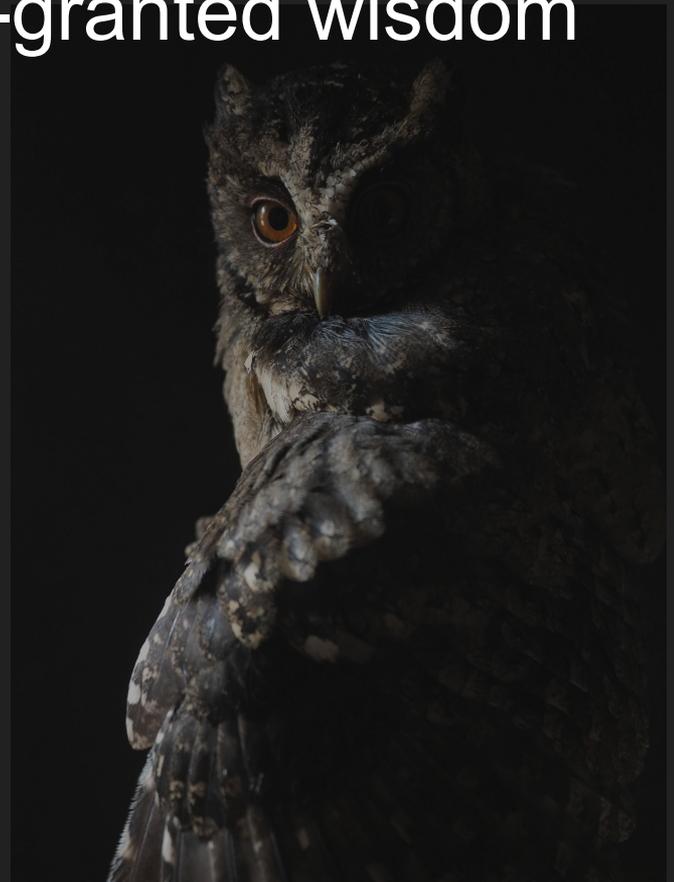
To check that functionality covered by automation still works as expected.

Reports are our main source of information about tests execution.

- Are errors clear?
- Do we need to rerun test manually to understand why it fails?
- Where are the screenshots?
- Are test failures cryptic? “The test failed”
- Are there runtime exceptions without proper error messages?
- What about test itself? Can we clearly see what this test was verifying? What was the step? Where is test documentation?

Rowena Ravenclaw's Diadem -granted wisdom

1. Reporting is one of the most important parts of the framework.
2. When test fails we need to be able to get all information about failure from the report.
3. Make sure that all errors are as informative as they can be.
4. Different levels of reporting for different consumers.
5. It should be crystal clear what functionality test verifies, what is the step it failed at, what was the input data.
6. Failed test and passed test should have different level of logs, and details in reports.
7. Clear test stats - how many tests were executed, pass/fail numbers, run time.



Harry Potter - center of it all

Can you integrate your framework into the CI/CD pipeline?

Can it easily be done or does it take 5 engineers and 2 weeks?

Can you easily spin up environment for tests to run?

Where are reports?

Do you fail the build based on the test results?

At what point do you run which tests?

Harry Potter - integration point

- You need to consider how you integrate test execution into CI/CD pipeline on the planning stage of framework
- Keep in mind that CI/CD tool can be changed
- Automation framework itself should not require complex setup
- Invest in convenient and easy to use automation framework CLI
- Will tests be executed in on-demand environment or preset environment?
On-demand requires more time to spin-up. How much time you want to wait each build? Deploy?

A dark, atmospheric tunnel with graffiti on the walls. In the center, a glowing rectangular frame with a white border and a blue and red inner glow contains the text "Q&A" in white. The tunnel floor is dark and reflective, and the walls are covered in various graffiti tags and designs.

Q&A