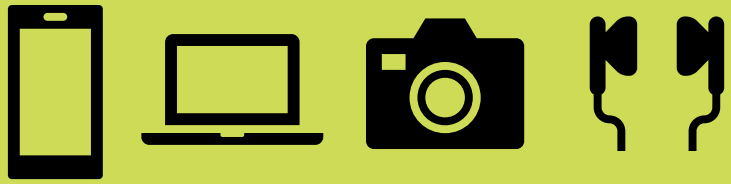
Increase the efficiency of test automation

2022

Business growing at XY company

XY Company



- E-Commerce company
- HQ: Hungary, Budapest
- Monthly visitors: 1,4 million
- Avg Visit Duration: 00:04:21
- Annual Revenue: ~ 10M EUR
- Countries where available: 8

„Selling with pleasure”

Mid term
(~2 years)

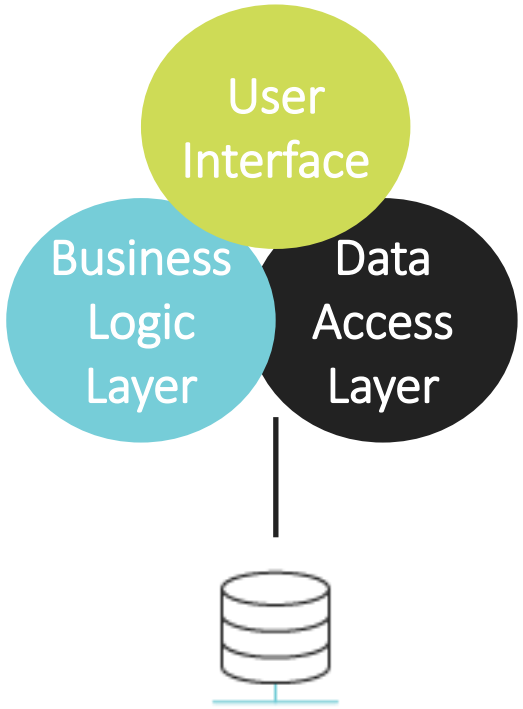
- Monthly visitors: **5+ million**
- Avg Visit Duration: **5+ minutes**
- Annual Revenue: **50+ M EUR**
- Countries where available: **30+**
- **Shorter time to market**
- **Phase1: New flexible webshop**
- **Phase2: New Mobile App**
- **Better performance / more secured**

Long term
(~5 years)

- Monthly visitors: **30+ million**
- Avg Visit Duration: **6+ minutes**
- Annual Revenue: **Up to 200M EUR**
- Countries where available: **80+**
- **Top 5 E-Commerce in Europe**

Architecture upgrade

Monolith



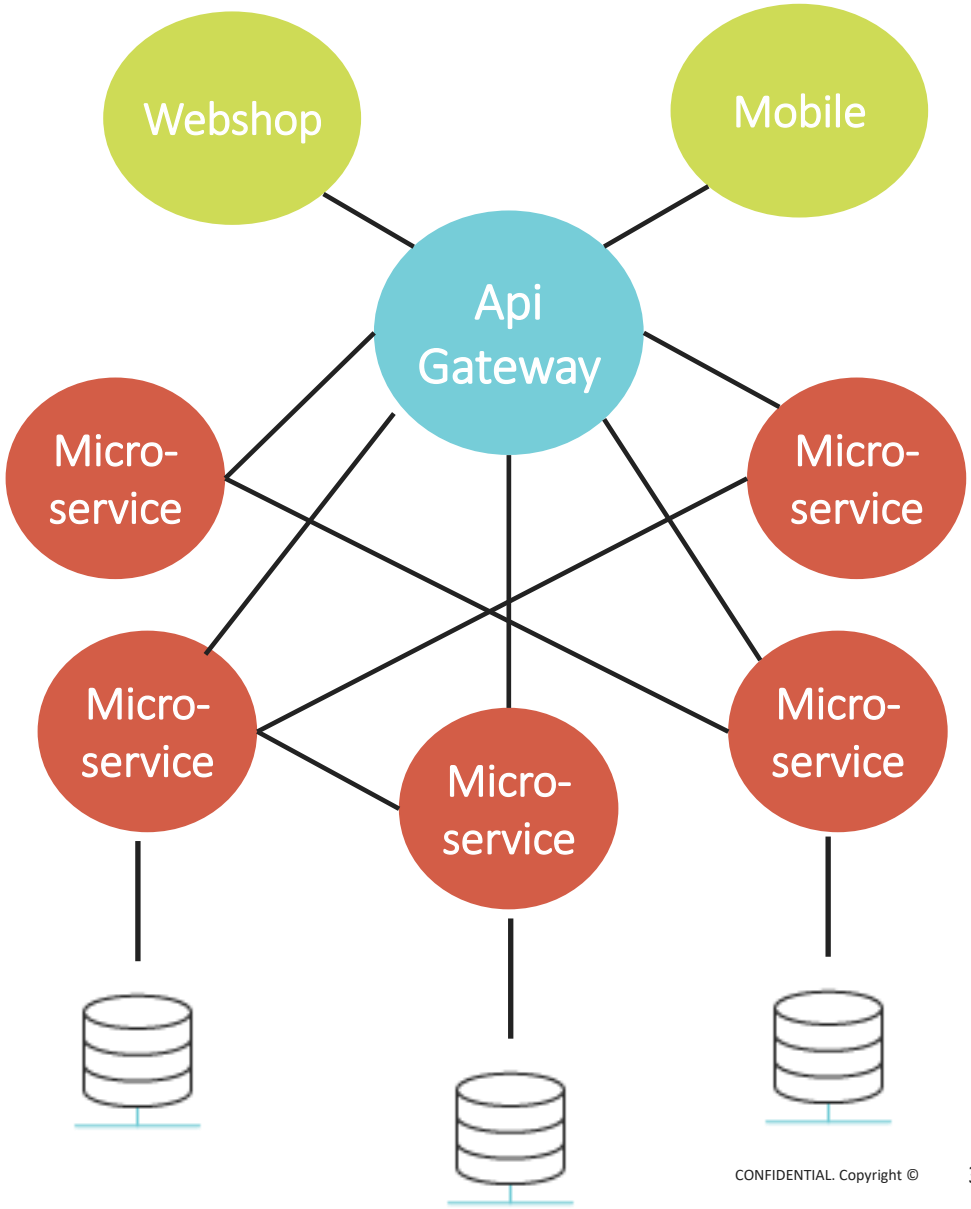
Microservice architecture Pro:

- Independency in technology/changes/during operation/scaling
- Small size, easy to manage/start up/onboard new team member

Microservice architecture Cons:

- More complex to manage architecture/deployment/testing/ debugging
- More network usage and less secured due to network communication

Microservice

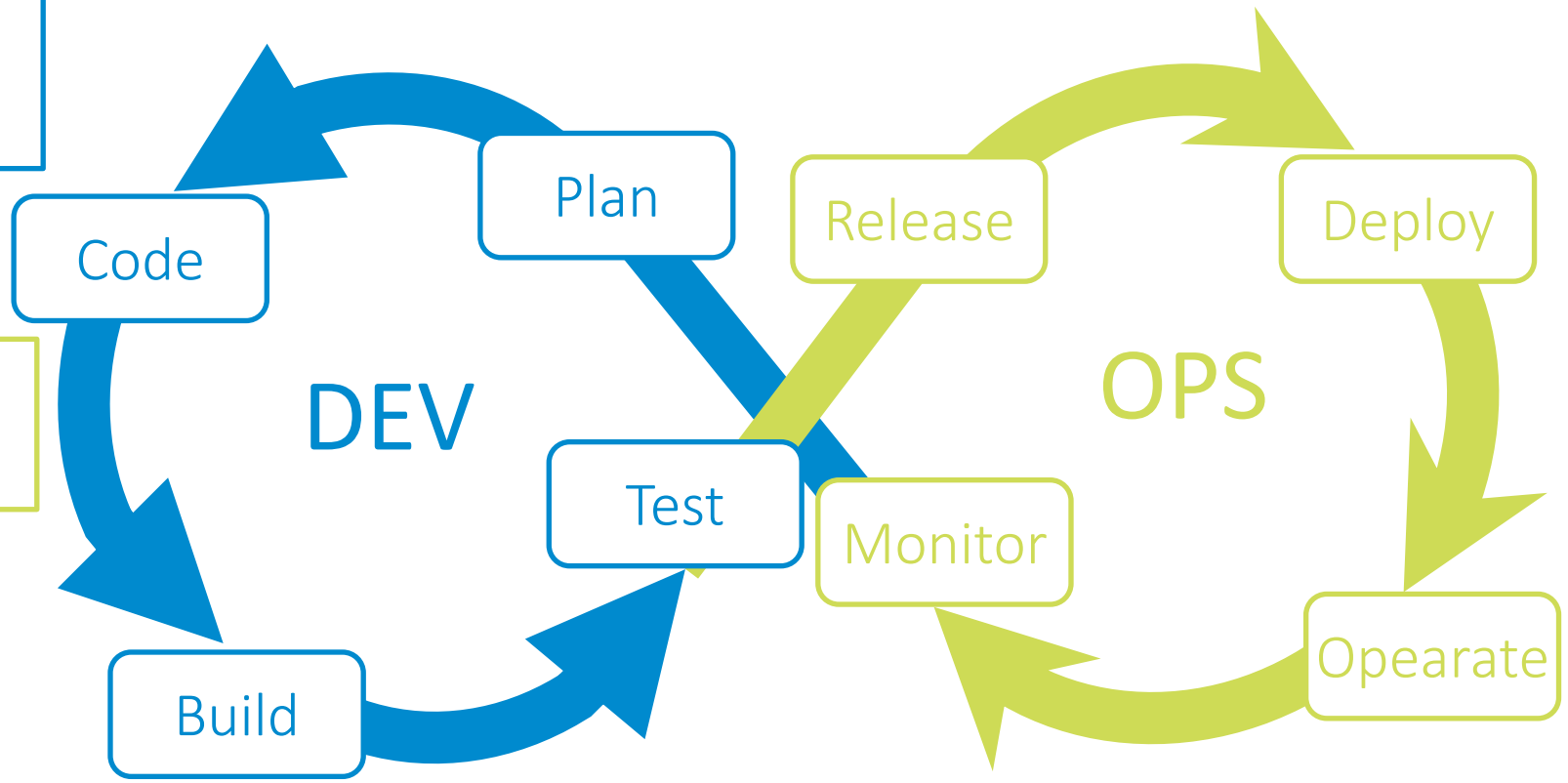


DevOps is a key to reach business goals with the architecture upgrade

DevOps is a combination of tools, practicies, philosophies which aims to deliver softwares with shorten SDLC, high quality, continous frequent deliveries.

Automating builds, tests, processes, reports, report evaulations by using various tools, plugins in BuildServers.

DevOps is the key how to manage Microservice architectures through Buildservers and/or clouds.



Toolset and testing pyramid

CICD

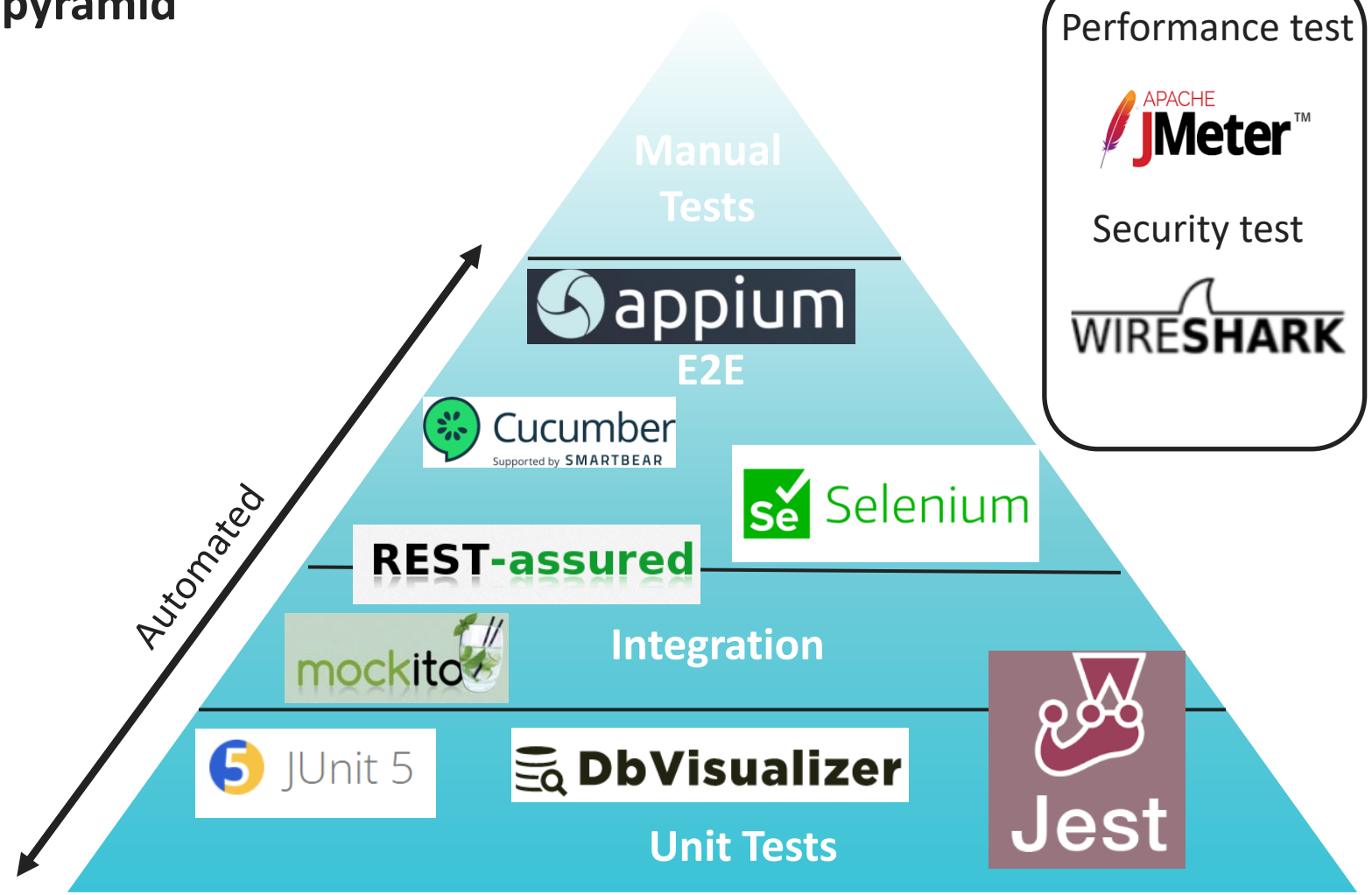
Jenkins

Monitoring

splunk>

Static code analysis

sonarqube



Performance test

APACHE JMeter™

Security test

WIRESHARK

- ❑ Slow
- ❑ Expensive
- ❑ Complex

- ❑ Fast
- ❑ Cheap
- ❑ Simple

Number of tests

Facing reality

New webshop

- Succesful release
- 20 micro services are in place
- Mobile developement in place
- Monthly visitors reaches **2,4 Million** at the first month!


New Mobile App

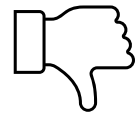
- Webshop releases with various quality
- 30+ micro services
- Mobile App release went to PROD (**2. time only**)
- **5+ Million** visitors on Webshop >> more revenue
- Webshop Prod problems


First Large release

- Poor Mobile App Rating
- Qualtiy and performance problems with Webshop
- Problems with realeasing
- **2 months shift** was need for the next release
- Extra cost to reduce backlog


Second Large release

- New features received NOGO for Prod
- Less than 4 minutes page visit time
- PROD Stability problems
- **1 month to fix PROD issues and finish current release**
- **6 month to finish Phase1**


6 months → 1 year → 1,5 year → 2 year

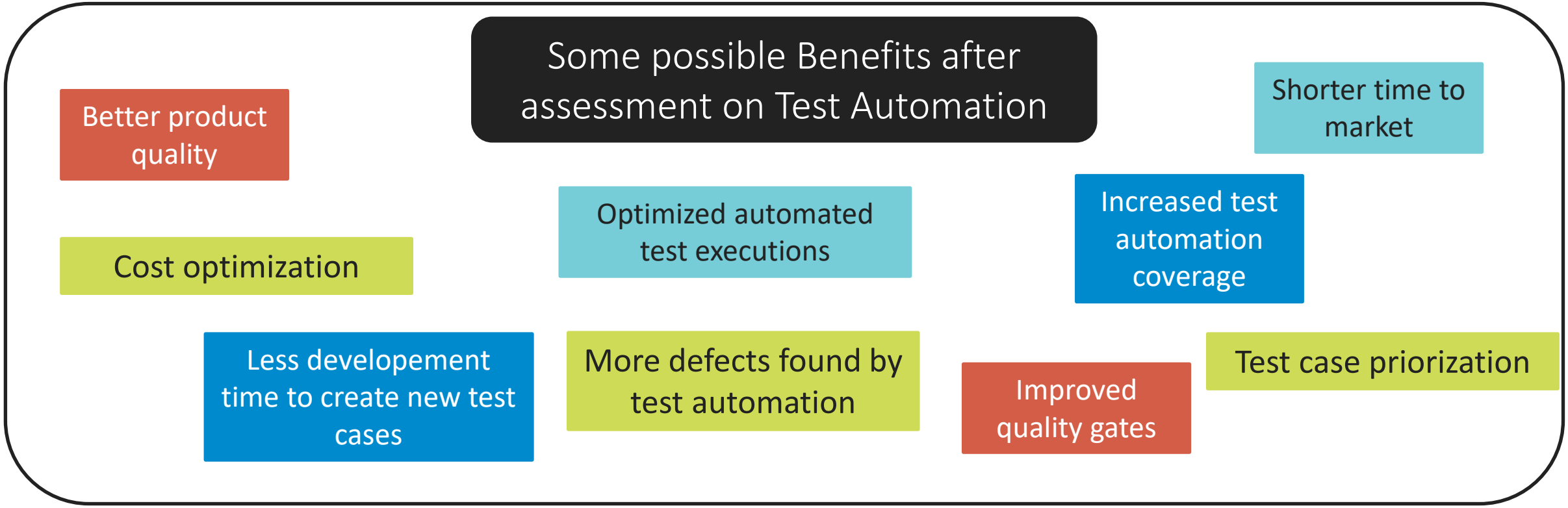
Assesment on Test Automation

Audit teams start to examine how to reach the target

Problem to solve by our imagined assessment team:

Improve the efficiency of test automation.

(There are other assessment teams to improve Testing, Process management, Devlopment processess, Security, etc...)



Improve test executions on environment level

Test Environments					
DEV	FAT	SIT	UAT	Staging	PROD
Local (only devs)	Isoleted Env for component tests	Integrated Env for System integrated tests	Integrated Env for Acceptance tets	Same version as PROD	Production

Current state

Test automation is part of DoD

- ❑ Automated API tests are on FAT, SIT
- ❑ Automated UI tests are performed on UAT
- ❑ Automated Mobile tests are performed on UAT
- ❑ Performance tests are performed on UAT
- ❑ There are no other automated test cases

Improvement

✓ Start automated tests in earlier stages and create smoke/sanity tests

➢ Dev:
Do contract tests
➢ FAT: +Include
API + UI Automated component tests, Performance tests
➢ SIT:
Automated E2E UI Regression sets to be added
➢ UAT, Staging:
Both API/UI/Performance: Smoke test sets, Sanity tests
➢ PROD:
Both API/UI: Smoke test sets, Sanity tests

Speed up test executions

Findings

- **The following automated test cases are in place:**
  - API: 2000+ test cases
    - 3-5 hours run time
  - UI: 500+ test cases
    - 3-6 hours run time
  - Mobile UI: 300+ test cases
    - 2-3 hours run time
- Requirement coverage is up to **95%** in all Automated test cases
- All Test automation integrated into CI
- BDD is in place!

- Tags are partially set but not used.
- Parallel test execution is in place but makes tests more instabil.

Impact

- Smoke test sets/Sanity test sets are not in place
- Automated tests cannot give back an immediate feedback about product quality
- Manual test engineers needs to do smoke/critical regression tests manually
- Tests are executed during nights only

Recommendations

- Include at least each of the following tags for the different test automation types:
  - Regression, Smoke, Sanity

- Analyse maximum threads for parallel execution

- See how to improve stability on the next page →→→

Test Run stability

Current state

Different reasons beyond Flaky tests:

1. Contract changes **~ 30%**
2. Unexpected deployments on Integrated environment **~ 20%**
3. Test data is changed/deleted. **~ 10%**
   (Anyone can use any test data on the system)
4. Slow application can cause timeouts **~ 30%**
5. Overcomplicated/not defnied test cases **~ 10%**

Pact

Improvement

Possible solutions to reduce flaky tests

1. Use „Pact” contract test on DEV level
2. Stabilise release management by setting processess, define roles who and when can do it
3. Improve test data management to have isolated test data for the different test sets and improve test data maintenance. (Eg: Each team maintain their test data)
4. Improve performance engineering → „slide 13”
5. Mute overcomplicated tests untill have time to simplify

Test prioritization

Findings

- Production data is not used in testing
- Most common test data combinations are not identified!

API (RestAssured):

- API coverage is not measured!
- Only **6/20** of most common API combinations are covered!
- There are no test cases for **48%** of all API combinations!

URI	Method	Status code	Appearances
/api/Testing/SampleRequest	GET	200	14596
		401	897
	POST	200	2435

Webshop(Selenium):

- **59%** of the users use Webshop from Mobile browsers.
  (No Automated test for webshop in Mobile Browser)

Mobil App (Appium)

- Mobile Project is in Firebase but crashlytics statistics are missed!

Impact

- Not using production statistics can lead to miss critical scenarios
- Using only requirement coverage metric can show false results
- Complete scenarios can be missed. (Webshop from mobile, see API coverage)
- Smoke tests are not efficient without correct prioirty

Recommendations

- Analyse/use production data regularly

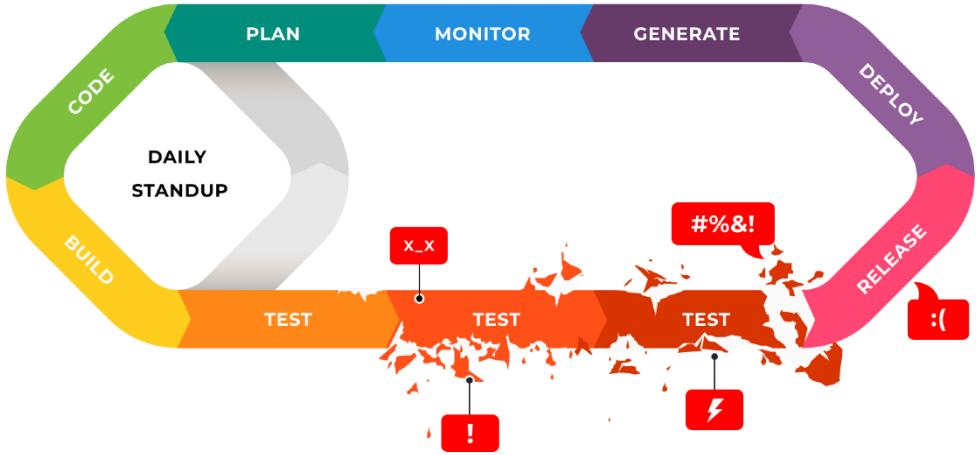
Low-hanging fruit:

- Use better test data
- Increase API coverage with common, missed combinations
- Use crashlytics for mobile automation

Mid-long term goal:

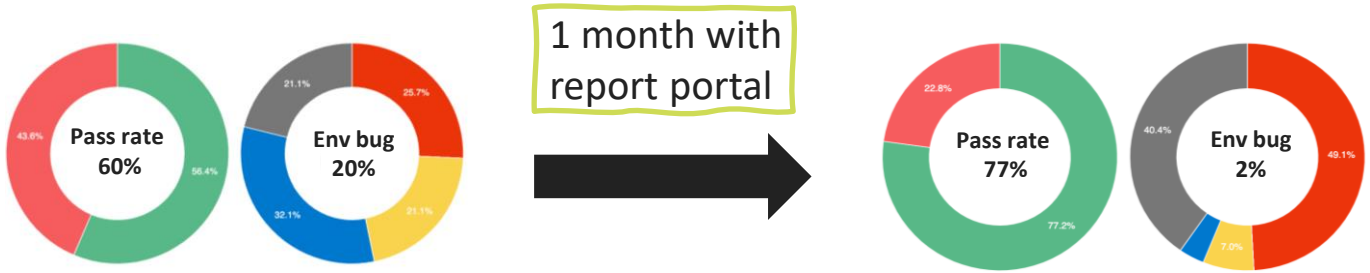
- Start develop Mobile tests for webshop. Untill that more manual testing effort is needed!

Increase test stability, time spent on analysis



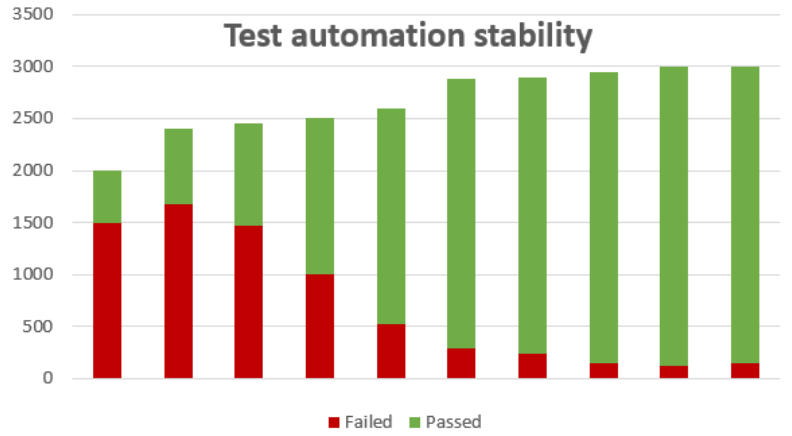
AI Powered Open-Source test automation dashboard

- Manage all your automation results and reports in one place
- Make automation results analysis actionable & collaborative
- Establish fast traceability with defect management
- Accelerate routine results analysis
- Visualize metrics and analytics
- Make smarter decisions together



Key features with benefits:

- ✓ Auto-analysis greatly decrease time spent on result analysis
- ✓ Full test automation health visibiltiy
- ✓ Test execution results are accessible in real time an no need to wait until finish!
- ✓ Ability to submit and open defects directly in JIRA for failed tests



From: https://reportportal.io/

Test automation team runs performance tests

Current State

- Test automation needs to run performance test scenarios
- Non-Functional Requirements are not presented
- Production data is not analysed →
  - Load profile is not identified
- Test vs PROD HW ratio is not identified
- Most common test data and business flows are not identified
- Performance tests are performed on irregular basis only on UAT
- Performance scripts are outdated
- Only backend is measured!
- Logs are not analysed for performance test executions
- Test execution slowness due to poo performance

Start performance engineering

- Identify load profile, HW, most common test data, business flows
- Collect NFR's
- Start clientside measures
- Make it regular and start in earlier stage
- Identify Test vs PROD ratios....

→→→→→→→→→

- Make performance tests as DoD

Process Speed up → getcarrier.io

- Easy to rollout, start and understand
- Report-portal integration

CARRIER

Improved Quality Gates

*Staging environment is not part of the delivery pipeline and used for PROD tickets

DEV

Automated
- ❑ Build
- ❑ Deploy
- ❑ Unit tests
- ❑ Contract tests
- ❑ Static code analysis
- ❑ Static security tests

Manual
- ❑ Peer Review
- ❑ Dev test

FAT

Automated
- ❑ Build
- ❑ Deploy
- ❑ Artifact publish
- ❑ API: New feature + Smoke
- ❑ UI, Mobile: Smoke test
- ❑ Performance Component test

Manual
- ❑ UI, Mobile: New feature test
- ❑ Walkthrough review

SIT

Automated
- ❑ API, UI, Mobile: Regression
- ❑ Integrated Performance tests

Manual
- ❑ UI, Mobile: New feature test
- ❑ Monitoring
- ❑ Build

UAT

Automated
- ❑ API, UI, Mobile: Smoke, sanity
- ❑ Integrated Performance sanity tests

Manual
- ❑ UI, Mobile: Acceptance test
- ❑ Security tests
- ❑ Monitoring
- ❑ Build

*Staging

Automated
- ❑ Health check
- ❑ API,UI, Mobile: Smoke + Sanity
- ❑ Static code analysis

Manual
- ❑ UI, Mobile: Acceptance test
- ❑ Security tests
- ❑ Monitoring
- ❑ Build
- ❑ Walkthrough review

PROD

Automated
- ❑ Health check

Manual
- ❑ UI, Mobile: Acceptance test
- ❑ Security tests
- ❑ Monitoring
- ❑ Build

Current state after 2 years

2 years expectations:

- Monthly visitors: 5+ million
- Avg Visit Duration: 5+ minutes
- Annual Revenue: 50+ M EUR
- Countries where available: 30+
- Shorter time to market
- Phase1: New flexible webshop
- Phase2: New Mobile App
- Better performance / more secured

After 2 years developement

XY Company

- Monthly visitors: 8+ million
- Avg Visit Duration: 4- minutes
- Annual Revenue: 72 M EUR
- Countries where available: 41
- Time to market is better, but not with expected quality
- Phase1: New flexible webshop
- Phase2: New Mobile App
- Worse stability/response times
- More secured (No security issues)

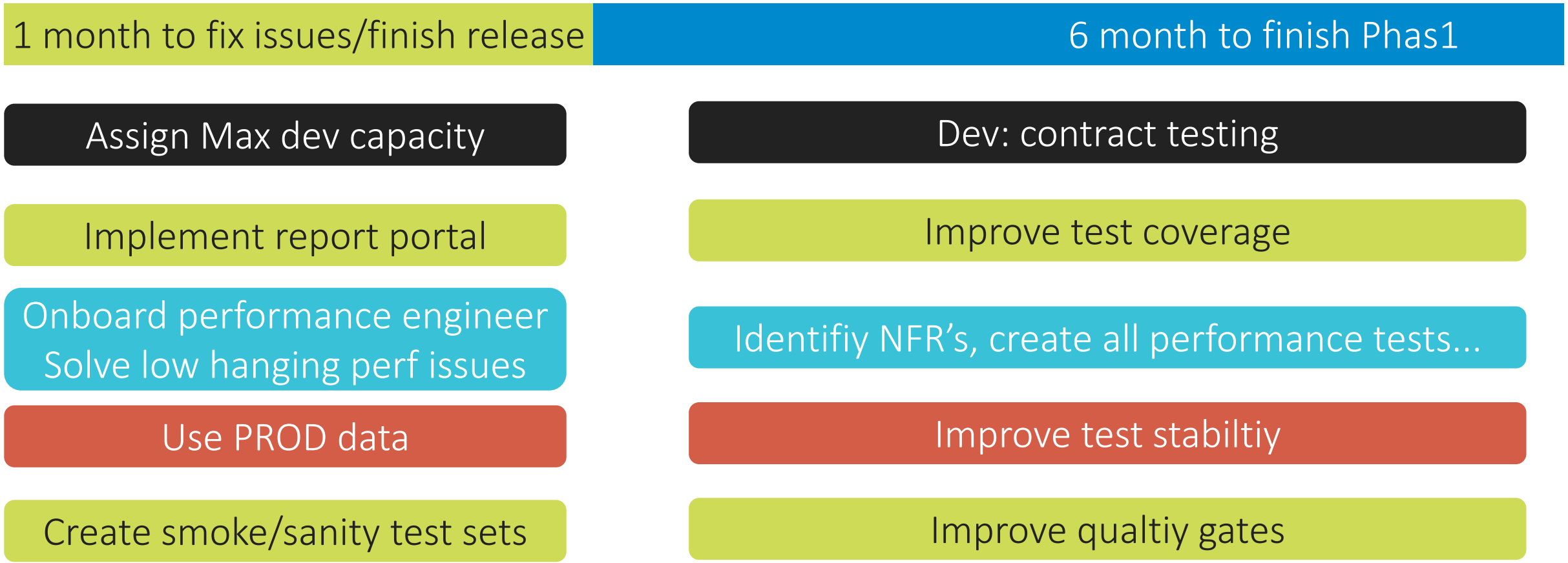
„Selling with pleasure"

Focusing on next 3 years

1 month to fix issues

Roadmap

There is another version from the other assessments. It contains test automation only.



Q&A

Thank you!

For more information contact

Adam Toth
Lead Software Test Automation Engineer

Adam_Toth5@epam.com

Budapest

Bókay János street 44

1083

‹epam›

Appendix

Tools in the testing pyramid:

- Appium: https://appium.io/
- Selenium: https://www.selenium.dev/
- Cucumber: https://cucumber.io/
- RestAssured: https://rest-assured.io/
- Mockito: https://site.mockito.org/
- Junit5: https://junit.org/junit5/
- Jest: https://jestjs.io/
- DbVisualiser: https://jestjs.io/
- Jenkins: https://www.jenkins.io/
- Splunk: https://www.splunk.com/
- SonarQube: https://www.sonarqube.org/
- Jmeter: https://jmeter.apache.org/
- Wireshark: https://www.wireshark.org/
- Report portal: https://reportportal.io/
- Carrier: https://getcarrier.io/#about
- Contract testing: https://docs.pact.io/