OCIO

Test Automation Survey Results

BRITISH COLUMBIA

Ministry of Citizens' Services

Introduction

- An initiative to create guidance for implementing UI Test Automation
- The deliverables will enable new teams to quickly set up, run, and debug when required, automated tests including API, Component, end-to-end, and smoke tests; specifically excluding unit tests.
- We have recently published a survey to get an idea about the technical landscape and current practices.

Survey Objectives

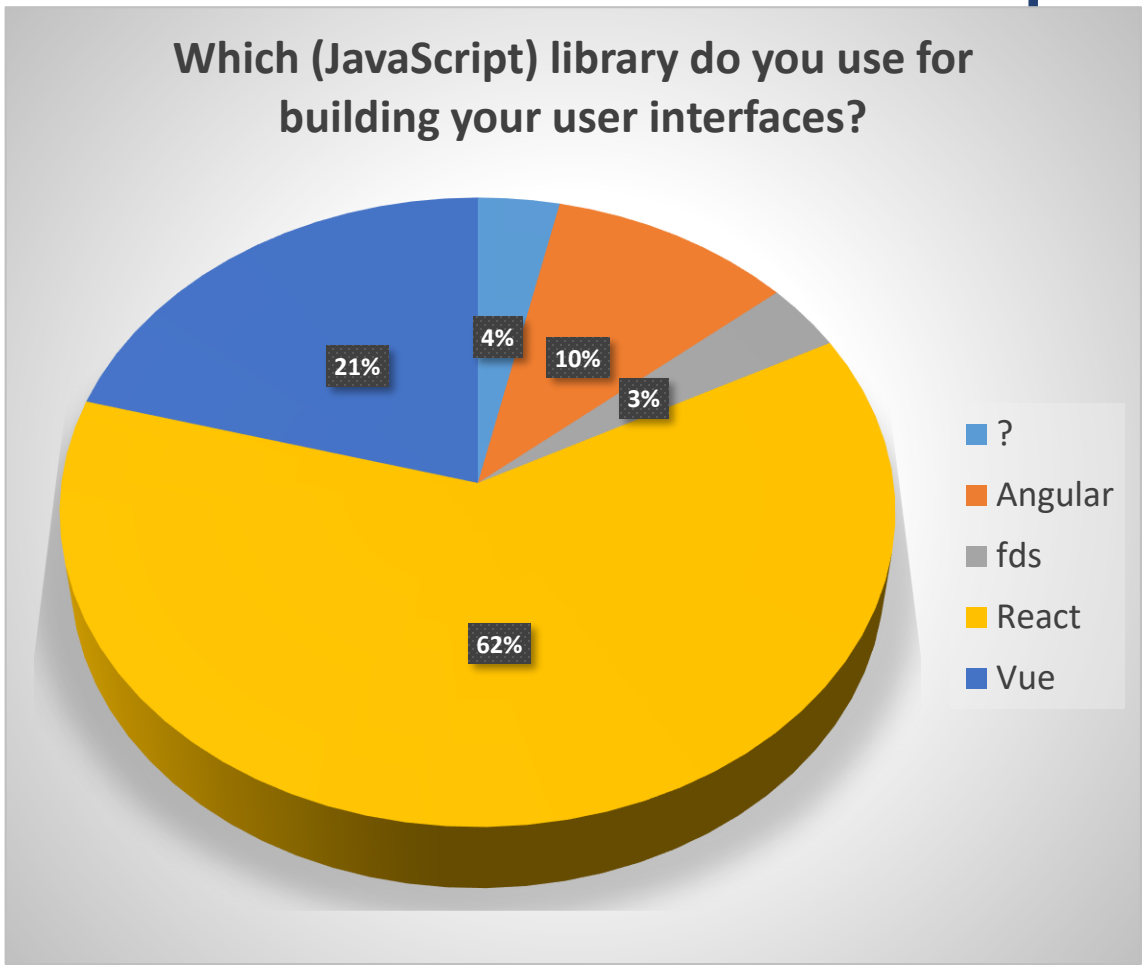
- To better understand:
  - Technical Landscape
  - Current Tool Usage
  - Experiences with Test Automation
  - Ideas, Opinions, Remarks

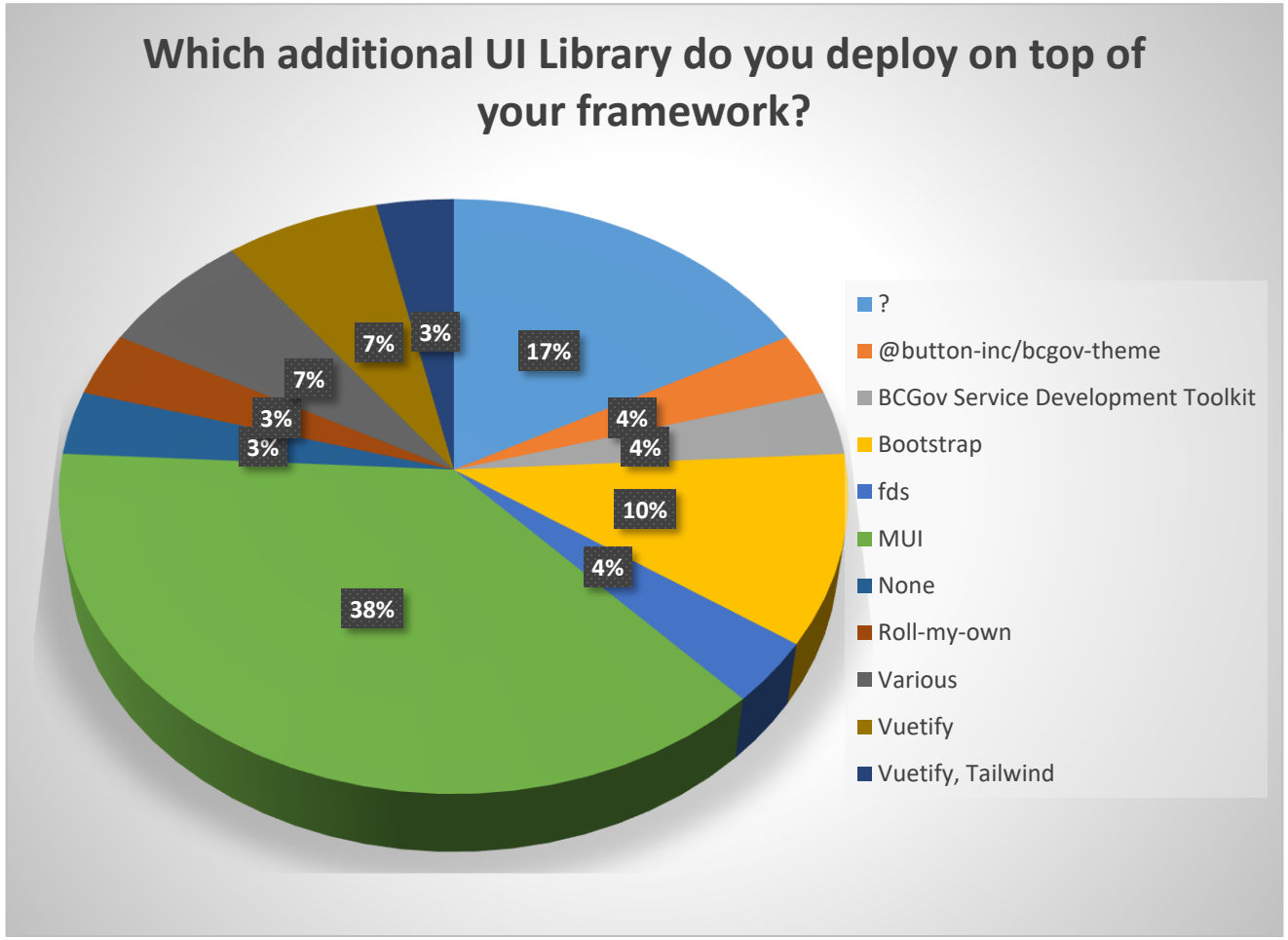
- Confirm our direction for the project
  - Add value to our community
  - Spend our time wisely
  - Address key concerns/challenges

Results

- 28 respondents
- Many offered more elaborate views and insights
- Because of open-ended questions, we had to interpret the data (raw results are available for transparency)
- The results seems to confirm our initial thoughts about the direction that our project needs to take.

Technical Landscape





BCDS CSD DLM GDX OCIO OCIO CIRMO OCIO CONN OCIO ES PSD RPD SBC

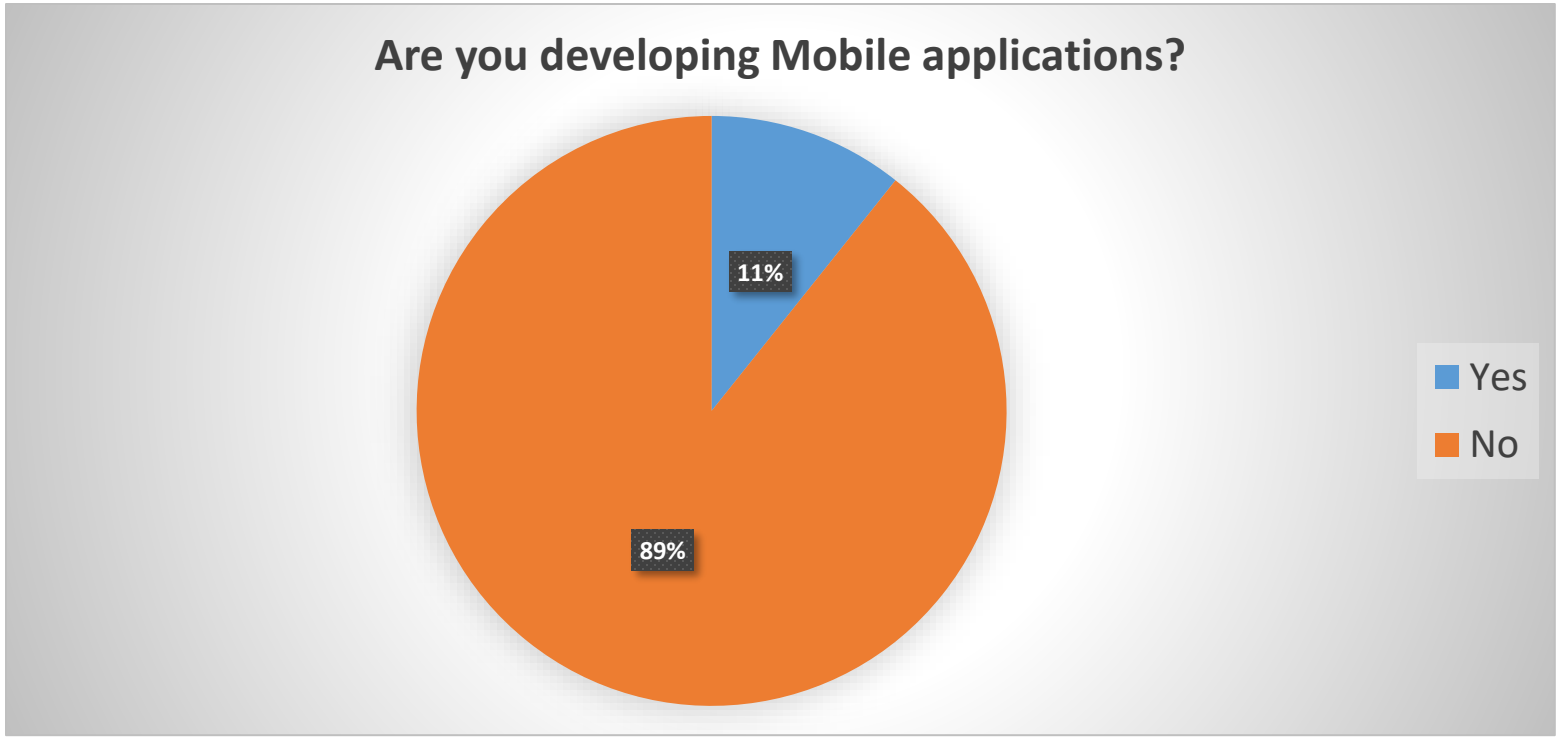
Technical Landscape – Observations / Conclusions

- React is most mentioned
- MUI (Material UI) seems popular, but quite a few others being used
  - Comment:
    - It would be amazing if we (CITZ generally, but the Digital Office/DevEx and GDX in particular) could come up with a single canonical BC Design System implementation that either worked with all three major front-end frameworks or chose React (I assume) as the big one to support, with or without a particular UI library involved.

- Conclusion:
  - We'll focus on Test Automation with React, with perhaps some guidance on popular UI frameworks (time permitting)

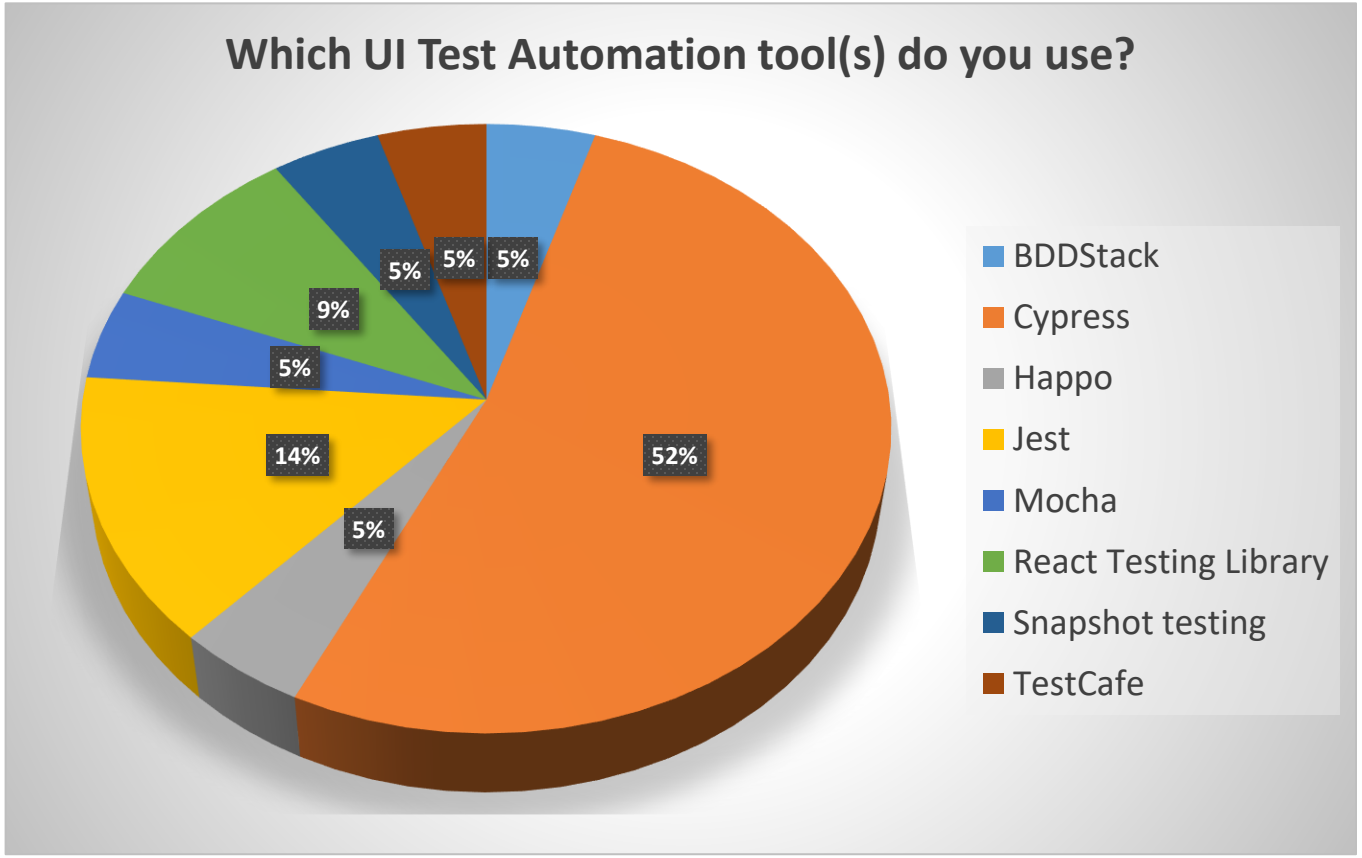
Mobile?

- In our project we will **not** specifically address mobile application testing

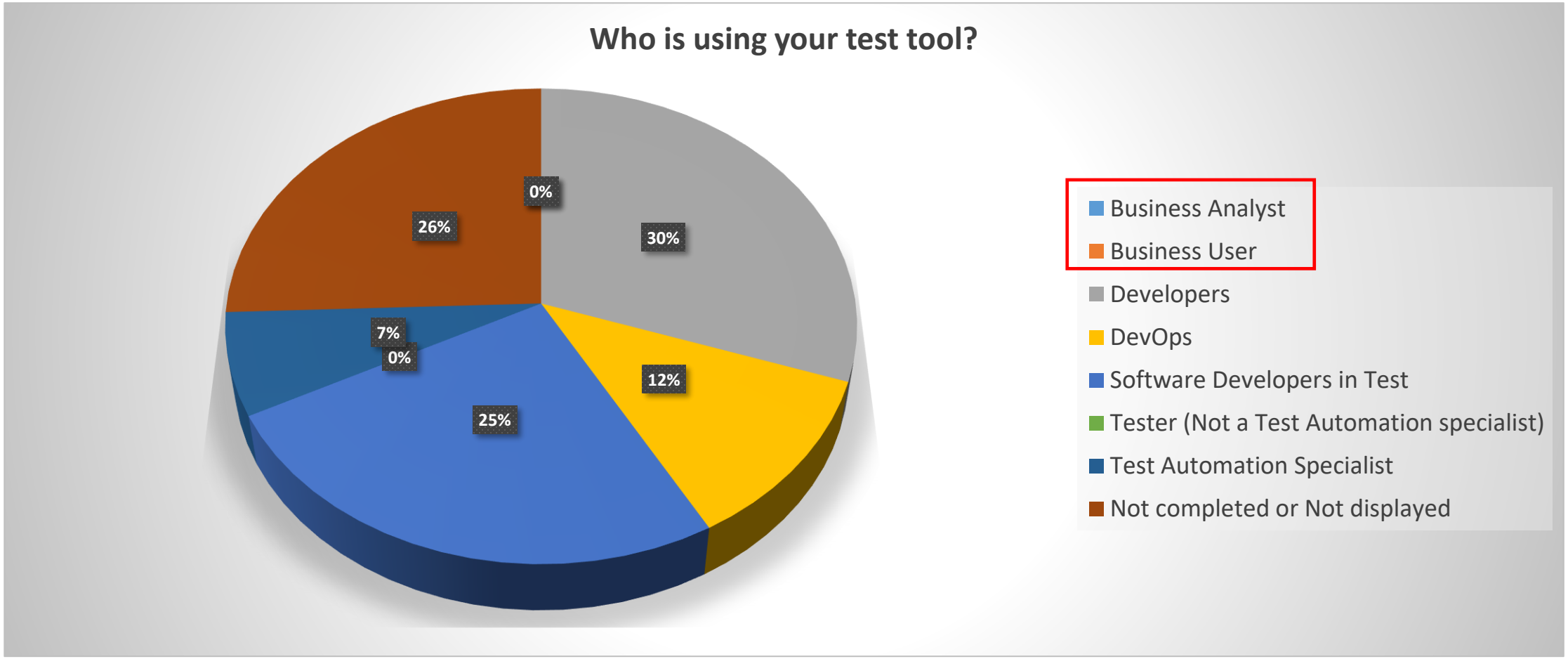


Test Tools

- We'll focus on Cypress



Usage

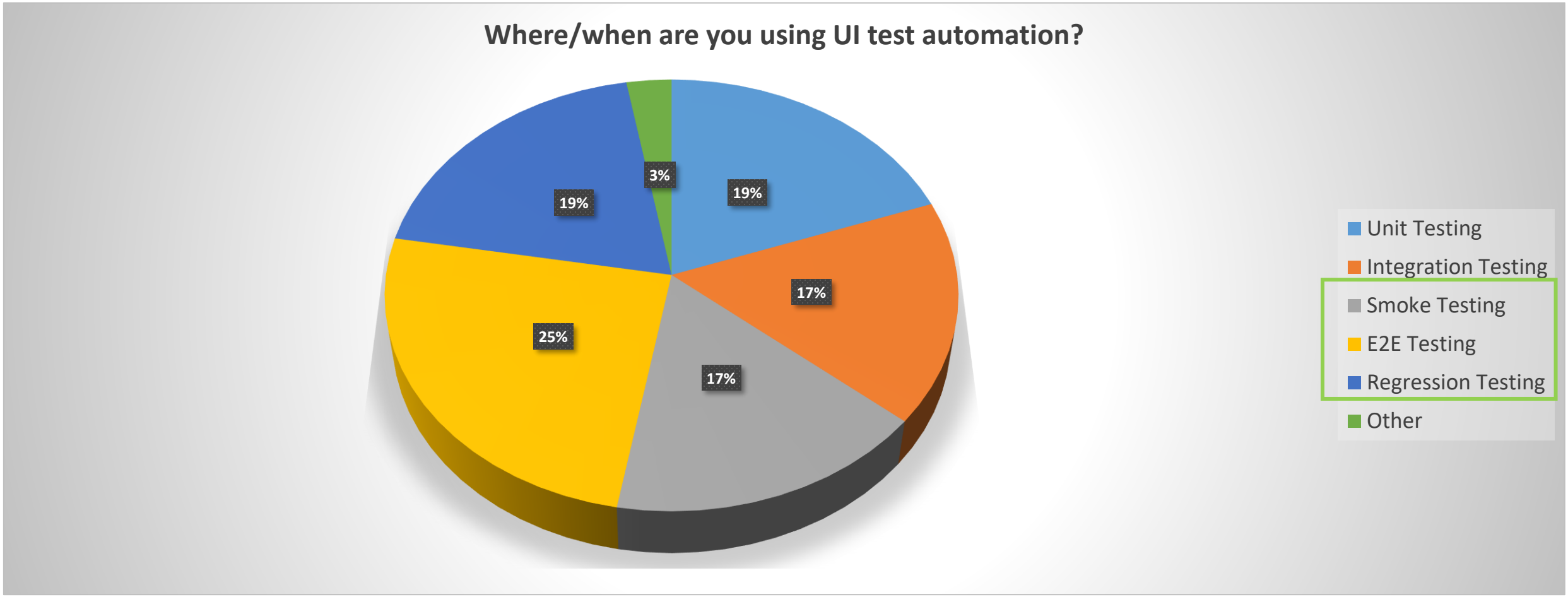

Who is using your test tool?
0%
26%
30%
7%
0%
12%
25%
Business Analyst
Business User
Developers
DevOps
Software Developers in Test
Tester (Not a Test Automation specialist)
Test Automation Specialist
Not completed or Not displayed

Usage: Observation / Conclusion

- Test tools are used by the technical team members
- Developers seem to have added tasks

- Possible conclusion:
  - Test Automation requires additional support
  - A known anti-pattern is to pile TA on the developers plate

When is UI Test Automation used

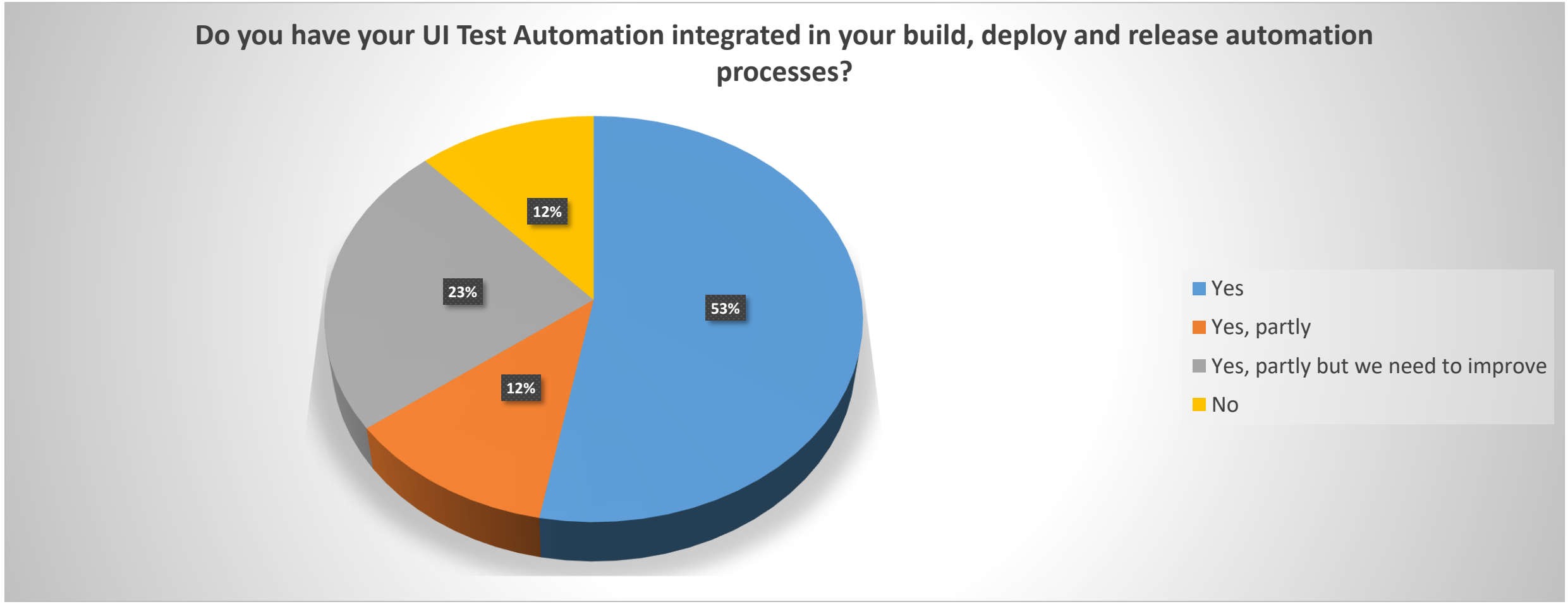


When is UI Test Automation used: Conclusion

- It seems that the majority is focussing on:
  - E2E
  - Smoke
  - Regression

- These are indeed typically the key target areas for UI Test Automation

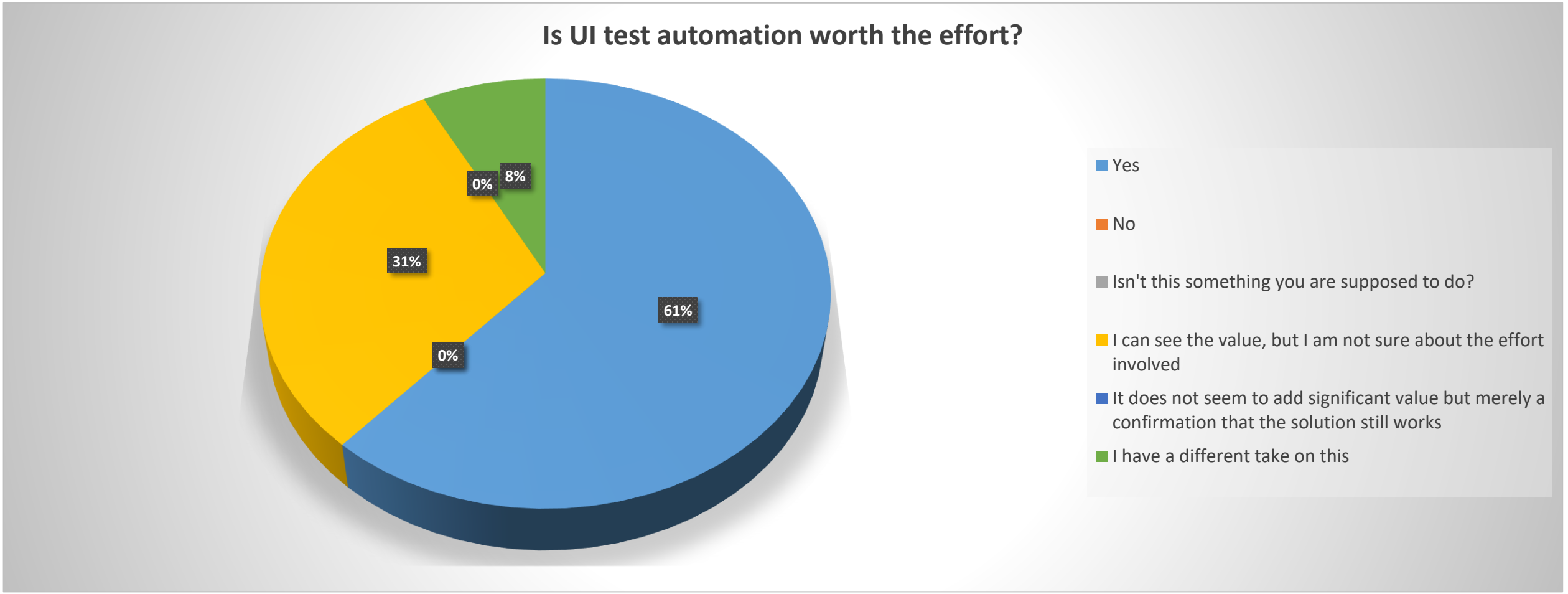
Integration with CI/CD



Integration with CI/CD: Conclusions

- Most of you have automated testing integrated in your CI/CD processes.
- Reading between the lines though: Most of this is unit test or really simple functional tests.
- A significant amount indicated that they are aware of the fact that more can be done. This is also reflected in the answers to the question about what are people planning to do with UI Test Automation. (Improve and expand)

Worth it?


Is UI test automation worth the effort?
61%
0%
31%
0%
8%
Yes
No
Isn't this something you are supposed to do?
I can see the value, but I am not sure about the effort involved
It does not seem to add significant value but merely a confirmation that the solution still works
I have a different take on this

Comments – A selection

- Time and budget don't allow for UI test automation for most projects.
- Our tests work, but there is high barrier to entry for understanding how to write good tests. Most tests end up being basically useless unit tests. Higher value tests are harder to write, so they don't get written. There is no test specialist, so it's up to each full stack dev to try and get good at this over time. I appreciate our tests but don’t put much faith in them.
- Many Government internal apps are quite simple and become stable and unchanging for many years, in this case a heavy test automation effort may provide less value.
- I think there's no one size fits all for how much test automation should be required.
- UI tests are overall worth having, but it seems difficult to balance the value vs the cost, as they are somewhat more brittle than a unit test for example.
- A few good tests go a long way. Writing too many tests? Chasing coverage is a fools errand.
- Worth it, but expensive in time.
- …and many more

Raw Data

- The raw results can be found here: https://github.com/bcgov/automated-testing/blob/main/survey/LimeSurvey.mhtml

What does this mean for us…

- Focus on:
  - **Tool:** Cypress
  - **Test Focus:** E2E UI Test automation of the complete application
  - **Framework:** React
  - **UI Libraries:** MUI, Bootstrap
  - Tool and Methodological Guidance for effective and efficient implementation
  - Recipes for:
    - GitHub Actions/OpenShift Pipelines, local deployment
    - Keycloak, Cucumber, Test Data use, external libraries
    - Setup, Configuration, parametrized running
    - Etc.

Thanks

- For your participation, thoughts and comments.