##################################
INTRODUCTION
##################################

Difficulty - easy, business as usual
Time - 2 business days, but 60% of time was producing and refactoring this document, not generating the actual business value

##################################
CONCISENESS AND INFORMATION ARCHITECTURE
##################################

I want to prove that I am one of the best in my job but I also have no illusions that you will read the whole of this massive document. If you need to skip, please prioritize content at the top and around titles, subtitles, bold words

##################################
WHAT TO AUTOMATE - Not Everything
##################################

Summary - writing tests is easy, but then they will fail. Processing large volumes of noisy test results tends to massively burn company money. So we must be practical.

Details
In this task we have 6 topics, each with their own fields, around 80 fields in total. It is easy to come up with 5-10 test cases for each, resulting in up to 800 test cases. By reusing code, it is also easy to automate them. The purpose of tests is to fail and when that happens, it is realistic to have 120 failures for a single bug for these 800 test cases. This is noise. This noise wastes company money. Our job is not to waste money.

Test management complexity is also a factor.

##################################
WHAT TO AUTOMATE - Principles
##################################

In this section I list some general principles, so you do not have to read carefully the text for all test cases

- Return of Investment - if something is easy to automate, it may be an easy win
- We should not automate third party functionality, but we should at least check (manually) that we integrated it properly

- If the critical business value is already covered in another test, maybe a dedicated test is not needed. In this way we may end up with tests with multiple responsibilities. Such tests add management complexity, so balance is needed.
- Return of Investment - strategies for step and code reuse - flexible and reusable automation apis, parametrization, tables, example structures - all of these things make writing tests easy, but also make them easier to manage, which is absolutely critical
- Take project history into account - if an existing functionality has proven with time that it is stable, it is ok to disable (not necessarily delete) lower priority automation
- Take project history into account - if an existing functionality has proven to be fragile, maybe we should increase coverage
- Risk aversiveness - I personally tend to be skeptical of unit test coverage, so I tend to add lower priority tests and then disable them if time proves that they are not critical. This is specific to my experience.
- Generally our automation is alive. We do not just build and forget. Instead we build it to be flexible, because we need to extend and change it all the time
- Return of investment - some tests are unreasonably expensive (time, disk space, management complexity) for the business value they add

##################################
WHAT TO AUTOMATE - What about manual testing
##################################

It is nice to have at least a single exhaustive manual test iteration on a new functionality. As a QA, I assume this exam tests me if I can do that as well. I will not list 800 test cases, because no one will read them, but at the bottom of this document I add a concise section on manual form validation

In addition, a perfectly acceptable optimization would be to test extensively a single field of a given type, and not repeat all tests on all instances of that field type. Of course, I generally make decisions like this based on available business needs and resources.

##################################
BDD - In files
##################################

Please find my BDD output not here, but in accompanying multiple feature files

##################################
BDD - Code Comments
##################################

The task description asks me to explain my automation decisions. A lot of that is in the BDD files in the form of code comments. This is not just for this exam, but how I actually do my work.

Comments should explain less about how the code works and more about why it is designed that way

Comments should also be concise

##################################
TEST CASE
Content, design, styling
##################################

This test focuses on page 1 as it is just FAQ content, but the design of the submit form is also under test

- Select each of the topics
- Expand each of the questions
- On hover, validate both Topics and Questions change design as expected
- Read each block of text, validate its contents and style are as expected by design
- Check the correct text snippets are actual links and those links lead to correct destinations
- Validate I can expand and contract FAQ questions independently. For example - expand 5 items and contract 2 in the middle. Validate I cannot break the UI with such interactions
- Proceed to next page for all of the topics and validate their designs as well

Automation Priority - Medium, utilize screenshots

Automation Explanation:

Automation this scenario is less a matter of IF and more a matter of HOW. As a rule of thumb, test automation that duplicates text content is a bad practice, as the maintenance cost is unreasonable. As a rule of thumb test automation that validates styles is too fragile and also considered a bad practice.

I would generally avoid UI automation that checks for a lot of text and CSS rules

Fortunately for us, there is a practice that is cheap and scalable. We can have a pipeline which validates screenshots of all content, not just for this specific page. If the framework detects a change in the product, screenshots require manual human effort to be validated and updated, but this effort is minimal, compared to the effort required for content duplication and / or css validation in automation

The screenshot functionality of Playwright is serviceable, it can do the job, but has shortcomings. There are many tools on the market that can automate this work with minimal effort

I would not automate hovering over items here, because the business value here is minimal

I would not automate a complex scenario of expanding and contracting items, because I do not expect his functionality to change and thus I do not expect it to break. The business value here is substantial, because it is still possible a regression to make critical content inaccessible BUT this will already be covered by the screenshots tests that validate content. So an automated test for expanding and contracting items adds minimal to no business value.

Automation Syntax - please see the accompanying feature files

DISCOVERED RELATED ISSUES

- some blocks of text are formatted inconsistently (narrow centered text, looks like a poem)
- links are not highlighted as links, so the user is not informed they are interactive
- some links are not interactive, that is they are text and not links
- I noticed a punctuation and grammar mistake
- all embedded video in “Maintenance and Usage” are misaligned - expected alignment is center, actual alignment is right

##################################
TEST CASE
Error case - no chosen topic
##################################

- open page 1
- click Next without choosing a topic and see a reasonable error message
- choose a tepic and expand some questions, observe for unexpected side effects

Automation Priority - None

Automation Explanation

I expect minimal business impact, if this error case regresses and is not caught

Automation Syntax - None

DISCOVERED 1 RELATED ISSUE

- I open the url
- I click Next without choosing a Topic
- I read a reasonable error message
- I choose a topic
- Expected behavior - the FAQ content is presented to me and the error message from the previous error case is hidden
- Actual behavior - the error message stays on screen and is not hidden

##################################

TEST CASE
Submit Forms - happy path
##################################
- Open page 1
- Choose a topic (test for all)
- Click Next
- Fill in happy path test data
- Click Submit
- Confirm submit is successful

Automation Priority - Critical, but not all

Automation Explanation
- at the bottom of this document you will find a dedicated sections on how I will do a manual test on all field types, this is where the manual test data for this case lives
- in the accompanying feature files, you will see what I will automate

Automation Syntax - please see the accompanying feature files

##################################
TEST CASE
Submit Forms - error cases
##################################
- Open page 1
- Choose a topic (test for all)
- Click Next
- Fill in error case test data
- Click Submit
- Confirm expected and actual errors match

Automation Priority - Critical, but not all

Automation Explanation
- at the bottom of this document you will find a dedicated sections on how I will do a manual test on all field types, this is where the manual test data for this case lives
- in the accompanying feature files, you will see what I will automate

Automation Syntax - please see the accompanying feature files

DISCOVERED RELATED ISSUE
- phone field has no validation
- first field for every topic has an error in small letters already displayed even without submitting anything
- generally very few validation rules, which is probably fine for the business needs

- when asking for a full address, postal code is required for one topic, but is completely missing for all others. May not be a bug, but should check with Product Owner

##################################
TEST CASE
Navigation between the 2 pages
##################################

- Open base page
- Explore the page with both happy path and error case interactions
- Click Next
- Explore the page with both happy path and error case interactions, without finalizing the process
- Click Previous
- Repeat and observe for unexpected side effects

Automation Priority - None

This test is just a part of our exploratory testing and I write it down so I have a proper place to put the discovered related issues. Not a candidate for automation.

Automation Syntax - None

DISCOVERED RELATED ISSUE

- After pressing “Previous” button, styling on hover over questions is now unexpectedly blue

##################################
TEST CASE
Responsiveness
##################################

- Execute a happy path scenario under various screen resolutions, set of resolutions must be pre-agreed with Product Owner.

Automation Priority - Medium, utilize parametrization

Automation Explanation

I would not say automating priority here is high, but most frameworks will provide a trivial way to run multiple tests under multiple conditions, read from a config - like language, resolution, base url, etc. It would be an easy win to setup a small batch of happy path scenarios to run under a small set of different resolutions. If we do this, we should be selective and not multiply our screenshots by a lot, this is a very easy path to maintenance hell.

Automation Syntax - None

We should not have a dedicated test for this, this should be a parameter of other tests

##################################
TEST CASE
Different browsers
##################################
Rationality for this test case is the same as above

##################################
TEST CASE
Surrounding UI interactions
##################################

I assume the test scope is the Customer Support functionality and not the surrounding UI (navigation, social, footer, legal). As part of my exploratory testing, I would do minimal interaction with them. The test exists, because I want to confirm the devs reused working functionality and did not instead copy-pasted code and broke stuff. On the other hand, the test will be minimal, because I assume these interactions are not an immediate priority and may even be completely out of scope

Automation Priority - None

Automation Explanation:

I would not automate this, unless I see that the devs indeed did not reuse, but instead copy-pasted code and broke functionality. If that is the case, automation should be almost free, as it is trivial to adapt an existing test cases to test headers, social links, footers on a list of URL-s, where the URL is a parameter of the same test

Automation Syntax - None

##################################
TEST CASE
Abnormal file upload
##################################
Try to upload abnormal files - empty files, corrupted files, executables, large files (15.1 Mb, 100 Mb, 1 Gb), multiple files. Can we do a DDOS attack this way ? I have obviously not done this, because I have no access to your server and cannot repair it if I do actual damage (which I probably cannot, but still …). My point is upload api-s require attention.

Automation Priority - None

Automation Explanation:

It is vital to check for scenarios with security implications, but automating them is a very tempting road to hell.

Automation Syntax - None

##################################
TEST CASE
Captcha
##################################
Captcha-s have a special place in testing, hence this section. It is a third party software, so we should not waste business resources on testing it. But we should still check if it is integrated correctly, hence the discovered issue item below.

Automation Priority - None

Automation Explanation:

The good practice is for captchas to be disabled on environments under test automation. If we decide to automate capthas, especially if it is an AI based captchas, which tracks mouse movement, we could very easily throw our automation budget in the trash and light it on fire.

Automation Syntax - None

DISCOVERED RELATED ISSUE

- When Captcha asks us to click pictures, the pictures are rendered in a very small frame and are not visible. This produces a terrible user experience.

##################################
MANUAL FORM VALIDATION - Checkboxes, Radio buttons
##################################

- In our app under test, checkboxes and radio buttons have a small number of possible values, so I will just test them all
- I will also check what happens if I use the default value. In our case (but not always) this means the value is unset. In some cases “unset” is not the same as “false”

##################################
MANUAL FORM VALIDATION - Dropdowns
##################################

- empty / unset
- default value (may be different than empty / unset)
- a popular value
- an more exotic value

##################################
MANUAL FORM VALIDATION - Text fields
##################################

- empty
- default value (may not be the same as empty)
- regular happy path text
- very long text, just below character limit
- very long text, above character limit
- special language characters
- text with common and more obscure special characters
- text with new lines
- text with spaces and tabs in beginning, end, middle

##################################
MANUAL FORM VALIDATION - Text Areas
##################################
Same as text field, but pay more attention to new lines

##################################
MANUAL FORM VALIDATION - Phone (has format)
##################################
Our phone field does not have format validation and is instead treated like a text field. Sync with the product owner.

##################################
MANUAL FORM VALIDATION - Number (no characters)
##################################
We do not have number fields, all are treated like text. Sync with the product owner.

##################################
MANUAL FORM VALIDATION - Date
##################################
Dates are complex so we usually use a third party software, which means there is no point in testing it extensively. So I will just go with the following values:

- empty
- invalid string
- valid date

##################################
MANUAL FORM VALIDATION - Email
##################################
Email verification is less complex than dates, but we still often use some lib. So I will go with the following:

Email

- empty
- malformed
- valid

Email verification

- empty
- correct match
- incorrect match

If devs do not use a lib, but instead just a regex, maybe we should do more tests than the above

##################################
BDD - Shortcomings
##################################

Problem

I have substantial experience with BDD. BDD has value but also has a notorious reputation for incentivising code duplication and being strongly resistant to refactoring.

As a result BDD tends to produce massive volumes of text and duplicated code. This leads to automation bugs which waste company money BUT also a more subtle problem. As we mentioned above, no-one reads walls of text. This applies to BDD and code. In a project with a massive volume of duplication, people completely lose visibility of what and how is being tested, which is the opposite of the problems BDD tries to address.

Solution

I try to design steps that are very flexible and reusable. They are more like a programming API and less like natural language. I also heavily use parametrization, tables (with optional columns), example structures, designing with the ability to change on the fly what is being tested by just commenting / uncommenting a single line, etc

##################################
BDD - Note on Libs
##################################

I have experience with the “behave” library and it is not ideal. I do not have experience with pytest BDD, but I have heard positive feedback

##################################
SIDE NOTE - Certificates
##################################

Navigating to page returns net::ERR_CERT_AUTHORITY_INVALID

This usually happens when DevOps sets up and / or updates CA and is not related to the functionality under test

The issue resolved shortly before the task being completed

##################################
WHAT TO AUTOMATE - Note on manual regression testing
##################################

Often when we agree to not automate something, some business owners expect us to still do a regression test on it manually. My general principle is that if we do not automate something, when we actively make an implicit call of judgment it is not part of our regression suite. Exceptions exist but are rare.

##################################
SIDE NOTE - GPT and LLM
##################################

I assume you raised this topic, because candidates tend to offload their tests to AI.

In my experience this tech is faster than a search engine for questions with trivial complexity, but for anything else AI is either counterproductive or harmful.

I try to have opinions based on data and tests, not emotion, so I designed and performed actual tests against popular AI services to prove if and how you can accidentally (or even intentionally) harm a business with them. If the topic is interesting to you, feel free to discuss further.