
expoQA®25
MADRID
May 20th,
21st & 22nd
2025
sogeti
expoQA
expoqa.eu

# Rethink and rebuild your testset; showcase your professionalism

Suzanne Kraaij

Expo:QA 2025

# About me

## Suzanne Kraaij

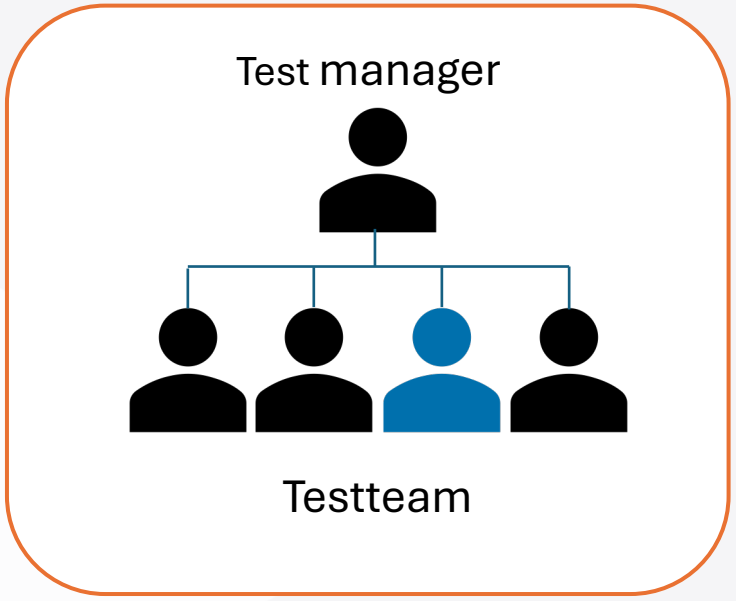
- Agile Test Expert
- 14 years experience as a tester
- Guild Leader: BI & Data Testing Guild
- Member of the core team for the testing community

https://www.linkedin.com/in/suzanne-kraaij/

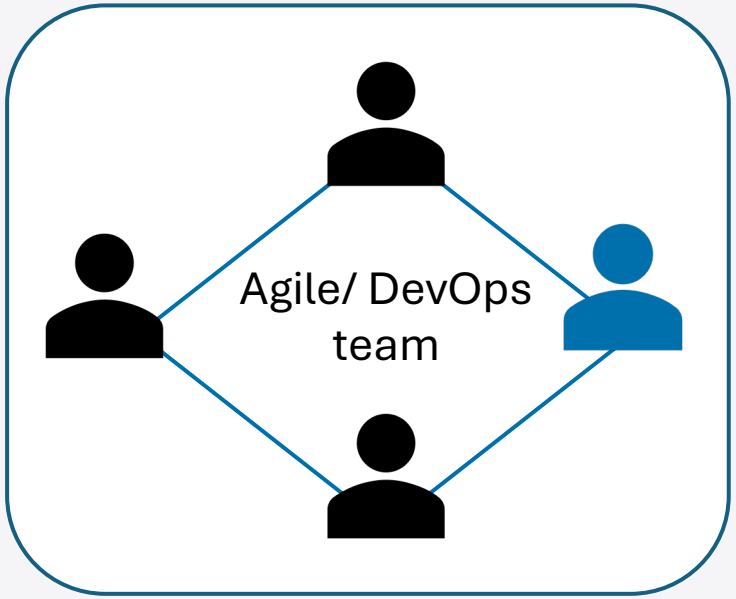
# The Why – First challenge

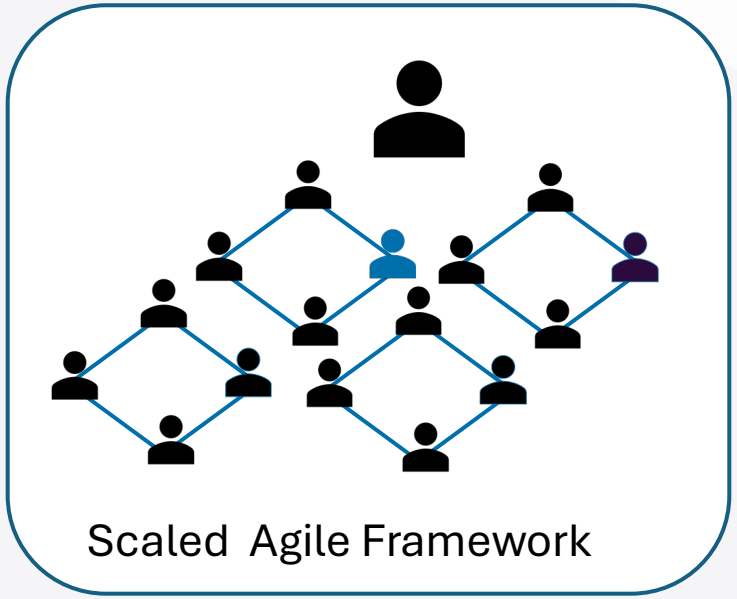
## The Times They Are a-Changing…

From….



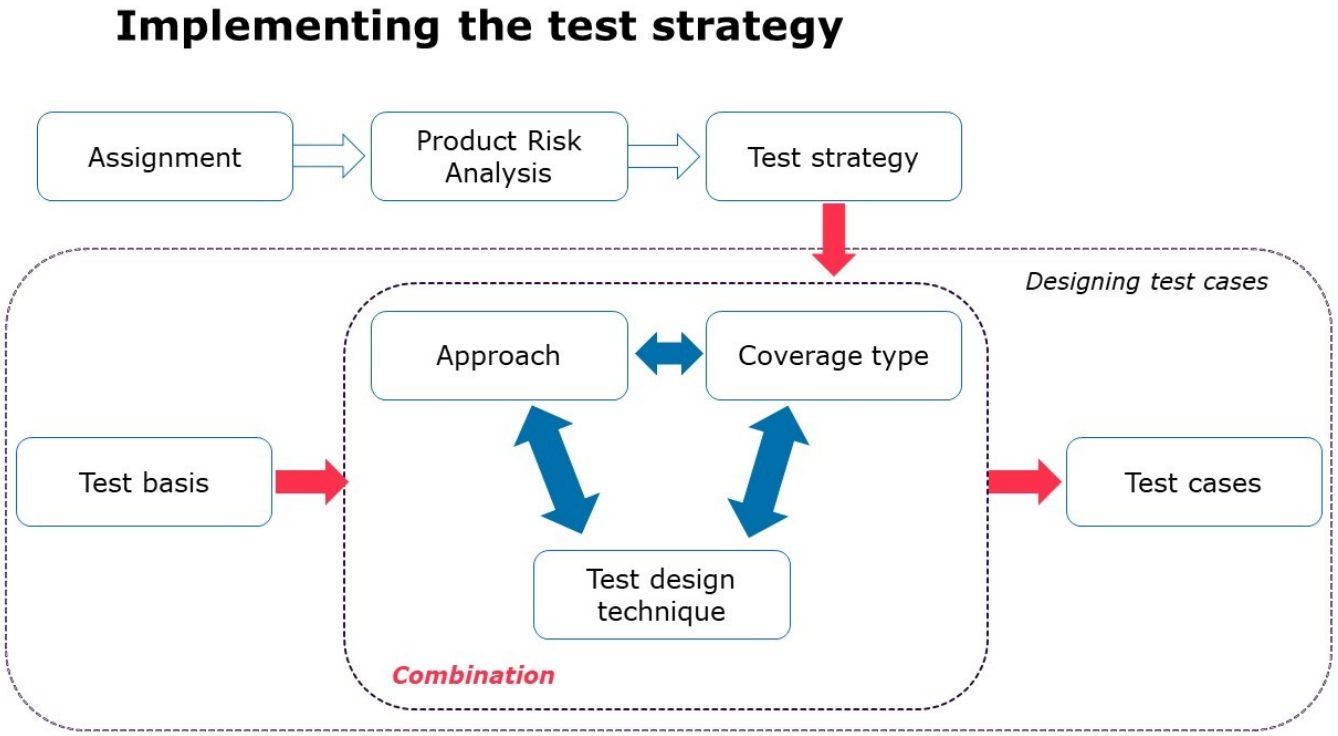

….To



# The why – Second challenge

There already are multiple test strategy methods out there, so why create another approach? For example from TMAP:




...the strategy always comes before creating testcases...

# The gold standard versus reality


## The world isn’t perfect!

You don’t always start testing from scratch, what if you inherit:

- An existing regressiontest
- A set of testcases attached to userstories

➔What is the value of those testcases?

# 1. Starting Questions

What got this journey started?
Guiding principles

# What is the value of your testset?

## Risk

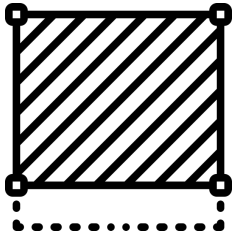
Good test sets cover risk and help you make decisions about risks

## Balance

Good test sets focus on high risk and prevent spending too much effort on low risk

## Enough

Good test sets reflect that you can never test everything, but you are conscious about what you DO test



# Other practical questions

- When do you delete a testcase?
- When you have X testcases to execute/automate, which one is next?
- Which tests go into the regressionset?
- How do you determine how much testing is needed for new userstories?

# How to approach these questions?

**Risk**

Helps you focus, but they can change over time.

**Effort**

Testcases cost time not only to create, but also to EXECUTE and MAINTAIN.

**Criteria**

Explicit criteria can help to make more effective choices more easily.

**Change**

A good test set is flexible, it can change every release or sprint if needed.

# Combined

**Risk & criteria**

Knowing the risks and having clear criteria help find the tests that add the most value.

**Effort & change**

A good testset and regressiontest is up-to-date and can be executed in a specified timeframe.

# Testing takes time

so we cannot test everything

This method helps you focus your effort

# Distribution of effort

is the goal of any testing strategy.

This method works for any type of testing:
manual or automated

# Testing is continuous

Testing is needed as long as there are changes to the System Under Test

This method is repeatable,
and can look both behind and ahead

# 2. The method Explained

This is why you are all here

# There are four simple steps


## Overview

Define scope and level of detail


## Scoring

Add risk and priority to the overview

Acronym: **OSRS**


## Strategy

Define and apply strategy for any changes to the overview


## Reflect

Reflect on the existing testset

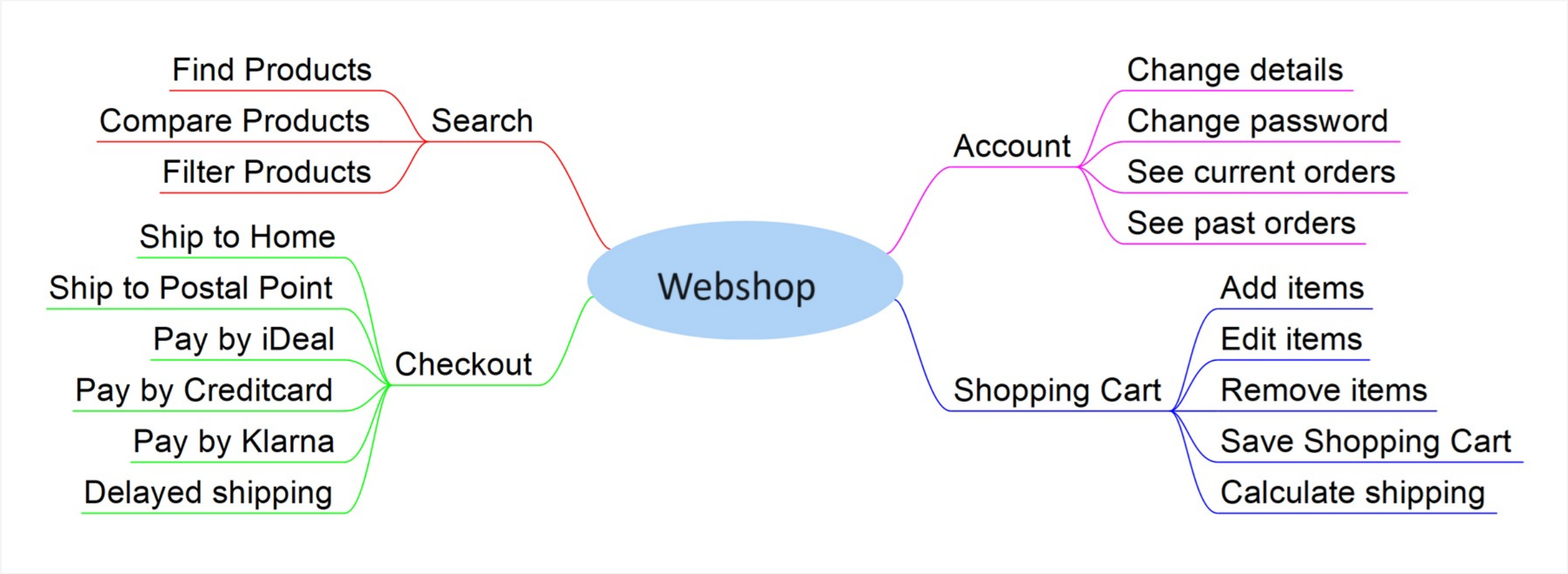
# Overview

- You cannot talk about coverage unless you know the whole.
- Overview is about mapping out the whole.

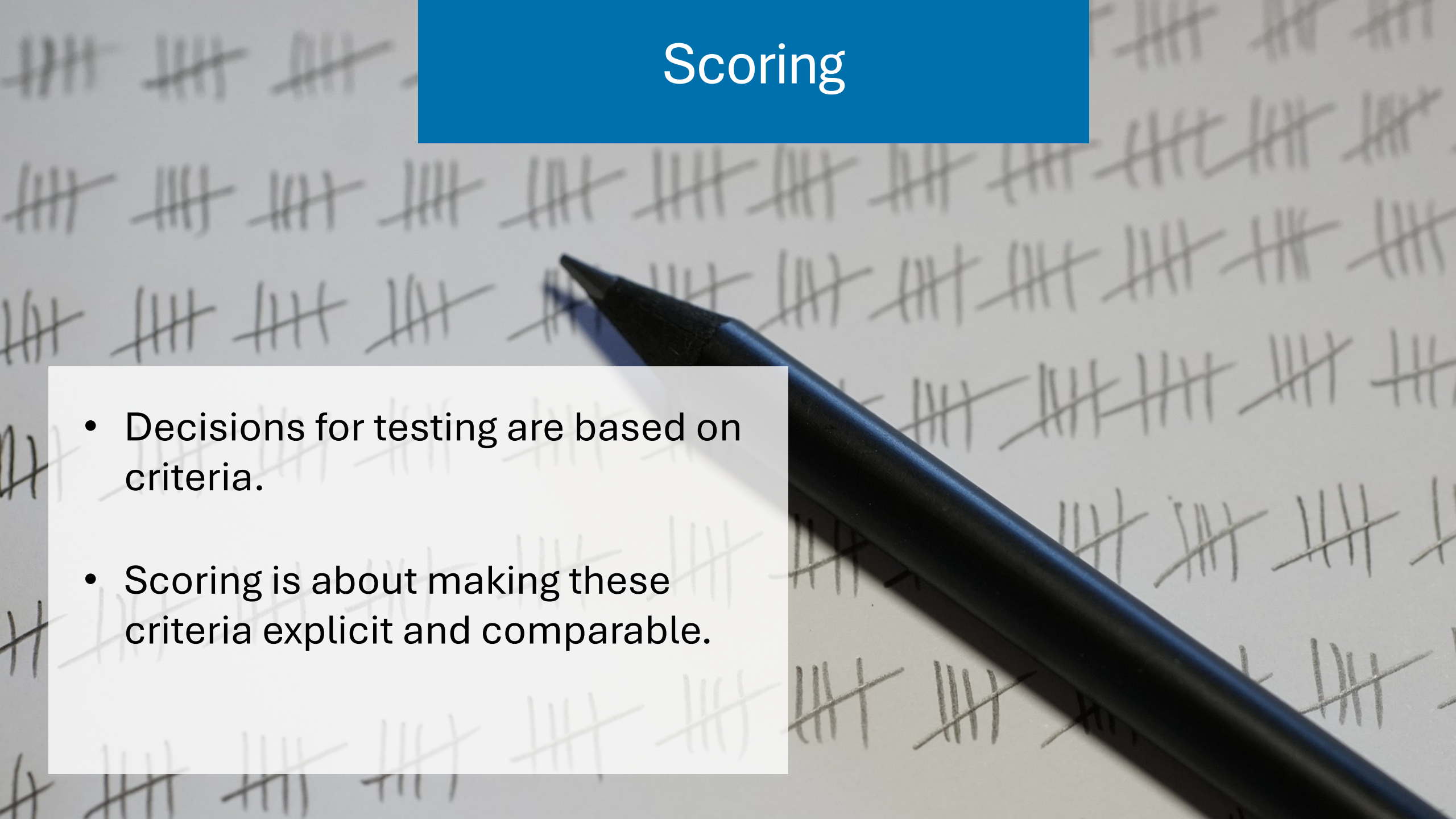
# Example of an overview

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Webshop | | | | | | | | | |
| 2 | | **Overview** | | | | | | | | |
| 3 | **Category** | **Functionality** | | | | | | | | |
| 4 | *Search* | | | | | | | | | |
| 5 | | Find Products | | | | | | | | |
| 6 | | Compare Products | | | | | | | | |
| 7 | | Filter Products | | | | | | | | |
| 8 | *Shopping Cart* | | | | | | | | | |
| 9 | | Add Items | | | | | | | | |
| 10 | | Edit Items | | | | | | | | |
| 11 | | Remove Items | | | | | | | | |
| 12 | | Save Shopping Cart | | | | | | | | |
| 13 | | Calculate Shipping | | | | | | | | |
| 14 | *Checkout* | | | | | | | | | |
| 15 | | Ship to Home | | | | | | | | |
| 16 | | Ship to Postal Point | | | | | | | | |
| 17 | | Pay by iDeal | | | | | | | | |
| 18 | | Pay by Creditcard | | | | | | | | |
| 19 | | Pay by Klarna | | | | | | | | |
| 20 | | Delayed Shipping | | | | | | | | |
| 21 | *Account* | | | | | | | | | |
| 22 | | Change Details | | | | | | | | |
| 23 | | Change Password | | | | | | | | |
| 24 | | See Past Orders | | | | | | | | |
| 25 | | See Current Orders | | | | | | | | |

# Another example of an overview



# Scoring



- Decisions for testing are based on criteria.
- Scoring is about making these criteria explicit and comparable.

# Criteria

## Risk criteria

Important for the Business

High scores = High risk

Example: Frequency of use
1 – Hardly any users
5 – Lots of daily users

5 means higher risk, thus a better reason to test more.

## Priority criteria

Important for the Tester

High scores = High priority

Example: Testability
1 – Very difficult to test
5 – Very easy to test

5 means higher priority, because it’s easier to create value

# Examples of risk criteria

### Frequency

How often something is used

### Damage/impact

How big the problem is if it fails in production

### NPS impact

Net Promotor Score; how much it impacts client satisfaction

### Dependencies

How dependent the functionality is on other connected systems

### Elapsed time

How long ago it was changed. The longer ago, the lower the risk

# Examples of priority criteria

### Testability

How easy it is to test this functionality

### Automatability

How easy it is to automate this functionality

### Test data

How much test data is needed (in preparation and/or maintenance)

### Boredom

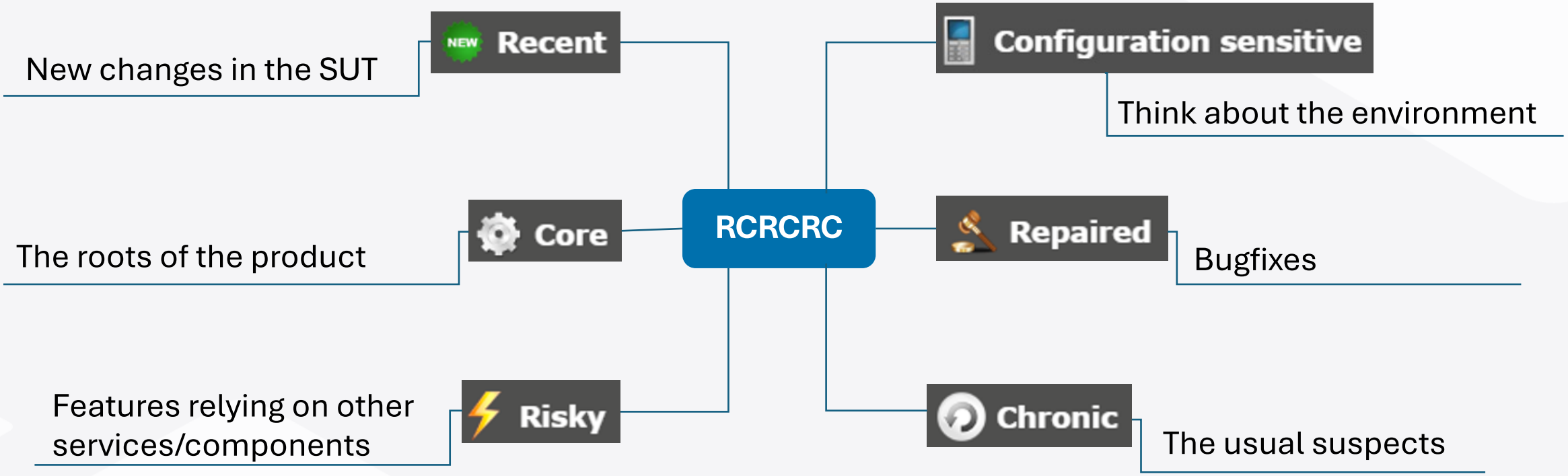
Boring tests bring greater joy when automated

### Time gained

Difference between manual and automated execution time

# When you have absolutely no idea...

...use a heuristic



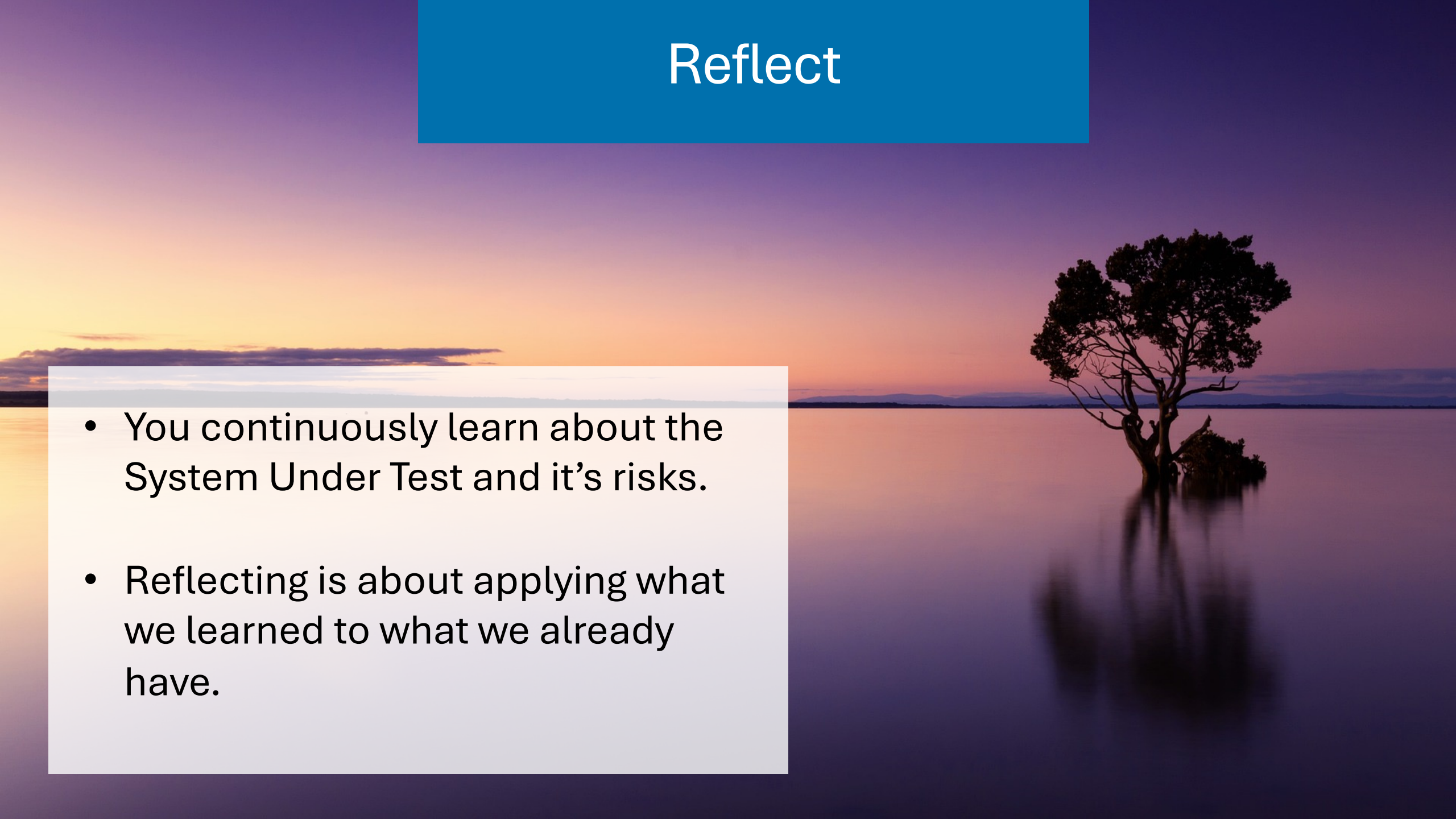
# Example of scoring

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Webshop | | | | | | | | | |
| 2 | **Overview** | | **Categories** | | | | | **Scoring** | | |
| 3 | **Category** | **Functionality** | **Frequency** | **Damage** | **Last changed** | **Testability** | | **Risk** | **Priority** | **Score** |
| 4 | *Search* | | | | | | | | | |
| 5 | | Find Products | 5 | 5 | 3 | 4 | | 25 | 12 | 37 |
| 6 | | Compare Products | 3 | 2 | 1 | 3 | | 6 | 3 | 9 |
| 7 | | Filter Products | 3 | 4 | 2 | 3 | | 12 | 6 | 18 |
| 8 | *Shopping Cart* | | | | | | | | | |
| 9 | | Add Items | 3 | 5 | 1 | 5 | | 15 | 5 | 20 |
| 10 | | Edit Items | 2 | 4 | 1 | 5 | | 8 | 5 | 13 |
| 11 | | Remove Items | 3 | 3 | 1 | 5 | | 9 | 5 | 14 |
| 12 | | Save Shopping Cart | 2 | 1 | 3 | 4 | | 2 | 12 | 14 |
| 13 | | Calculate Shipping | 2 | 2 | 1 | 3 | | 4 | 3 | 7 |
| 14 | *Checkout* | | | | | | | | | |
| 15 | | Ship to Home | 5 | 5 | 2 | 4 | | 25 | 8 | 33 |
| 16 | | Ship to Postal Point | 3 | 4 | 1 | 3 | | 12 | 3 | 15 |
| 17 | | Pay by iDeal | 5 | 5 | 3 | 2 | | 25 | 6 | 31 |
| 18 | | Pay by Creditcard | 3 | 2 | 1 | 1 | | 6 | 1 | 7 |
| 19 | | Pay by Klarna | 2 | 1 | 3 | 1 | | 2 | 3 | 5 |
| 20 | | Delayed Shipping | 1 | 1 | 2 | 1 | | 1 | 2 | 3 |
| 21 | *Account* | | | | | | | | | |
| 22 | | Change Details | 4 | 1 | 1 | 5 | | 4 | 5 | 9 |
| 23 | | Change Password | 3 | 2 | 1 | 5 | | 6 | 5 | 11 |
| 24 | | See Past Orders | 4 | 3 | 1 | 3 | | 12 | 3 | 15 |
| 25 | | See Current Orders | 5 | 4 | 2 | 4 | | 20 | 8 | 28 |

# Recommendations for scoring

- Gather input from different team members, not just the testers
- Using more than 6 columns generally will not make your model better
- Low scores don’t equal no testing, it simply means other items are more important

# Reflect



- You continuously learn about the System Under Test and it’s risks.
- Reflecting is about applying what we learned to what we already have.

# Example of reflection – Scores

| | A | B | J |
|---|---|---|---|
| 1 | Webshop | | |
| 2 | | | |
| 3 | **Category** | **Functionality** | **Score** |
| 4 | *Search* | | |
| 5 | | Find Products | 37 |
| 6 | | Compare Products | 9 |
| 7 | | Filter Products | 18 |
| 8 | *Shopping Cart* | | |
| 9 | | Add Items | 20 |
| 10 | | Edit Items | 13 |
| 11 | | Remove Items | 14 |
| 12 | | Save Shopping Cart | 14 |
| 13 | | Calculate Shipping | 7 |
| 14 | *Checkout* | | |
| 15 | | Ship to Home | 33 |
| 16 | | Ship to Postal Point | 15 |
| 17 | | Pay by iDeal | 31 |
| 18 | | Pay by Creditcard | 7 |
| 19 | | Pay by Klarna | 5 |
| 20 | | Delayed Shipping | 3 |
| 21 | *Account* | | |
| 22 | | Change Details | 9 |
| 23 | | Change Password | 11 |
| 24 | | See Past Orders | 15 |
| 25 | | See Current Orders | 28 |

The scores are a tool to help you find:

- What to test
- How much to test
- In what order to test (often overlooked!)

Use this knowledge for creating new testcases and to reflect on your existing testset by **coverage** or by **effort**.

# Example of reflection – Execution time (effort)

| | A | B | J | K | L |
|---|---|---|---|---|---|
| 1 | Webshop | | | | |
| 2 | **Overview** | | **Scoring** | | **Reflection** |
| 3 | **Category** | **Functionality** | **Score** | | **Time** |
| 4 | *Search* | | | | |
| 5 | | Find Products | 37 | | 60 |
| 6 | | Compare Products | 9 | | 25 |
| 7 | | Filter Products | 18 | | 20 |
| 8 | *Shopping Cart* | | | | |
| 9 | | Add Items | 20 | | 20 |
| 10 | | Edit Items | 13 | | 15 |
| 11 | | Remove Items | 14 | | 10 |
| 12 | | Save Shopping Cart | 14 | | 5 |
| 13 | | Calculate Shipping | 7 | | 10 |
| 14 | *Checkout* | | | | |
| 15 | | Ship to Home | 33 | | 25 |
| 16 | | Ship to Postal Point | 15 | | 10 |
| 17 | | Pay by iDeal | 31 | | 30 |
| 18 | | Pay by Creditcard | 7 | | 40 |
| 19 | | Pay by Klarna | 5 | | 10 |
| 20 | | Delayed Shipping | 3 | | 20 |
| 21 | *Account* | | | | |
| 22 | | Change Details | 9 | | 20 |
| 23 | | Change Password | 11 | | 20 |
| 24 | | See Past Orders | 15 | | 5 |
| 25 | | See Current Orders | 28 | | 30 |

Quick wins! -> handy to add these to the regressiontest

Time to evaluate the testcases:

- Can it be done in fewer steps?
- Or in fewer testcases?
- Can they be quicker when automated?

# Example of reflection – Coverage

| | A | B | J | K | L | M |
|---|---|---|---|---|---|---|
| 1 | Webshop | | | | | |
| 2 | | | | | | |
| 3 | **Category** | **Functionality** | **Score** | **Total testcases** | **Testcases** | **Coverage** |
| 4 | *Search* | | | | | |
| 5 | | Find Products | 37 | 8 | 8 | 100 |
| 6 | | Compare Products | 9 | 12 | 3 | 25 |
| 7 | | Filter Products | 18 | 4 | 3 | 75 |
| 8 | *Shopping Cart* | | | | | |
| 9 | | Add Items | 20 | 5 | 6 | 120 |
| 10 | | Edit Items | 13 | 6 | 5 | 83 |
| 11 | | Remove Items | 14 | 3 | 1 | 33 |
| 12 | | Save Shopping Cart | 14 | 4 | 2 | 50 |
| 13 | | Calculate Shipping | 7 | 8 | 3 | 38 |
| 14 | *Checkout* | | | | | |
| 15 | | Ship to Home | 33 | 4 | 2 | 50 |
| 16 | | Ship to Postal Point | 15 | 6 | 6 | 100 |
| 17 | | Pay by iDeal | 31 | 2 | 2 | 100 |
| 18 | | Pay by Creditcard | 7 | 6 | 6 | 100 |
| 19 | | Pay by Klarna | 5 | 7 | 7 | 100 |
| 20 | | Delayed Shipping | 3 | 5 | 1 | 20 |
| 21 | *Account* | | | | | |
| 22 | | Change Details | 9 | 4 | 4 | 100 |
| 23 | | Change Password | 11 | 5 | 5 | 100 |
| 24 | | See Past Orders | 15 | 14 | 2 | 14 |
| 25 | | See Current Orders | 28 | 12 | 3 | 25 |

Looks good

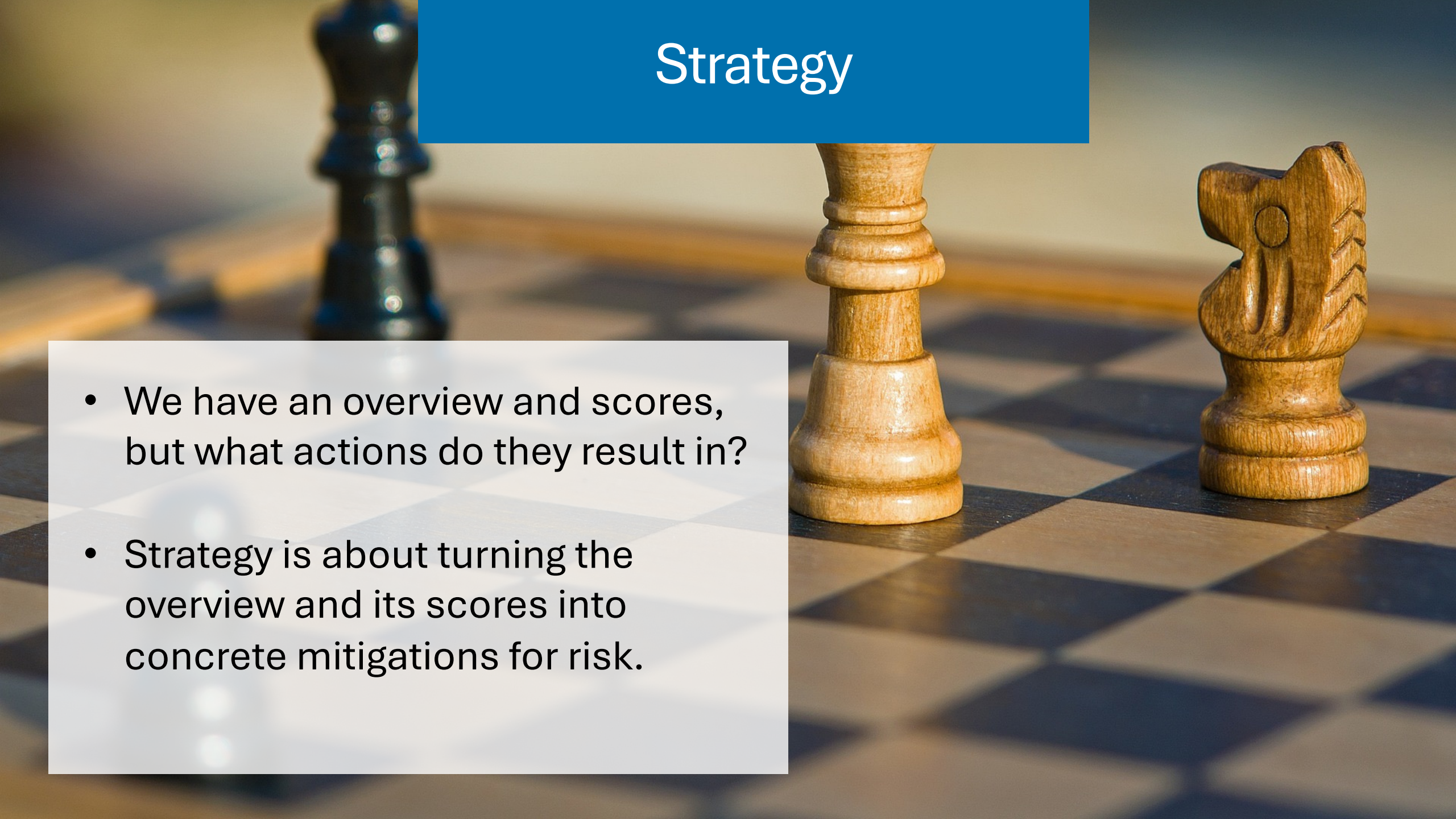
Add items: what happened?

Calculate shipping?

Time to delete some testcases!

Could use a bit more attention

# Strategy



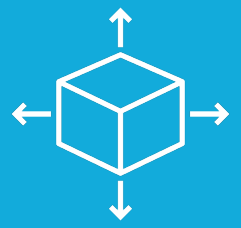
- We have an overview and scores, but what actions do they result in?
- Strategy is about turning the overview and its scores into concrete mitigations for risk.

# Strategy preparation

1. Find the highest and lowest score
2. Create risk levels
3. Define what each risk level means for your approach

# Defining the meaning of the risk levels

You have the opportunity to set the levels and their definition according to what is important for your testing.

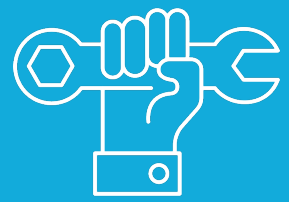


Size: should risk levels all be the same size?

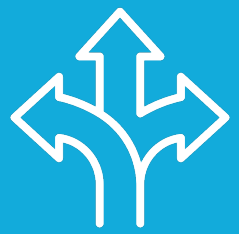
Order: In what order will you execute the tests?

Frequency: How often will you execute the tests?

Effort: Make use of timeboxes if needed



How: Exploratory testing; Manual or Automated?



What: All functionality, most important or happy flows?

# Example of Strategy

| | A | B |
|---|---|---|
| 1 | **Risk level** | **Activities** |
| 2 | Low | First deployment: 15 min exploratory testing |
| 3 | 1 - 15 | 1 testcase in regressiontest |
| 4 | | Testcase marked as 'Low' (can be skipped when under time pressure) |
| 5 | | |
| 6 | Medium | Most important flows in regressiontest |
| 7 | 16 - 25 | 20 min exploratory testing |
| 8 | | When first deployed: manual testing of remaining flows |
| 9 | | |
| 10 | High | Happy flows and most common error situations in regressiontest |
| 11 | 26 and higher | Testcases placed at top of regressiontest |
| 12 | | 30 min of additional exploratory testing every release |
| 13 | | |

| | A | B |
|---|---|---|
| 1 | **Risk level** | **Activities** |
| 2 | Low | 20 min exploratory testing |
| 3 | 1 - 10 | 1 automated testcase |
| 4 | | |
| 5 | Medium | 30 min exploratory testing |
| 6 | 11 - 22 | Most important cases automated (happy + errors) |
| 7 | | When first deployed: manual testing of remaining flows |
| 8 | | |
| 9 | High | 1h exploratory testing |
| 10 | 23 and higher | Happy cases automated |
| 11 | | Minimum 50% of error situations automated |
| 12 | | 1h of additional exploratory testing every release |

# 4. Be the expert

How can this help you  showcase your professionalism

# A professional approach to testing

With these steps you can show:

- The level of thought that you put into your tests
- The though choices in deciding when testing less/deleting tests
- The value of risk-based testing
- The way you understand the business and the value you are delivering to the business

# Take control as the expert

**You** gather input for the overview and create it

**You** bring the team together

**You** lead the discussion on the criteria

**You** moderate the scoring proces (prevent everything from being a ‘5’!)

**You are the test expert!**

# The rewards are for you

Optimized test set & Strategy

Shown others the level of professionalism you put into your tests

Taken the whole team along in the quality mindset

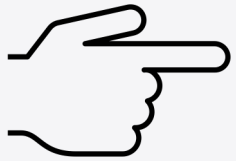
# 5. Wrap up

How to apply this (in a sprint)
Final Thoughts

# First execution of ‘OSRS’

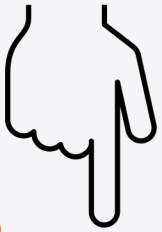

## Create overview

Go through the documentation & SUT and create the overview




## Decide on categories

And start scoring the overview




## Reflect on the test set

And see how well it is balanced and covers risk

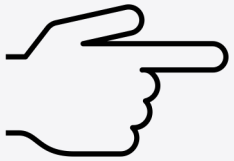

## Define strategy

Find the risk levels and attach actions to them

# Subsequent executions of ‘OSRS’

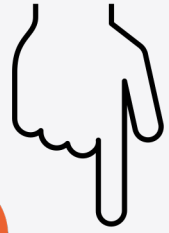

## Update overview

Go through the stories and append the overview




## Update scores

Score new items and update existing scores




## Reflect on the test set

And see how well it is balanced and covers risk


## Apply strategy

Execute the chose strategy and modify the test set if needed

# Starting questions

- What is the value of your testset?
- When do you delete a testcase?
- When you have X testcases to execute/automate, which one is next?
- Which tests go into the regressionset?
- How do you determine how much testing is needed for new userstories?

# Final thoughts

## Everybody can use OSRS!

- Start creating your overview
- Start thinking of the criteria that are important for your project
- Gather the team and start scoring…

# Questions?

https://www.linkedin.com/in/suzanne-kraaij/


expoQA25
MADRID
May 20th,
21st & 22nd
2025
Thank you for attending
expoqa.eu