

# TestSphere

Cards Against Shallow Testing



Beren Van Daele  
@isleoftesting





Beren Van Daele

Ghent, BE

Co-Creator of TestSphere

Freelance Test Coach & Advisor

[@isleoftesting.com](https://isleoftesting.com)

Organiser of BREWT peer  
conference

Remote-working Product Owner

Once professional semi-nude model



# Exploratory Lifestyle

All my needs sometimes fit into:

- One foldable touring bike
- Four bags

And sometimes into:

- One duffel bag



# Spreading love for deliberate testing



Beren Van Daele  
@isleoftesting



# Helped by wonderful people

## Work Together When You Can



**Marcel Gehlen**

@Marcel\_Gehlen

. @isleoftesting needs help to n understanding memes. His homework is to come up with #testsphere memes. Who ever wants to give him ideas can tweet them.

8:22pm · 13 Aug 2018 · Tweetbot for iOS

📍 Tiergarten, Berlin, Duitsland

15 REPLIES 4 RETWEETS 13 LIKES



Beren Van Daele  
@isleoftesting







# Why this talk?



Concept Name

# TESTABILITY

## QUALITY ASPECTS

Category

SubCategory

**Maintainability Aspect:** Everything and anything that impacts how easy or hard your testing efforts are.

Quick explanation

3 different Examples

**1** Cheat codes are often ways to test a game more easily. If you want to test being all powerful, just enter "↑, ↑, ↓, ↓, ←, →, ←, →, B, A, start, select". Don't forget it's a shortcut!

**2** Do you have enough environments to test on? What do you use them for? Is this clear for every team member?

**3** Can you manipulate system time to test functionality that only comes into play at certain dates?



# How have people u



**lisacrispin**  
@lisacrispin

Volg je nu

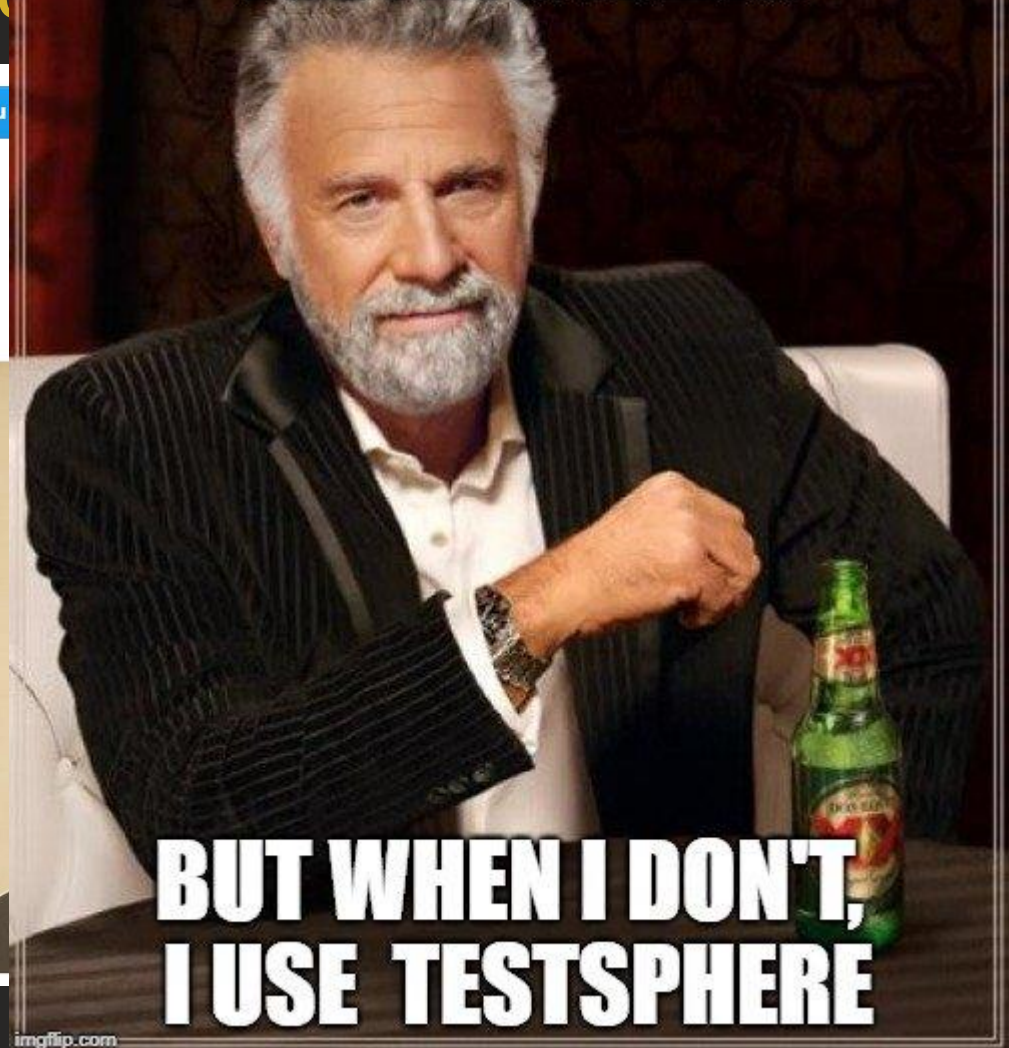
We're trying out risk storming with @isleoftesting's @TestSphere cards (thanks @ministryoftest for my #TestBash Dublin speaker gift!) #CAST2018

f-gun,  
ork!



7:47pm · 9 Aug 2018 · Twitter for iPhone

**I ALWAYS HAVE  
TESTING IDEAS**



imgflip.com



Beren Van Daele  
@isleoftesting





# Ice Breaker at Conferences



**Jokin Aspiazu** @JokinAspiazu · 9 nov. 2016

Hey, #testers, you really want to know the @TestSphere card deck!  
#VLCTesting16



# Home Decoration



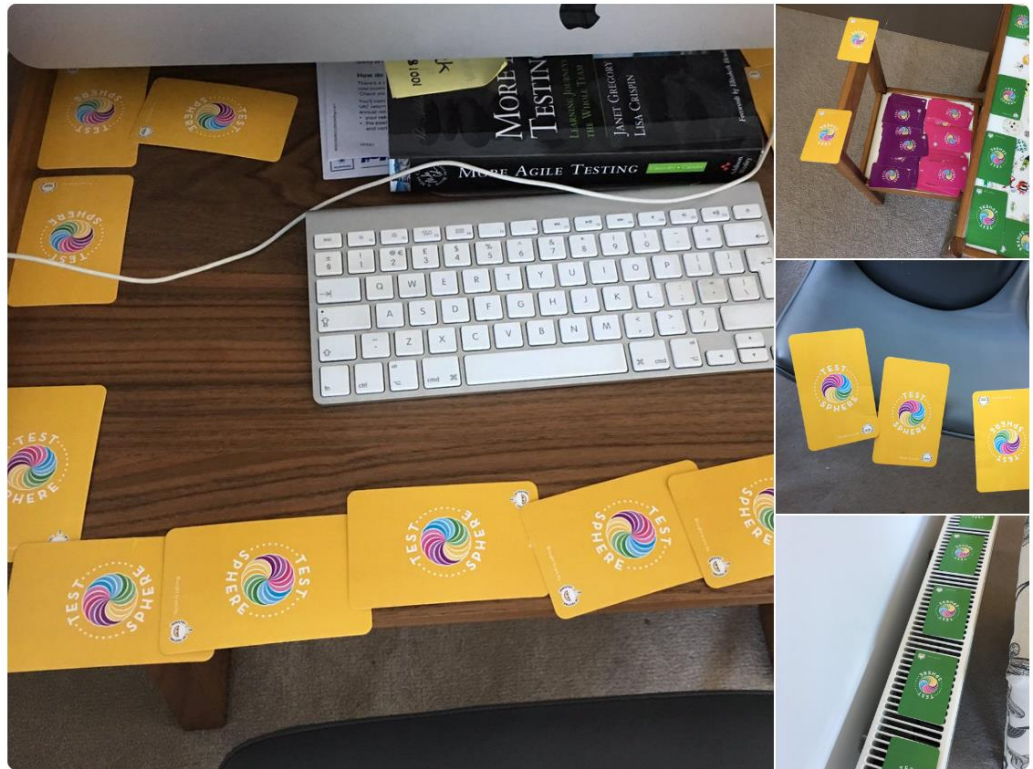
**Simon Tomes**

@simon\_tomes

Volg je nu



A break from the sun and the lounge gets [@TestSphere](#)'d by our daughter!



18:32 - 9 apr. 2017



Beren Van Daele  
@isleoftesting



# Handy Coke Chopper



Beren Van Daele  
@isleoftesting





# Underwear!



Beren Van Daele  
@isleoftesting





# Why this talk?

Testers are **dispersed**  
**and immersed** into development teams  
but need **diversity** to  
bring **value**.

We need to refresh our Ideas &  
Collective Mind.





# Problem 1:

The only **Tester** in the Team



# We lose our Identity, our Craftership

Knowledge sharing?

Co-Learning?

Inspiration?



*Cucumbers get more pickled than brine gets cucumbered.* - Prescott's Pickle Principle

- [Gerald Weinberg](#), may he rest in peace.

## Take-away:

Gerald Weinberg. Read everything he ever wrote.



# Story-Telling Lunch and/or Drinks



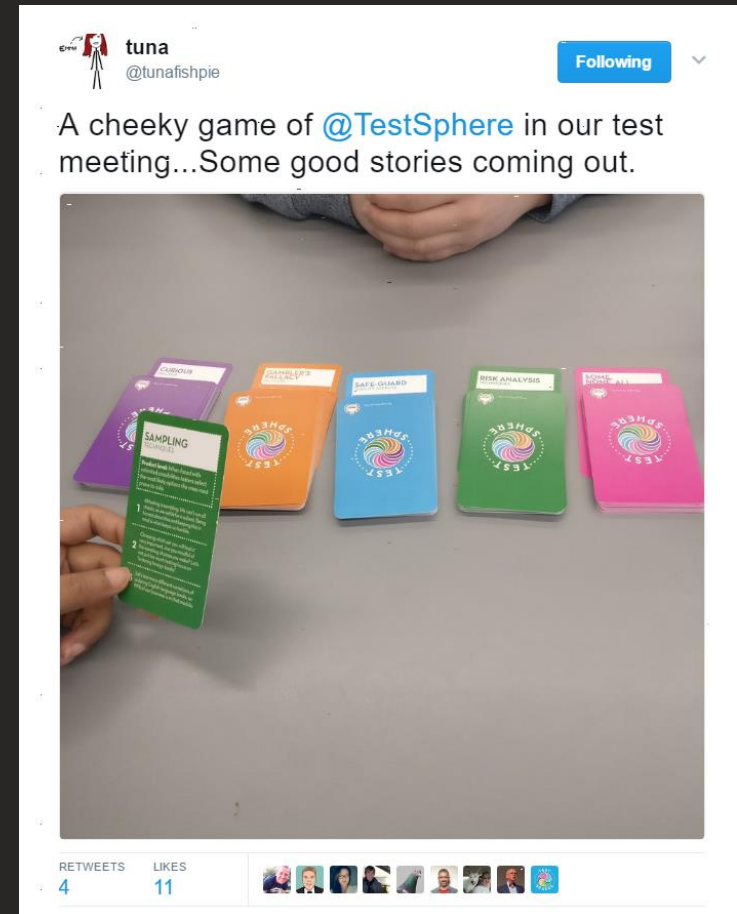


# The Story-Telling Game

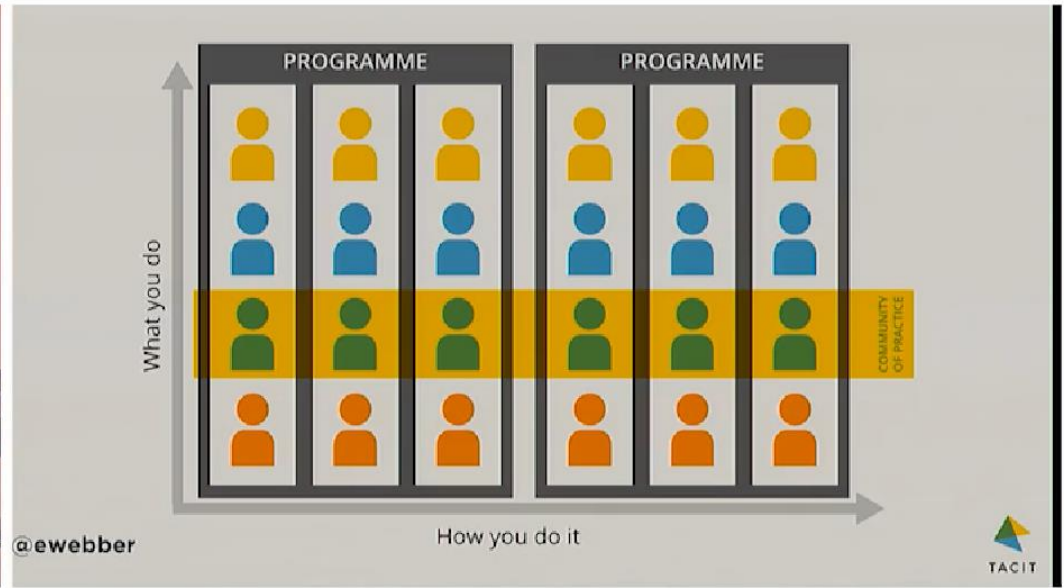
- Step 1: Find a group of 4 to 8 persons
- Step 2: Divide the deck by category (20 cards each)
- Step 3: Depending on the experience of the group: reveal one or more cards
- Step 4: When a person can think of a story that features all revealed cards he or she knocks on the table
- Step 5: Tell the story
- Step 6: This person takes the revealed cards as full points
- Step 7: Other people can also tell their stories to get unrevealed cards for half points.



# Get People Together



## Communities Of Practice, The Missing Piece Of Your Agile Organisation - Emily Webber



 #testbash

Take-away:  
Check out Emily Weber



Beren Van Daele  
@isleoftesting



# Retrospectives



**sparrowsgo** Hannah

Jul '17

I decided to try using TestSphere as a tool in hosting my first retro today! The team had been using the 😞 😐 😊 categories so I thought it would add some depth to that discussion.

So I laid out the cards on the table grouped into the positive/negative/neutral categories and then I asked everyone to choose 1 positive and 1 negative card that best described their feelings over the last week. They had 30 seconds to do this and I timed it by playing retro arcade music (retro, get it? har har).

After that we went around and discussed why people had chosen the cards and what experiences had made them feel that way. It was a great start to the retro and I got some great feedback that people had really enjoyed it!

3 ❤️ 🔗 ... ↩ Reply





Whether you're an Introvert or Extravert,  
Do-er or Thinker,...

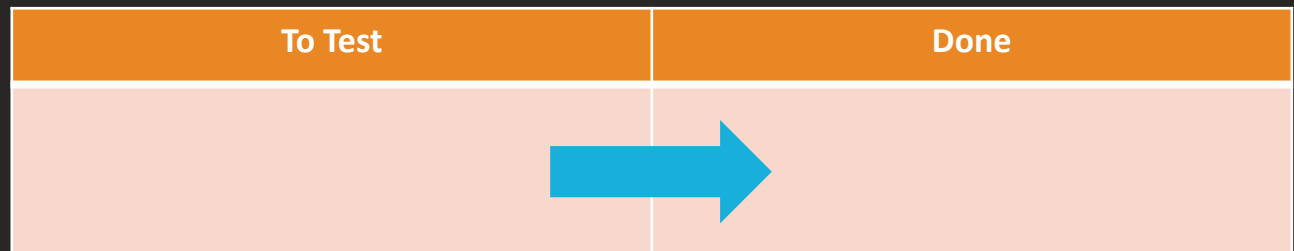
There are little more effective ways for  
Learning and Motivating than the  
Honest Cooperation between a Well-  
Intended group of people with a Common Goal.



# Problem 2



We forget **why** we test





How's Testing  
going?



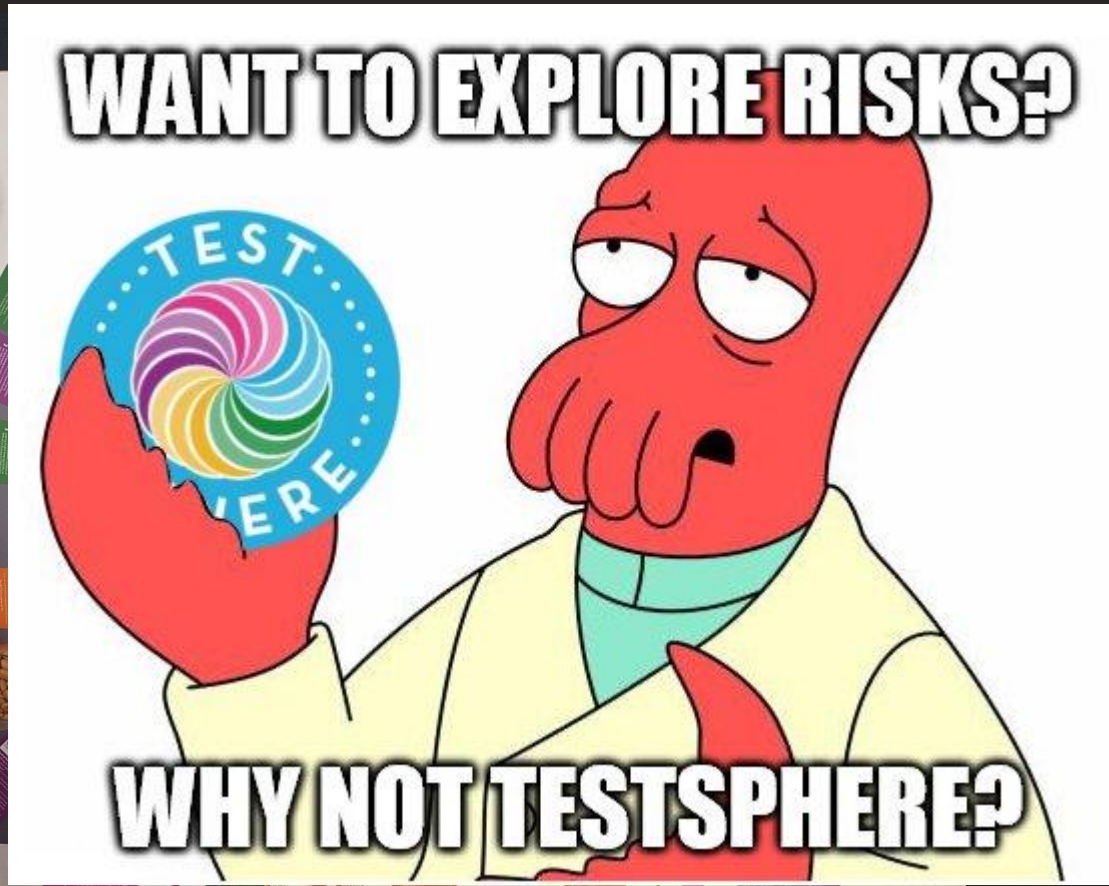
# Why RiskStorming?

Because I believe, one of the hardest questions to ask a tester is:

What is your current **Test Strategy**?







Take-away 2:  
How to RiskStorm  
with TestSphere



Beren Van Daele  
@isleoftesting



# Phase 1

What's **most important** to your app?



## INTERNATIONALISATION

### QUALITY ASPECTS

**Usability Aspect:** Currency, time zone, language, right-to-left... Can you use your application anywhere in the world?

1 What languages does your application support? Does it keep specific differences into account, such as right-to-left reading in e.g. Arabic?

2 Currency. Is it visualised correctly? How are conversions done? Is there a conversion pattern or a dynamic service? What happens when the service is unreachable?

3 Laws, permissions, network security from country to country. If you want your product to be used in China, you will likely need to adjust a few things.

## OPERATIONS

### QUALITY ASPECTS

**Maintainability Aspect:** How easy or hard is it to rectify a problem in production after the product is released?

1 Is there documentation about the application, so that a different team can handle maintenance of the product?

production release. How fast and efficient can you roll back?

Is the code easy to read and maintain your way through? You may be asking for a change from a new team member.

## TESTABILITY

### QUALITY ASPECTS

**Maintainability Aspect:** Everything and anything that impacts how easy or hard your testing efforts are.

1 Cheat codes are often ways to test a game more easily. If you want to test being all powerful, just enter "↑, ↓, ←, →, ←, →, B, A, start, select". Don't forget it's a shortcut!

Do you have enough environments to test? What do you use them for? Is it clear for every team member?

Can you manipulate system time to test functionality that only comes into play at certain dates?

## RESOURCE MANAGEMENT

### QUALITY ASPECTS

**Efficiency Aspect:** How does your app handle resources? Can they be reduced?

1 How much data is stored in cache and does it really help your app? Or does it only slow you down?

2 How much storage does your application need? Some smartphones have a limited capacity for storage.

3 Can you eliminate unnecessary features?

## USER FRIENDLINESS

### QUALITY ASPECTS

**Usability Aspect:** Will users find the product convenient and easy to work with? But more important: will they enjoy it?

1 What makes an app attractive? Speed, ease-of-use, reliability?

2 Are the most important features available in the middle of your screen? Have your app eye-track-tested and find out whether people easily find what they're looking for.

3 Some things are intrusive, yet people love them. For example the group-on-space-cat-logo. This makes your app unique, identifiable and memorable.



# Phase 2



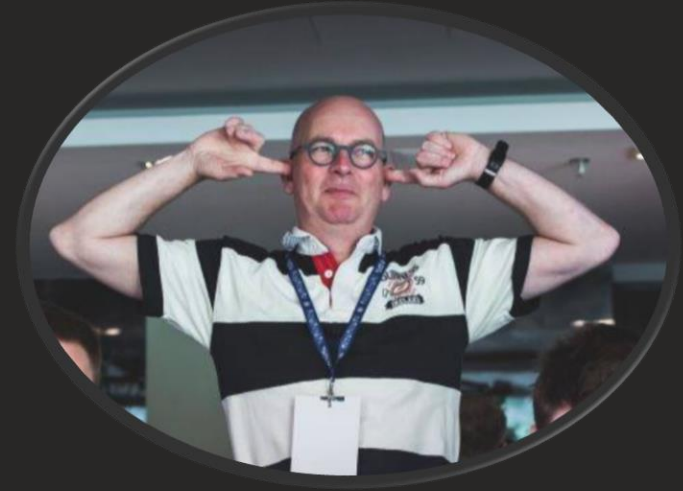
What **Risks** can impact these most important **Quality Aspects**?



## What do risks look like?

There is a risk that **Event** will happen and **Cause** so that we are impacted by **Result**.

There is a risk that **some users can't access critical paths** because **we forgot about certain minority user groups**, therefore **we might lose our certifications as a healthcare software provider**.





# Phase 3



How do we **mitigate** those risks?

Test Techniques  
Heuristics  
Patterns



## PERSONAS TECHNIQUES

**Project level:** Testing and analysing from the viewpoint of agreed upon generated user profiles.

- 1 You identify arch-users and test from their point of view. For example: Tracy from the finance department. She finds language important and knows a lot about contracts.

## 2 INTERRUPTION PATTERNS

**Approach:** Stop, replay, pause and begin again. Can your application handle exceptional flows?

- 1 Wizards sometimes have a point where they save initial data. What happens to that data if you cancel the wizard near the end?
- 2 What happens when you leave your application alone for a coffee break? A lunch break? A whole night?
- 3 You're walking in the park and suddenly a pokémon jumps from behind a tree. You tap it and go into 'catch the pokémon' view. Deactivate the screen, make a phone call and then resume catching it.

## ADVENTUROUS FEELINGS

**Positive feeling:** The feeling when you want to explore something new or go to new depths you never been to before.

- 1 Go have lunch with a key user you don't talk that often with.
- 2 Try out a new tool to enhance your testing.

## 3 DATA ANALYSIS TECHNIQUES

**Product level:** Inspecting, cleaning, transforming, and modelling data with the goal of discovering useful information.

- 1 Legacy data is often very fragile. Gaps, bad data or displacement could riddle the database. Usually a few good queries can quickly find whether your data is valuable or not.
- 2 Consider improving the visualisation of your data if it's too hard or too extensive to go through. Add colour, zoom out and look for patterns. Data input and output are both immensely important.
- 3 Pay attention to how your input gets changed, saved and returned. How do you store data elements?

## LOG-DIGGING PATTERNS

**Approach:** Like treasure maps, logs hold locations to most valuable treasures. You'll be looking for abnormalities. Weird values, error messages or peaks.

- 1 At certain times you are going to encounter problems. Logs can give you more insight in what really happened. Increasing readability of logs greatly improves their use.

## 2 ALWAYS AND NEVER HEURISTICS

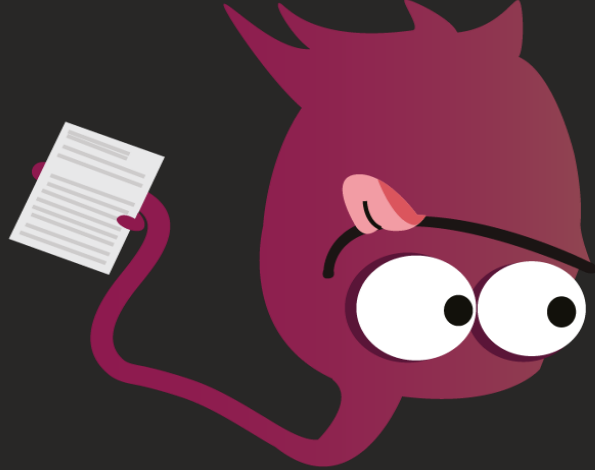
- 1 **Exploratory Heuristic:** Look for the keywords "always" and "never" in your specifications. Try to break those rules.
- 1 There should always be at least one document linked to your transaction.
- 2 The name field should never take more than 50 characters.
- 3 The application should never have any downtime.





Beren Van Daele  
@isleoftesting





## What does a good Strategy look like?

1. State your mission
2. What are the risks?
3. Think about Coverage
4. Strengths of the Team & Tools
5. Interesting measurements
6. What do you have vs. What do you need?

**Take-away:**  
One page Test  
Strategy format

Special thanks to:  
Claire Reckless  
[@clairereckless](#)



Beren Van Daele  
[@isleoftesting](#)



| <b>Mission</b><br>What's most important to test and why do we test for it?                                                                                                                                                                                                                                              | <b>Coverage</b><br>What do we test and what don't we?                                                                                                                                                                                                                                                                                                                                                                   | <b>Resources</b><br>What do we need to test?                                                                                                                                                                                                                   | <b>Strategy</b><br>How will use our resources to cover for the most important risks?                                                                                                                                                                                                                                                                                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Why?</b><br>Functionality, User experience, Performance & Security all embody the core aspect of the application. We need to show new and clear functionality that is performant so that Backoffice sees business value.<br>Security is a constant issue as user data is highly confident and interactions are many. | <b>Do Test</b> <ul style="list-style-type: none"> <li>- Main business flows</li> <li>- Input/output of fields frequently used by Customer</li> <li>- Different flows identified by persona's</li> <li>- API communication between Front End and Back End</li> <li>- Regression risks between new and old functionality (story-driven)</li> <li>- UX tours</li> <li>- Different browsers as per specification</li> </ul> | <b>People</b><br>2 Testers and a Test Lead (per 10 Developers)<br><b>Security Specialist needed</b>                                                                                                                                                            | <b>Time</b><br>One sprint of two weeks, assuming we can start immediately and aren't blocked for more than half a day.<br><br>Nightly performance runs.<br>Daily Test Sessions.<br><b>Needs process to handle Test Data</b>                                                                                                                                          |
| <b>Risks</b><br>Lose business engagement through different bugs on malfunctioning, complicated and overdesigned software. Thereby having our freedom & money cut off.<br>Security leaks, both software and people focused can leave holes for exploits.                                                                 | <ul style="list-style-type: none"> <li>- Tablet mode &amp; phone mode for specific devices</li> <li>- Special focus on saving and showing customer data</li> <li>- Explore API input through performance test tool</li> </ul>                                                                                                                                                                                           | <b>Tools</b><br>Charters<br>Main business scenarios/Test Cases<br>Personas<br>Neoload<br>Performance Test environment                                                                                                                                          | <b>Formalisation</b><br>Keep notes on Test Sessions in Charters<br>Update business scenarios after findings & discussion                                                                                                                                                                                                                                             |
| <b>Business value</b><br>The new customer module will bring a modern feel to the application and will, for the first time ever, enable the customer to add and update their own data.<br>Making this a high quality, useful feature will increase customer engagement to new levels.                                    | <b>Don't Test</b> <ul style="list-style-type: none"> <li>- Already accepted functionality</li> <li>- Accessibility tool runs for next sprint (pending decision by management)</li> <li>- Sorting, filtering, searching...</li> </ul>                                                                                                                                                                                    | <b>Dependencies</b> <ul style="list-style-type: none"> <li>- Outlined specifications or narrow contact with Product Owner/customers</li> <li>- Test Environment to be accessible and up to date</li> <li>- Documentation/test framework for the API</li> </ul> | <b>Report</b> <ul style="list-style-type: none"> <li>- Issues found, fixed and remaining per functionality and priority.</li> <li>- Remaining risks for go-live</li> <li>- How confident we are in the product, the testing and reasons why.</li> <li>- Performance reports on number of errors, latency, response times &amp; number of concurrent users</li> </ul> |



# Sell it to Management



Beren Van Daele  
@isleoftesting



## Don't Think So Close To Me - Managing Critical and Social Distance in Testing – James Bach



### Archetypical Examples

| Social Distance | Critical Distance | Pattern                                                                                                                                                                                                    |
|-----------------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LOW             | LOW               | <ul style="list-style-type: none"><li>self-checking</li><li>"thinking twice"</li><li>"cheerleader mode" spouse</li><li>your children when asking for money</li></ul> <b>Harmonious</b>                     |
| LOW             | HIGH              | <ul style="list-style-type: none"><li>test expert (co-located)</li><li>your therapist</li><li>"protective partner mode" spouse</li><li>your children (ages 0-11)</li></ul> <b>Productive</b>               |
| HIGH            | HIGH              | <ul style="list-style-type: none"><li>test expert (third-party or offshore)</li><li>IRS or SEC (supposedly)</li><li>ex-spouse or blind date</li><li>your children (ages 11-19)</li></ul> <b>Irritating</b> |
| HIGH            | LOW               | <ul style="list-style-type: none"><li>offshore "testers" who only follow scripts</li><li>1986 NASA Safety program</li><li>fearful employees</li></ul> <b>Reticent</b>                                      |

Copyright © 2016 Satisfice Inc.



# #testbash

**Take-away:**

Don't think so close to me, by James Bach



Beren Van Daele  
@isleoftesting



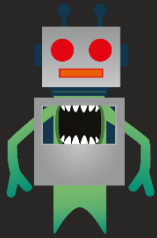
When we don't talk to other testers  
→ We stop bringing a Diversity of Ideas

= Low Social Distance, Low Critical Distance  
= Harmonious

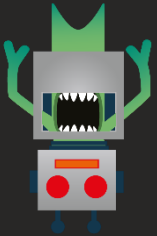
When we don't talk to our team  
about what Quality is for us  
→ We stop bringing Value

= High Social Distance, High Critical Distance  
= Irritating

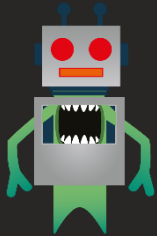




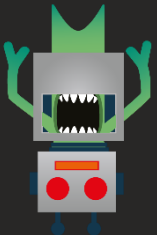
Characters from Ministry of Testing & Rob Lambert  
Communities of Practice provided by Emily Weber



One Page Test Plan provided by Claire Reckless  
Risk Phrasing provided by Ard Kramer



RiskStorming Strategy designed by Andreas, Beren & Marcel  
→ [isleoftesting.com](https://isleoftesting.com)



TestSphere co-created by Ministry of Testing and Beren Van Daele  
→ [Store.ministryoftesting.com](https://Store.ministryoftesting.com)

