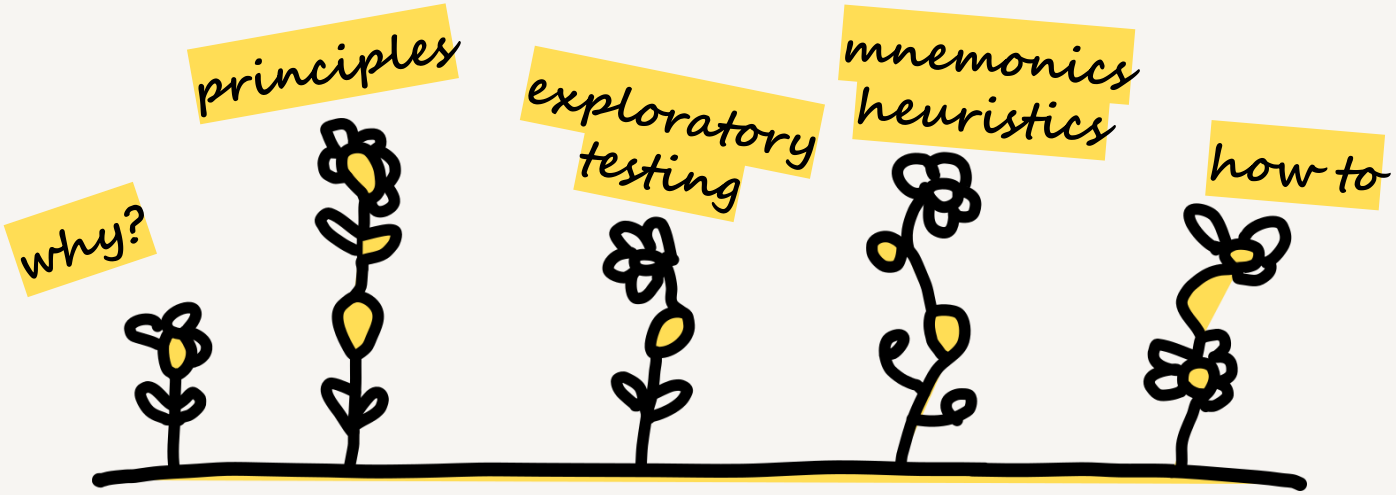
Exploratory testing

2026 (Oslo)

yulai de meer fjeld

agenda


why?
principles
exploratory testing
mnemonics heuristics
how to

ambitions

Testing is a
"thinking" activity

Understand the
fundamentals



Different
mindsets



why do humans test?

why humans test

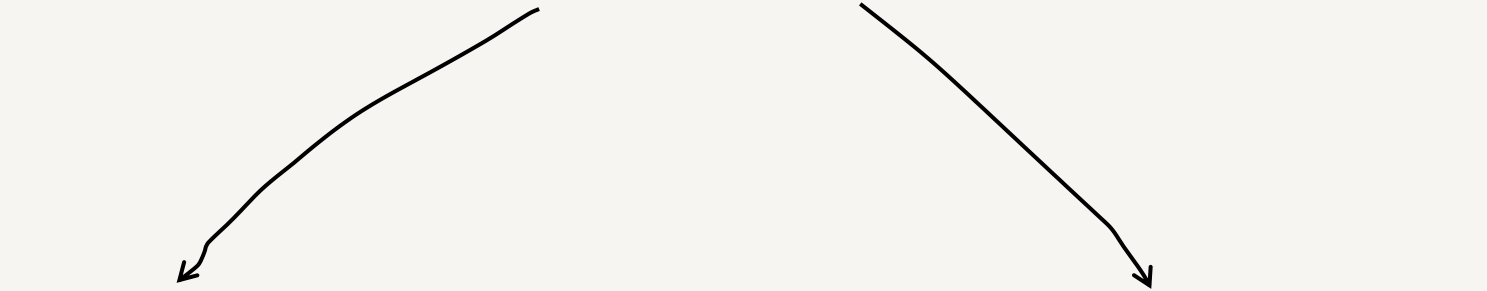
Make sure it
works as expected

Make sure it
works



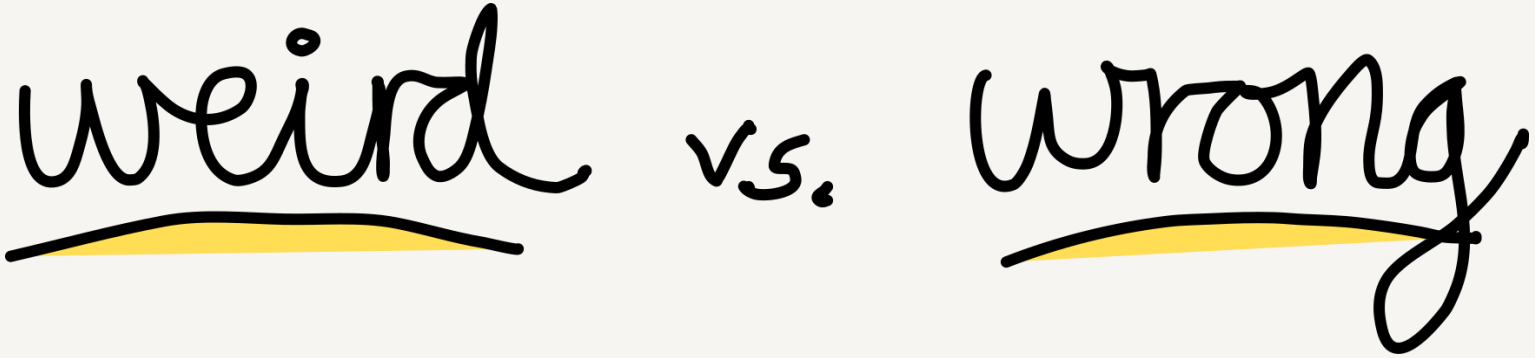
Make sure it is
good

Make sure it is good...



for whom?

in what context?



unstated: expectation, taste, rhythm

specified: the rule, the contract

where ai tends to fall short

patterns worth naming

TONE



error messages that are correct but condescending, alarming, or cold.

CONVENTION



a button in an unexpected place. a color that means the wrong thing here.

FRICTION



the app works. using it is tiring. models don't get tired.

ABSENCE



models test what's there. people notice what's missing.

principles

ISTQB

- testing shows the presence of defects
- exhaustive testing is impossible
- early testing saves time and money
- defects cluster
- pesticide paradox
- testing is context-dependent
- absence of errors fallacy



principles

istqb

- testing shows the presence of defects
- exhaustive testing is impossible
- early testing saves time and money
- defects cluster
- pesticide paradox
- testing is context-dependent
- absence of errors fallacy

human-centered

AN ACT OF LOOKING, NOT OF PROVING

YOU CHOOSE WHAT TO LOOK AT — THAT CHOICE IS THE WORK

UNDERSTAND WHAT IS REALLY DIFFICULT

CONTEXT IS LIVED

A WORKING PRODUCT CAN STILL BE THE WRONG PRODUCT



CONTEXT IS LIVED



testing is context-dependent

software systems are not created equal

- **intended for different** users
- **used in different** ways
- **pose different** risks

factors affecting the test effort

- type of technology (what, old, new)
- type of project (small, large, agile, waterfall)
- type of product (experimental, life-critical)

testing is context-dependent

require different focus
in test

- cannot test all systems the same way
- need information about the system
- tailor test to context

context is lived

same behaviour. three people.

NURSE

night shift, fluorescent light, tired hands



TEENAGER

on a bus, one thumb, spotty signal



PENSIONER

kitchen table, reading glasses, no rush



a 200ms delay isn't the same delay for all three.

You've been training your whole life

Testing is a thinking activity. You've been thinking your whole life.

EXPLORATORY TESTING



simultaneous learning, test design, and test execution



focuses on discovery and relies on the guidance of the individual tester



uncovers defects that are not easily discovered through other approaches



dynamic

EXECUTION OF SOFTWARE / COMPONENTS / SYSTEM

FOCUS ON FUNCTIONALITY / EXPECTED BEHAVIOUR

experience-based

BASED ON PREVIOUS KNOWLEDGE / INTUITION

OFTEN REQUIRES IN-DEPTH DOMAIN KNOWLEDGE

what it is - recap

simultaneous learning, design, and execution.

GUIDED BY THE TESTER, NOT BY A SCRIPT.

SCRIPTED ASKS

DOES IT DO WHAT WE SAID?

EXPLORATORY ASKS

WHAT IS THIS THING, REALLY?

"I before e, except after c"



How many days in May?

Which way do you set the clock in March?

What are the colours of the rainbow?

MNEMONICS



HEURISTICS

mnemonic

a memory technique to aid with retaining and retrieving information

heuristic

a problem-solving approach that involves using practical, intuitive strategies rather than following a rigid set of instructions

Clothes and More

Search for items

Basic T-Shirt

Remaining in stock: 10

1

Add to cart

COUNT

zero, one, many. too many, too few.



- **zero: search for a non-existent item**
- **one: search for a specific item**
- **many: search that returns several items**
- **too many / too few: searching both ends of the spectrum**

Forms and Feedback

Enter your text here

400/400 characters remaining

- [x] Option A
- [ ] Option B
- [x] Option C

Submit

GOLDILOCKS

too big, too small, just right

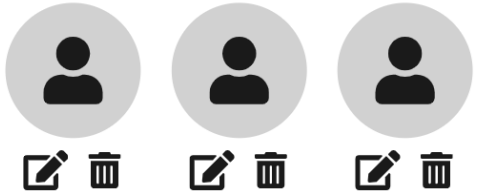
- too big: overload a text field
- too small: leave the text field empty
- just right: happy case input



Student Registry

Search for student

Search



Register Student

First Name

Last Name

E-mail

Phone Number

Register

CRUD

create, read, update, delete

- create: create new user / new user with duplicated attributes
- read: get user / get non-existent user
- update: update user / update non-existent user
- delete: delete user / delete non-existent user



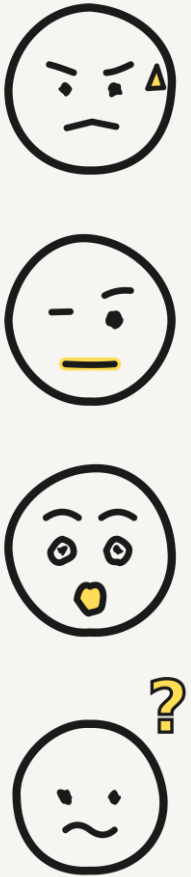
recent, core, risky, configuration, repaired, chronic

- **recent: new features / code added**
- **core: key functionality that simply must work**
- **risky: areas relying on other services / components**
- **config: areas affected by config / environment settings**
- **repaired: code that has been changed during bug fix**
- **chronic: areas that frequently have issues**

trigger heuristics

IDEA ASSOCIATED WITH AN EVENT OR CONDITION THAT TRIGGERS AN ACTION OR REACTION

EMOTIONS AND FEELINGS ARE POWERFUL TRIGGERS



trigger heuristics

emotions are data.

impatience  → the system is too slow for this task => delays?

confusion  → counterintuitive interface?

surprise  → an assumption has been violated

discomfort  → something is off — investigate it

BREAK

10 minutes

how to

how to do exploratory testing

I. DEVELOP A BUG CLASSIFICATION

II. UNDERSTAND THE SYSTEM UNDER TEST

III. CHOOSE A HEURISTIC

IV. CREATE A TEST CHARTER / PLAN

V. CONTINUOUSLY ASSESS THE FINDINGS

1. BUG CLASSIFICATION

- CLASSIFY BUGS FOUND IN PREVIOUS PROJECTS
- ANALYSE ROOT CAUSES FOR THESE BUGS
- DEFINE RISKS (IN LIGHT OF TYPICAL BUGS)

bug taxonomy



INPUT VALIDATION

bad input accepted, good input rejected.



BOUNDARY

edges, off-by-one, min/max values.



CALCULATION

wrong math, rounding, totals that don’t add up.



USABILITY

confusing flow, hidden affordance, dead ends.



SECURITY

auth gaps, leaks, injection, escalation.



LOGIC

valid input, wrong outcome. branches missed.



COMPATIBILITY

browser, device, locale, version drift.



SYNTAX

typos, format errors, malformed payloads.



PERFORMANCE

slow responses, leaks, timeouts, jank.

II. UNDERSTAND THE SYSTEM UNDER TEST

- WHAT KIND OF SYSTEM?
- WHAT KIND OF FUNCTIONALITY?
- WHO ARE THE USERS?

III. CHOOSE A HEURISTIC / MENTAL MODEL

- CHOOSE HEURISTIC THAT BEST FITS THE CONTEXT
- MULTIPLE HEURISTICS CAN BE APPLIED
- CREATE YOUR OWN HEURISTICS OVER TIME

IIII. CREATE A TEST CHARTER (PLAN)

self-reflection

be aware of your own testing decisions

motivation

what?

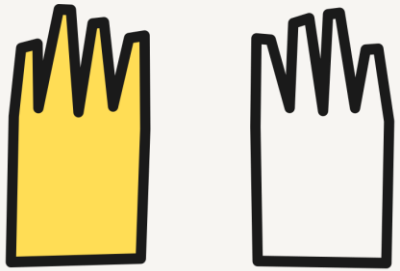
how?

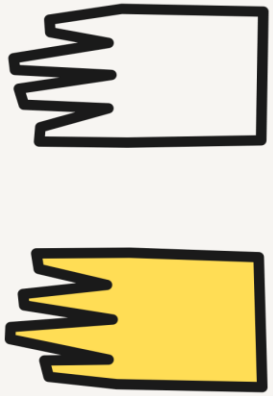
why?

V. ASSESS THE FINDINGS

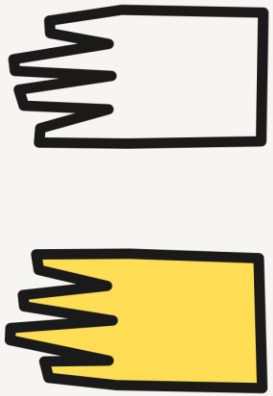
- CONTINUOUS FEEDBACK LOOP
- KEY QUESTION: WHAT DID I LEARN?
- USE FINDINGS TO GUIDE THE SUBSEQUENT TEST EFFORTS

HANDS ON





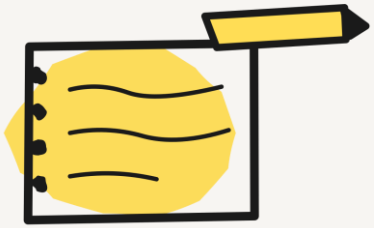
tinyurl.com/life-by-hour



tinyurl.com/the-bug-barn

your turn

test the app.



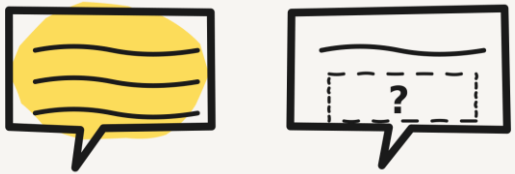
tinyurl.com/exploratory-demo

---

01 write down anything that feels off — even if you can't say why.

02 note the emotion that made you notice. it's part of the report.

claude, live



now watch a model do the same thing.

the question to hold in mind:

what did claude not say that you said?

debrief

three buckets



BOTH FOUND

shared ground. the obvious surface.

ONLY CLAUDE

coverage, structure, edge cases by enumeration.

ONLY YOU

what felt off. tone. absence. nuance.

advantages

- DIFFERENT KINDS OF BUGS => RANDOM NATURE OF TESTING
- RAPID FEEDBACK => LIMITED PREPARATION NECESSARY
- UNCOVER NEW TEST SCENARIOS
- SUITABLE WHEN THERE ARE NO REQUIREMENTS / SPECIFICATIONS

disadvantages

- QUALITY OF TEST DEPENDS ON THE TESTER'S SKILL AND EXPERIENCE
- REQUIRES A CERTAIN LEVEL OF CREATIVITY
- DIFFICULT TO REPRODUCE FAILURE SCENARIOS
- THE "AD HOC" NATURE MAY RESULT IN LIMITED DOCUMENTATION

developing a tester's mindset







- FOSTER CREATIVITY, PROBLEM-SOLVING, ADAPTABILITY
- ENCOURAGE PLAYERS TO QUESTION CONVENTION
- WHAT HAPPENS IF I DO THIS?
- PARALLELS TRIAL-AND-ERROR APPROACH THAT TESTING REQUIRES

thank you

questions?

further reading

- Florea, R. (2016). Exploratory testing. [online] Available at: https://www.uio.no/studier/emner/matnat/ifi/INF3121/v15/lecture-09.pdf [Accessed 25 Apr. 2022].
- Sam, B.H. (2019). Mindmaps and Heuristics for testing. [online] Bug Hunter Sam. Available at: https://bughuntersam.com/mindmaps-and-heuristics-for-testing/ [Accessed 24 Apr. 2022].
- MoT. (n.d.). Software Testing Heuristics: Mind The Gap! [online] Available at: https://www.ministryoftesting.com/dojo/lessons/software-testing-heuristics-mind-the-gap [Accessed 24 Apr. 2022].
- Kirkham, P. (2020). More Ways to Test – Learning from Fairy Tales. [online] Atomic Spin. Available at: https://spin.atomicobject.com/2020/01/17/software-test-goldilocks/ [Accessed 25 Apr. 2022].
- TestProject (2021). Software Testing Heuristics. [online] TestProject. Available at: https://blog.testproject.io/2021/07/05/software-testing-heuristics/ [Accessed 24 Apr. 2022].
- Vasylyna, N. (2011). Exploratory Testing: Advantage&Disadvantage - QATestLab Blog. [online] Available at: https://blog.qatestlab.com/2011/02/23/exploratory-testing-advantages-and-disadvantages/ [Accessed 24 Apr. 2022].
- Test Heuristics Cheat Sheet Data Type Attacks & Web Tests Paths/Files Timeouts Time Difference between Machines Crossing Time Zones Leap Days Always Invalid Days. (n.d.). [online] Available at: https://testobsessed.com/wp-content/uploads/2011/04/testheuristicscheatsheetv1.pdf [Accessed 25 Apr. 2022].
- Wordpress.com. (2022). [online] Available at: https://findingdeefex.files.wordpress.com/2015/05/testingmnemonics1.jpg [Accessed 25 Apr. 2022].

