Annemette Clement

Test consultant at Capgemini
Cand.scient ICT and product development

Experience:
Systematic, DXC, Aarhus University, Randers Regnskov,
Volkswagen, Økolariet, Alexandra Instituttet

Advanced Test Analyst
Tmap Test Engineer
02 What is Exploratory Test

?
“SIMULTANEOUS TEST DESIGN, TEST EXECUTION AND LEARNING WITH AN EMPHASIS ON LEARNING.”

James Bach
“Exploratory Testing is simultaneously designing and executing tests to learn about the system, using your insights from the last experiment to inform the next”

Elisabeth Hendrickson
Experienced based test technique

- Exploratory
- Error guessing
- Checklist based
Explore your surroundings
Quality / Purpose of Exploratory Testing

Fast Feedback

Learning

Revealing the unexpected

Gaining confidence
Having FUN
What am I thinking of
How many questions did we need?

What was your approach?

Did you use your experience from playing this game before?
When we do exploratory test we base our test on our experience and competences.

Test specialists are more likely to succeed than people without test knowledge.
Questions help you frame your exploration. Use them to seek answers during your session. Ask questions if you’re unsure of something. Questions remove assumptions. Clarify your understanding with a question.

Exploratory testing is an experiment mindset. Channel your curiosity and amplify your investigative approach.
- Problem solving
- Estimating
- Finding risks
- Presenting
Stop the robot

Without touching it or breaking it
What did we learn?
Several approaches to Exploratory test
- Structured (charter)
- Session based / timebox
- Working in pairs
- Heurestics and Oracles
Explore *(target)*
With *(resources)*
To discover *(information)*
Example

Me when I start at a new client

Target: I have to find out how to park without getting a ticket

Ressources: Reading signs, Reception

Information: Parking rules
**Target:** What are you exploring? It could be a feature, a requirement, or a module.

**Resources:** What resources will you bring with you? Resources can be anything: a tool, a data set, a technique, a configuration, or perhaps an interdependent feature.

**Information:** What kind of information are you hoping to find? Are you characterizing the security, performance, reliability, capability, usability, or some other aspect of the system? Are you looking for consistency of design or violations of a standard?
**Target:** what are you exploring?

**Resources:** what will you use?

**Information:** what do you want to know?

---

**Explore** editing profiles

**With** injection attacks

**To discover** security vulnerabilities

**Explore** entering new customer info

**With** some/none/all heuristics

**To discover** the effect of various data entered
Focus on:

- Risk
- Unanswered questions
- Use what you found yesterday / earlier
Let's test Vernie with a charter

- Target – Vernie
- Ressources – You & Vernie
- Information – Find out how Vernie reacts to different colors
What did we learn
Sometimes you’re in too deep and your brain overthinks what it would like to achieve. Stop the timer! Go get some fresh air or a glass of water. Observe those thoughts with some distance, let them drift past and come back fresh. It’s incredible what a short break can do.
A Timebox

Time provides focus. A timeboxed session helps you stop, take a step back and get the most out of a debrief.

Recommended is 60 – 90 minutes per session. The more you capture the more likely you are to overwhelm your audience.

Combine a short exploration time with a clear goal and you’ve got one powerful exploration combo.
What are risks?
Did it help hearing how the others define a risk?

Learn from another tester - Try their exploratory testing approach.

We can learn so much from our colleagues

We see different things and have interpret differently
Working in pairs
“Is there a way to make exploratory testing just as accountable as pre-scripted testing without sacrificing the creativity of the exploratory approach?”

Jon and James Bach
Imagine that I want a piece of bead with jam. It is a new jar with jam – meaning it is closed really tight. In the first attempt I make I fail to open it 😞

What do I do next?

I check that I'm turning towards the left to loosen the lid and try again. Then I retrieve a towel to establish a better grip when twisting the lid of the jar. Finally, in some frustration, I go and locate a stronger person – who succeed in opening the jar.
Heuristics are simply experience-based techniques for problem solving, learning, and discovery.

Every tester will have their own set of heuristics that guide their testing every day. These are innate and developed through experience.

*FX Date & time*
Oracles are simply the principle or mechanism by which we recognise a problem.

“When the explorers of old set sail for uncharted waters, they did not set out unequipped. They knew the sun and the stars, and they carried tools such as compasses, sextants, and clocks, not only for navigation but also for mapmaking.”

Michael Bolton
Which heuristics do we have for Vernie?
It is not an easy question!
Moving to IoT and robots
Unlike exploratory testing, scripted testing requires thorough documentation to be executed.

It allows QA engineers to repeat test cases if necessary. At the same time, exploratory testing permits testers to learn while working
Moving to IoT and robots
We will have to start focusing our test differently

Even with IoT and Robots functionality is still important

**But** Non-functionals require more attention *(compared to traditional testing)*

**And** also other non-functionals *(than we were used to test before)*
At the end of the day, the success of any given piece of software will be determined by its users.

This is especially key when working with IoT software, as apps will likely be evaluated by their ability to not only communicate and interact with a wide array of devices, but their capacity to bring tangible value to the lives of end users.
Let's look at the two concepts for human actions that Harry Collins and Martin Kusch describes in their book *Tacit and Explicit Knowledge*.

**Mimeomorphic** and **polimorphic**

*Mimeomorphic actions* are actions that we want to do the same way every time, almost as though we were machines.

*Polimorphic actions* are naturally and appropriately variable, and are rooted in social and human interactions and goals.
To test the mimeomorphic working procedures we can use our coverage based test techniques which can be automated and done by machines.

Mimeomorphic activities tend to be easy to observe, so they tend to be easy to identify and to explicate. As a result, conversation, writing, and training in testing has tended to focus on artifacts, on documents, on procedures, and on things that can be automated—the mimeomorphic actions
Because we focus so much on mimeomorphic testing, creating testcases which can be repeated and also teach: this is how testing is done

Test automation becomes the way to go – it can solve all

And we credit automation tools and behavior driven development
Polimorphic actions are often based on tacit knowledge

“can only be executed successfully by a person who understands the social context”

We need to find out how the “thing” interfere in our social lives
Will become:

- paying close attention to customer experience-based testing
- Less functional testing more testing the IoT experience

Still be:

- Based on tester expertise
- On experiences
Always have a notepad at hand. Every conversation is an opportunity to explore and capture learnings. For successful testing note-taking is essential.
Vil du lære mere? Så tag kurset **Exploratory Test**

Andre interessante kurser inden for exploratory test

- Testdesign i praksis
- Practioner in Agile Quality (PAQ)
- TMap Suite Test Engineer
- ISTQB Advanced Test Analyst

Se mere her: [https://www.capgemini.dk/Kursuskalender/](https://www.capgemini.dk/Kursuskalender/)