

Exploration in testing



Lector: Viktor Slavchev

www.pragmatic.bg

E-mail: mr.slavchev@gmail.com

Twitter: [@TheTestingTroll](https://twitter.com/TheTestingTroll)

Blog: mrslavchev.com

2019





What is testing?

- Why do we test?

- To test means - “to evaluate the quality of a product or SUT by learning through experimentation and exploration, including questioning, modeling, observation, inference etc.”

RST, James Bach and Michael Bolton.

- To prove that the product doesn't work / is not yet ready to ship.

- To advocate/argue about quality. Not to assure it!

- To investigate the risks that a product holds and the eventual impact, if they happen.

- To provide information about the product, in general.

What is testing? continued



■ Software testing(definition by Cem Kaner in BBST foundation course):

- is an empirical
- technical
- investigation
- conducted to provide stakeholders
- with information
- about the quality
- of the product or the system under test



What is a tester?

WEBPLUS.INFO 

A tester is someone who knows
that things can be different.

Jerry Weinberg



Your opinion

**How do you imagine testing?
What action would you perform
during testing?**





What is scripted testing



Problems with scripted testing/test cases

- Artifact driven over performance driven
- Slow and non-flexible
- Can't respond to the needs of modern development
- Documentation is not the purpose of testing
- **This is not the way we test!**



How do we test?

The exact same way as kids explore the world:

- Experimenting
- Questioning
- Exploring
- Modeling
- Sensory evaluation
- Empirically
- etc



What is Exploratory testing?



“a style of software testing that emphasizes the personal freedom and responsibility of the individual tester to optimize the quality of his or her work by treating test design, test execution, test interpretation, and test-related learning as mutually supportive activities that continue in parallel throughout the project”

Bach and Bolton



Shorter version:

“Test design, test execution and learning mixed together in a mutually supportive / parallel way”
James Bach

- Every testing in its nature is exploratory.
- There can be no script if you don't explore the application first.



Demo ET

30 questions game

Rules:

I will write down a word on a piece of paper.

You have 30 (or the number of students) questions to ask me to guess it.

Constraints:

Only questions with answer yes/no are allowed.

Examples:

Is it a living creature? Is it blue? Is it fish? etc.

Takeaways from the game



- Thinking of questions, before we start isn't a good idea
- Some questions might overlap
- Repeating questions is waste (!)
- Adapting to information from previous questions is critical for success
- Information overload - notes taking needed
- We have limited time for questioning.

Now replace “question” with “test”. This is what ET is about

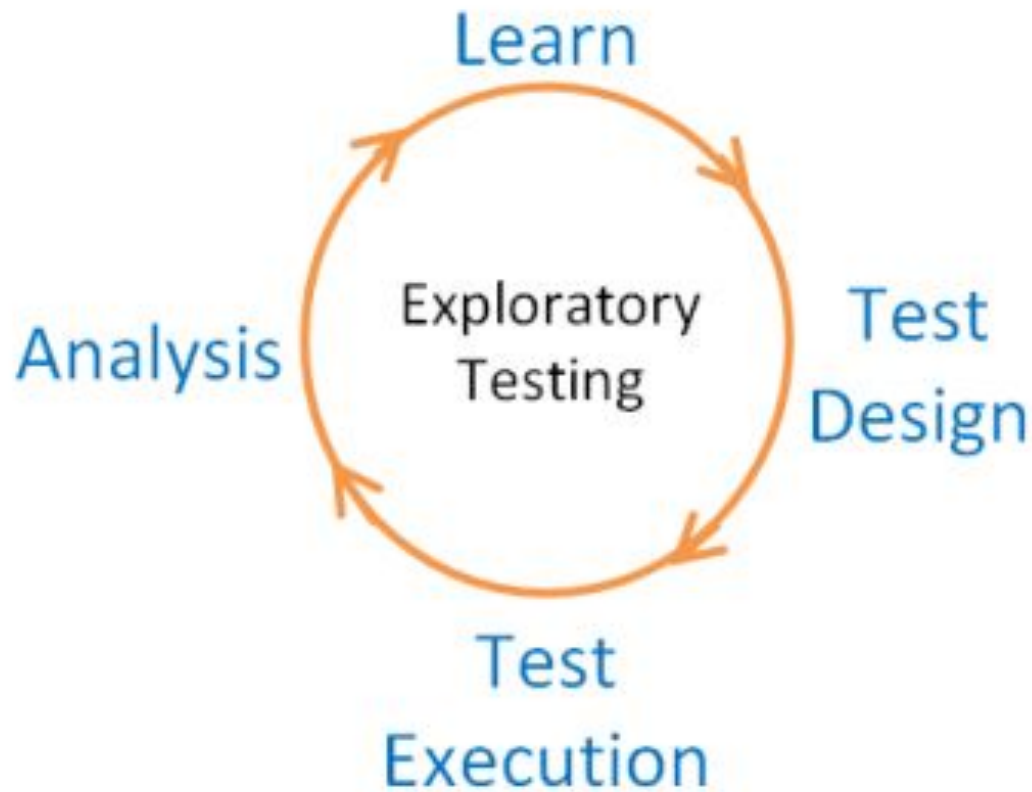


Takeaways for ET

- Thinking of **tests**, before we start isn't a good idea
- Some **tests** might overlap
- Repeating **tests** is waste (!)
- Adapting to information from previous **tests** is critical for success
- Information overload - notes taking needed
- We have limited time for **testing**.



ET diagram





What is not ET?

- Exploratory testing **is not**:
 - Test technique - it's an approach to testing
 - Testing without planning
 - Non-structured activity
 - Testing without documentation
 - Experience based only.
 - Excuse to perform crappy testing, omit documenting what we do or find an excuse if we are lazy.
 - Ad-hoc testing

The formal informal testing spectrum



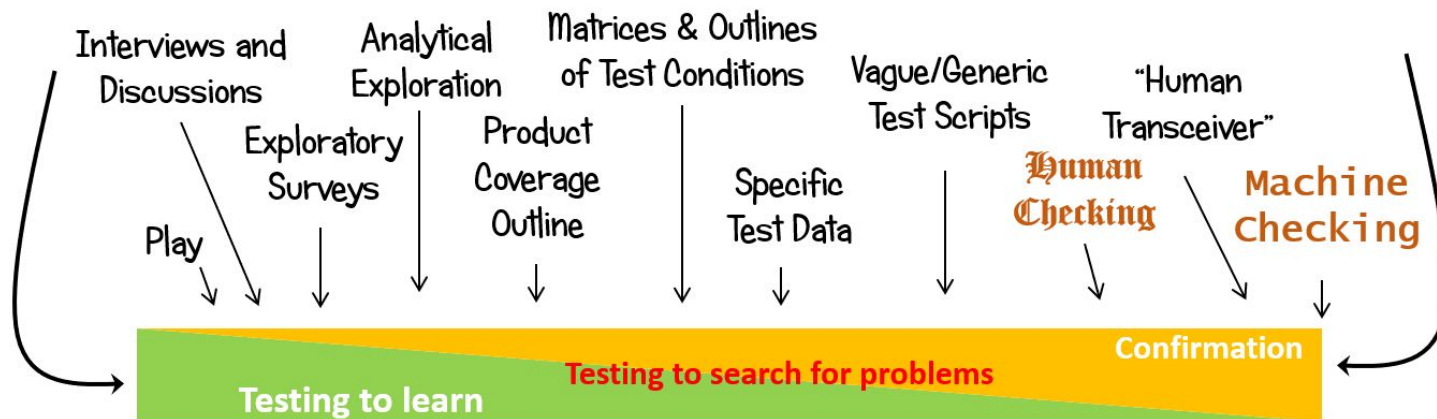
Credit to: [Breaking the Test Case Addiction](#)

INFORMAL

Not done in any specific way, nor to verify specific facts.

FORMAL

Done in a specific way, or to verify specific facts.



Loops of testing start with informal, exploratory work. If you want to do excellent formal testing (like automated checking), it must begin with excellent informal work.



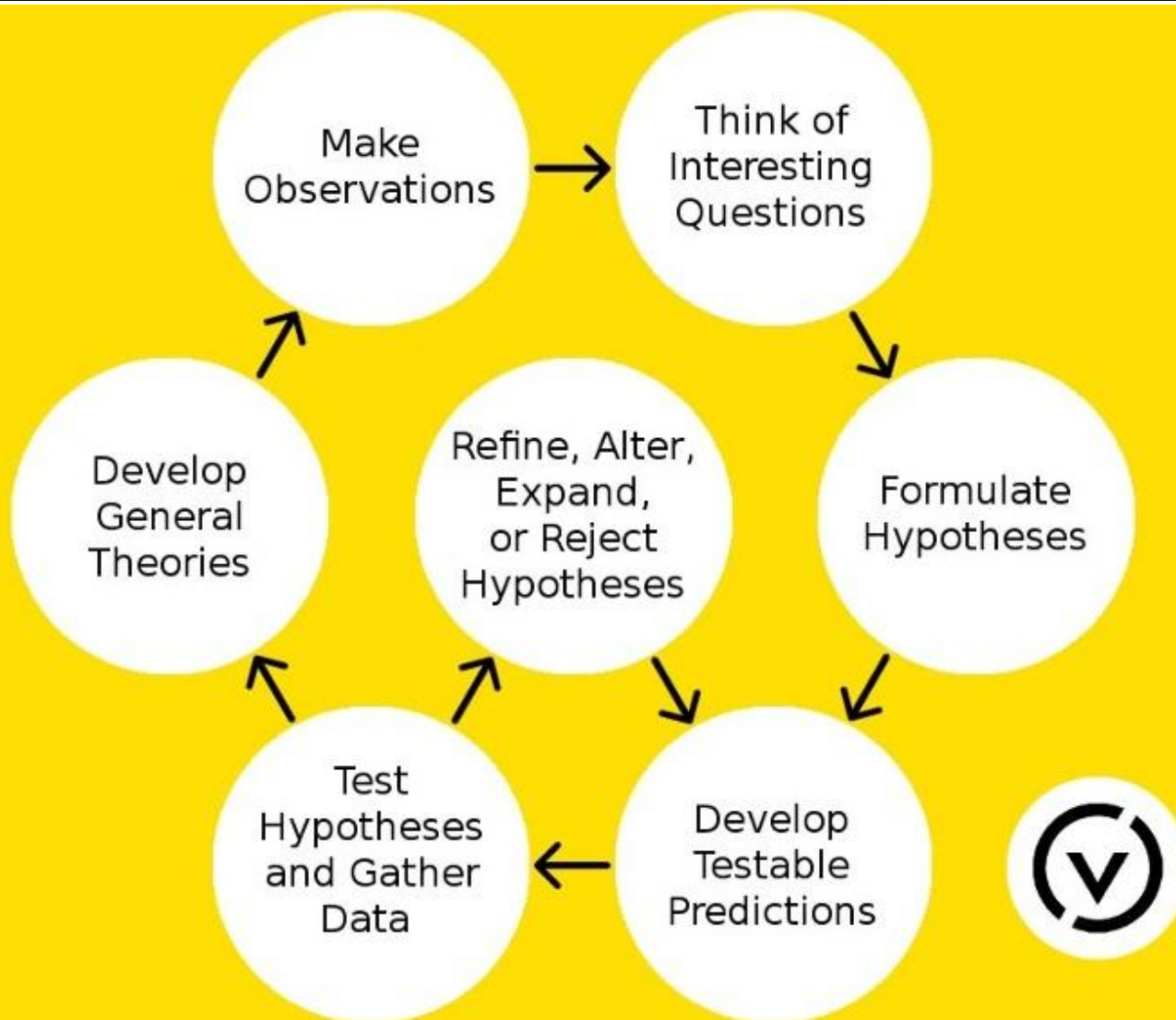
Theoretical task

How would you explore the cave?



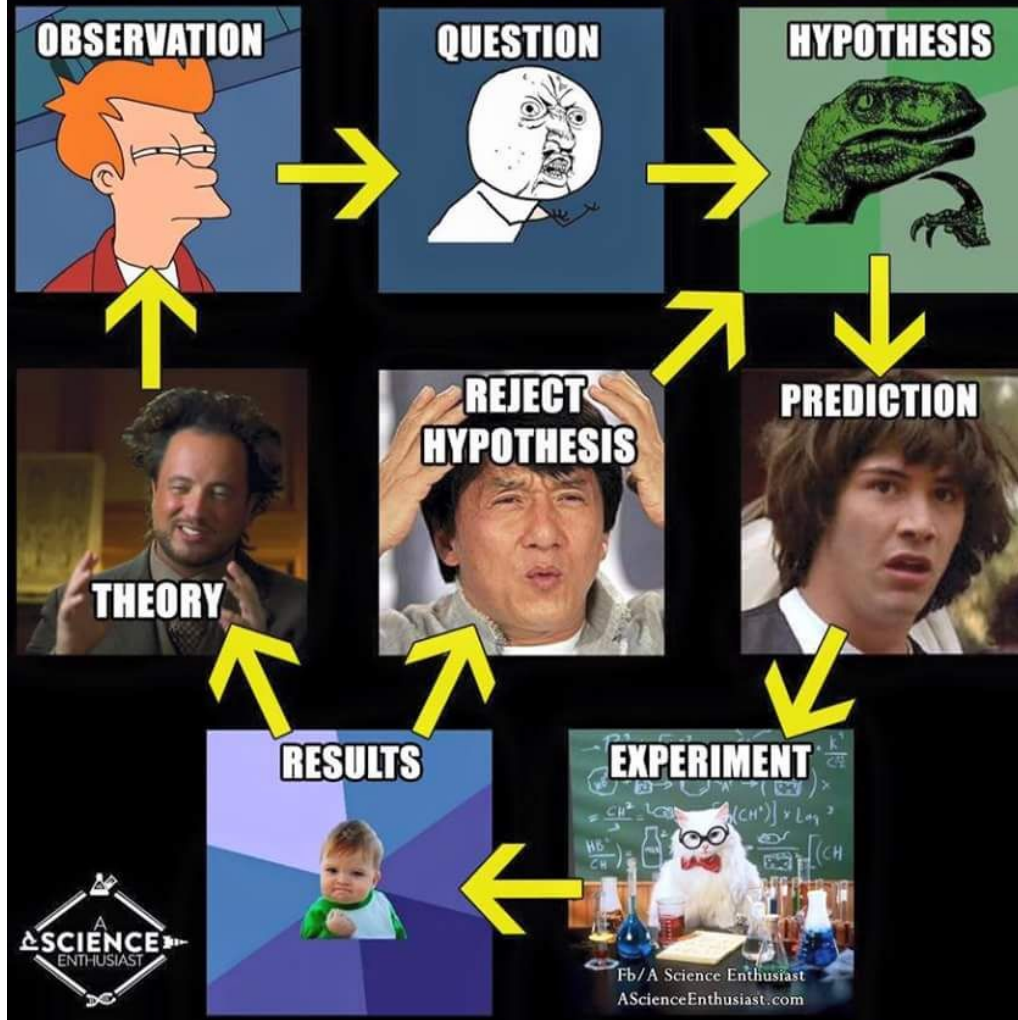


Science of testing



For those bored

The Scientific Method





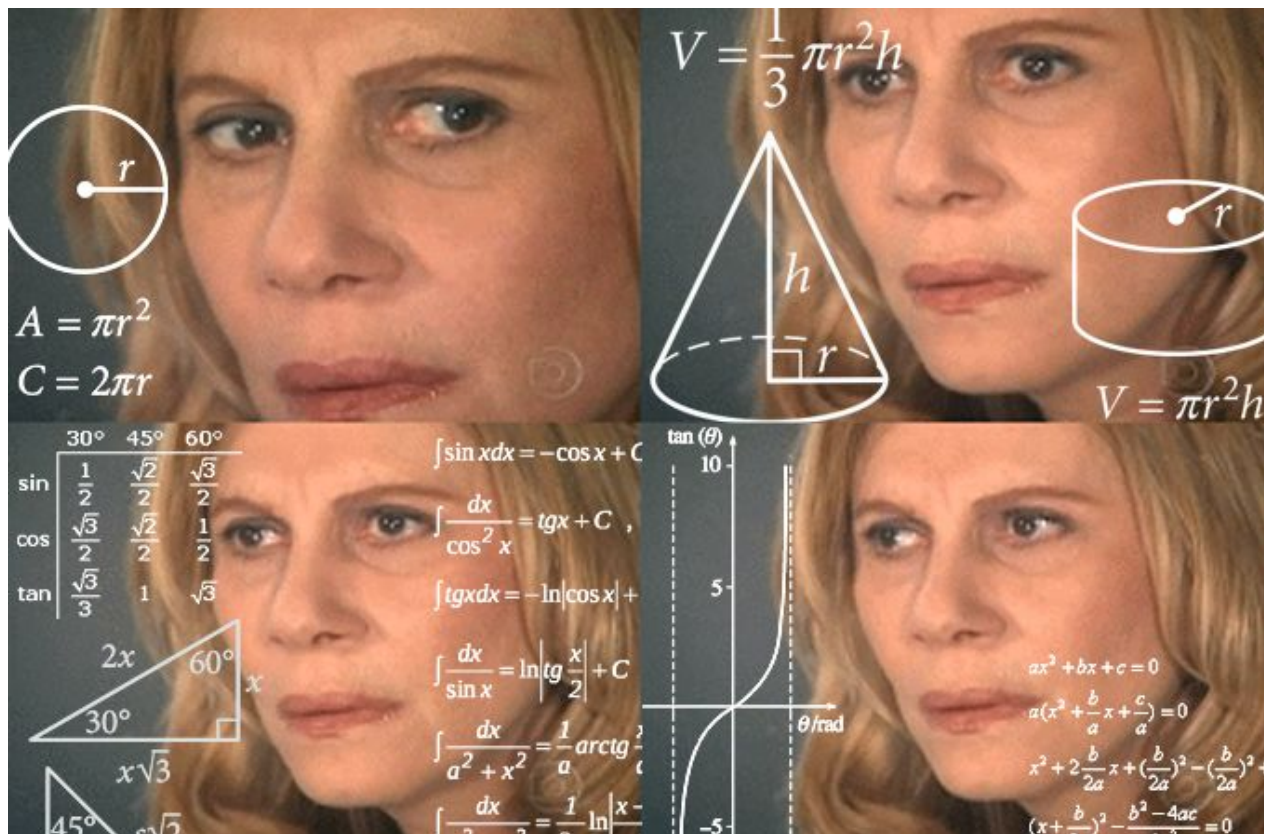
Exploration is also...

- Evolving work products - documents and artifacts
- Oracles - how to recognize a problem
- Context factors and information objectives
- Testing mission and strategy
- Coverage - what do we have to test?
- Test framing - linking our actions to the mission
- Heuristics and test ideas - how to perform testing
- Risks - think of a risk, perform a test for it
- Notes taking
- Expert bug reporting
- Expert testing reporting
- Stopping heuristics



or, in other words

Welcome to the world of software testing, where **things are not that simple**, as writing a script





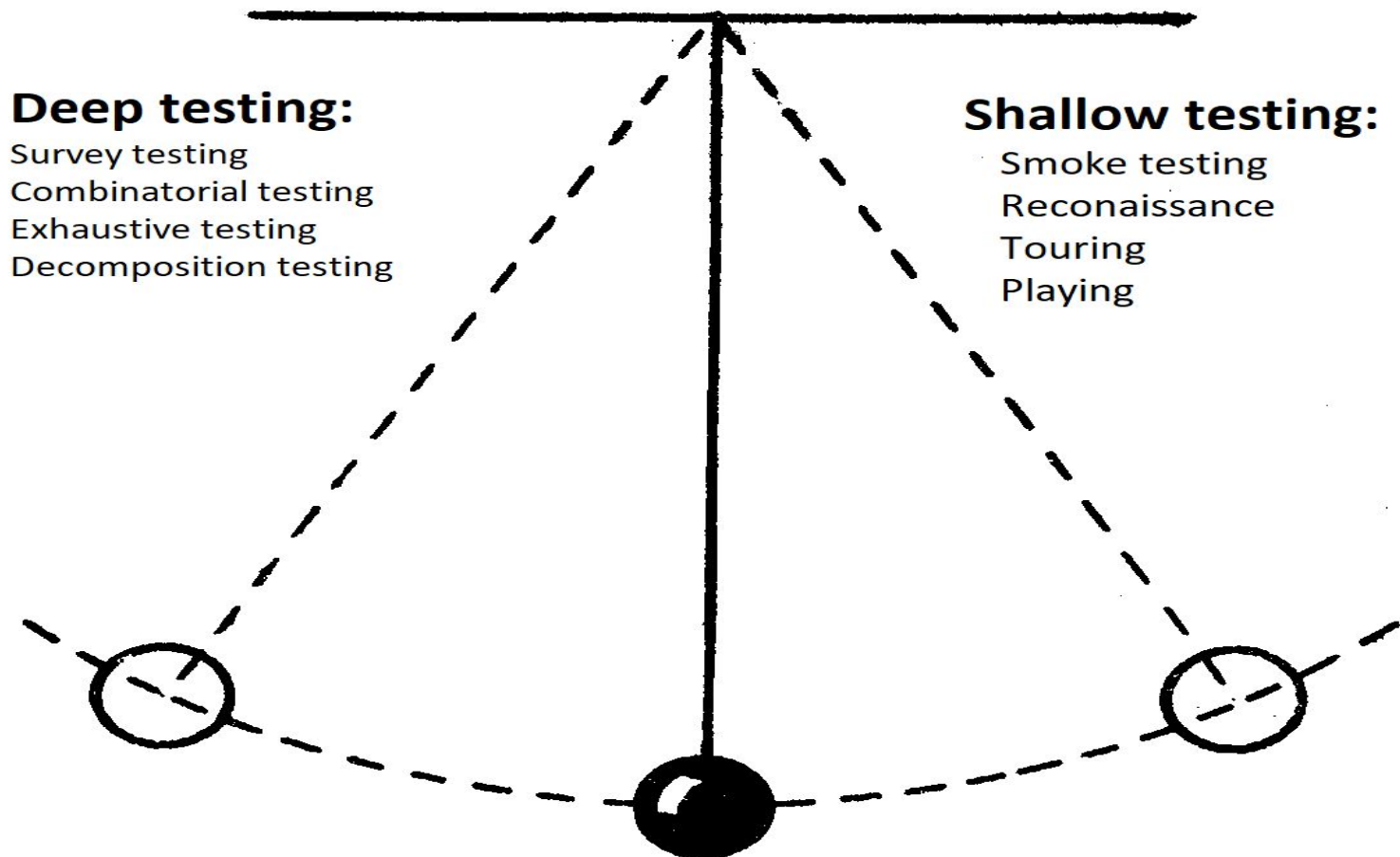
Types of sessions

- **Recon session** - comes from reconnaissance is type of testing that gives us general overview and information about what are we going to test.
- **Smoke testing** - very shallow, high speed, aims to prove main functionalities don't crash.
- **Survey testing** - more profound than recon, aims to give us information how the product works and identify interesting test scenarios.
- **Deep testing** - the process of directed testing against specific functionality or feature and its logic with the intend to expose problems and risks.



Testing pendulum

Taken by [Katrina Clokie's book](#)



Important components of ET





Testing mission

- It's just like a game mission.
- Simon Sinek story of crossing a room.
- Testing mission can vary, depending on the context:
 - Finding bugs (Early phase of the project)
 - Finding all possible bugs (Late/code freeze phase)
 - Finding showstoppers (CI)
 - Research the possible cause of an issue (postmortem)
 - Provide information about the product



Variables

- “In testing a variable is anything that you can change or cause to be changed indirectly while operating the software”. E. Hendrickson
- Variables could visible - ex. the max number of items in order or the max length of a domain name.
- Could be invisible or not directly modifiable - the amount of files that a storage system can process or depth of nested sub-items that could be created.
- We test software by **manipulating these variables**.



Personas

- A persona is imaginary actor that we impersonate while testing our scenario.
- We try to explore different user profiles in order to add more realistic contexts to our testing.
- Examples:
 - A geeky teen
 - A software engineer
 - An old lady
 - Casual person that is using it just for fun
 - The hectic guy
 - The multitasker



Test ideas

- Definition of test idea - *idea that helps us generate tests. or Ideas that guides the direction of your testing*
- The essence of a test.
- Test ideas could be realized also as test cases, test scenarios or use cases, but they are sufficient attribute to testing.

Enter Your Email



Test ideas example:

How would you test an email field for validity:

Test idea	Test data
Valid email as format	mrslavchev@gmail.com
Valid email with "." in it	mr.slavchev@gmail.com
Invalid email without @	mrslavchev.gmail.com
Invalid TLD	mrslavchev@gmail.asdf

How to generate test ideas



- What do we know about the subject of testing?
- What are its parts?
- What experiments can we derive from what we know?
- Who will use this? How is it going to be used?
- What are the common uses?
- What are the extreme uses?
- What is not obvious about the product?
- etc.



If you ever feel stuck

If you ever feel stuck and you think you can perform some more tests, take a look at:

- [Heuristic cheat sheet](#)
- [You are not done, yet!](#)



How do we do ET?

Answer yourself the following questions:

- What am I given, what do I see?
- Do I understand what I see?
- What I need to do/ask in order to understand it?
- What are the components of what I see?
- Do these components form system(s)?
- Do these components have dependencies?
- How can I interact with it?
- How can I make it misbehave?
- What rules can I break?



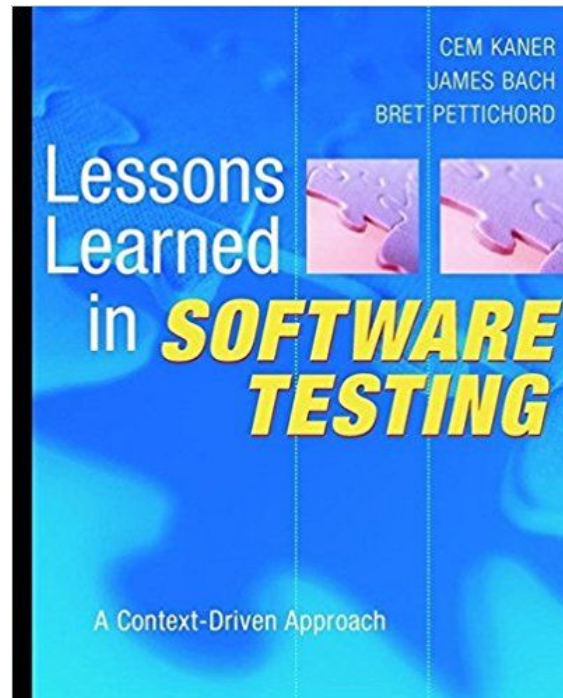
ET and test cases

- We can only use test cases to track coverage and focus on exploring
- We can follow a case or scenario and vary it
- We can use a test cases to totally break the script
- We can perform ET totally detached from test cases setting specific goals - test charter.

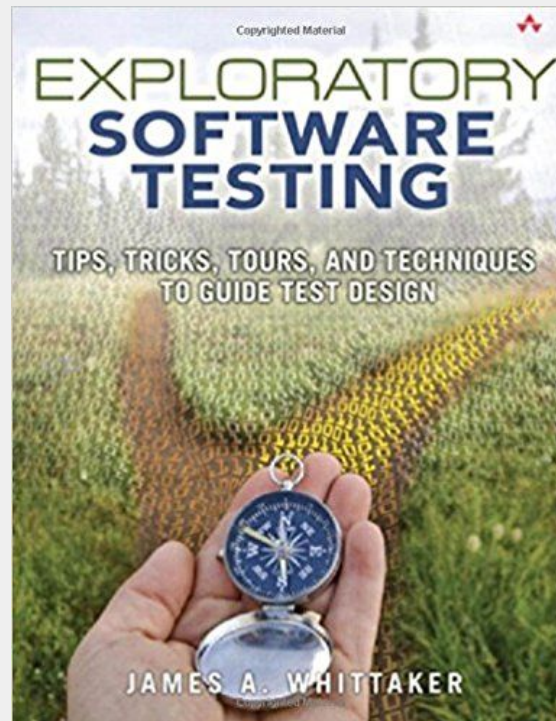


Must read

- Hindsight lessons about exploration - what is testing?
- Cem Kaner - a tutorial in exploratory testing
-



Books





Puzzle task

Go to: <https://goo.gl/fbqcYy>

Goals:

1. Explore the thing
2. Find system components - elements, relations, dependencies.
3. Find the system's purpose - what does it do?
4. Take notes of the experiments you perform.

Time to complete: 30 mins.