

A large, bright, blue-tinted full moon dominates the background. In the foreground, a person is silhouetted while riding a bicycle. A child is seated in a front-mounted child seat on the bike. The scene is set against a dark blue night sky. In the bottom right corner, the dark silhouette of a tree is visible.

E.T.

Tim Steffens

About me

- Tim Steffens / Tim.Steffens.extern@debeka.de
- IT Architect @ Red Hat
- Coding Enthusiast
- Great interest in QA
- Disclaimer: No Testing Pro
- But: Experience with ET in several Teams



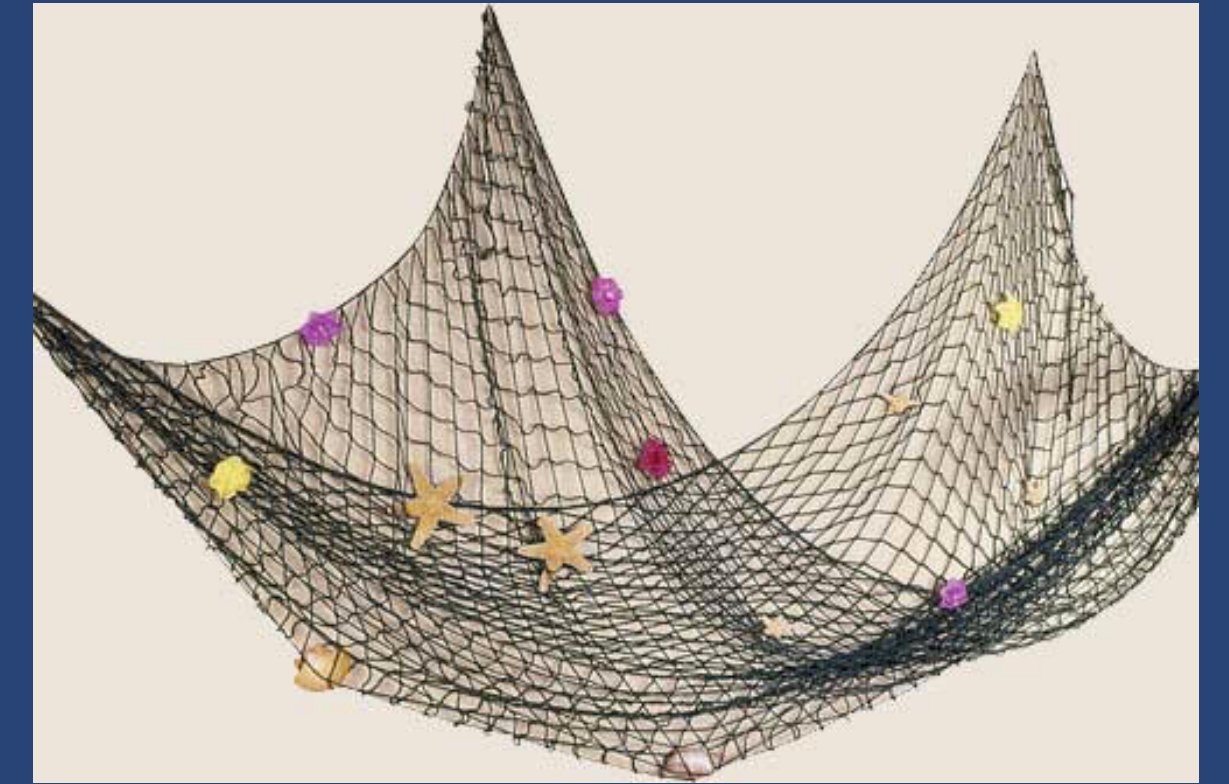
Please ...

- interrupt me if
 - I talk too slow / fast / silent / loud
 - you want me to speak in another language
 - you have question
 - I talk crap
- leave feedback!



Why Scripted Testing Is Not Enough

- Scripted (espec. automated) testing is important - e.g. for regression and possibly short feedback cycles
- But think of your scripted tests as threads of a net
- They catch whatever you thought of, but there are always things that are not caught
- Given the unlimited space of possibilities it is impossible to weave the net fine enough to catch every possibility
- This is where exploratory testing comes in handy!



What is E.T.?

1. an extra terrestrial life form
2. „Simultaneously designing and executing tests to learn about the system, using your insights from the last experiment to inform the next“

- *Elisabeth Hendrickson*



What is E.T.?

- addition to scripted testing, no replacement!
- done in repeating sessions of 1 to 2 hours
- e.g. once per sprint, release, story ...
- actions: Designing, Executing, Learning, Steering



What is E.T.?

Designing Tests

- identify interesting things to vary & interesting ways to vary them

Executing Tests

- execute as soon as you think of a test, no up-front design
- enables steering into most interesting direction
- think of tests as a small experiment



What is E.T.?

Learning

- discover how software operates
- look for subtle clues for bugs

Steering

- push harder on conditions not handled well
- steer towards the most interesting information to uncover



What's the Aim of ET?

- Effectively navigate through the space of possibilities to quickly identify unknown risks
- Focus your resources on the most important things
- Fit into an agile process
- Make use of intuition & associations



What can you test with ET?

Anything. It isn't limited to the front-end. You can test databases, the back-end, anything really.



How do you do this Exploratory Testing thing?

- It is recommended that you use **Charters** for your ET session
- Charters help you by giving you focus.

Explore *<target>*

With *<resources>*

To Discover *<information>*

target = feature, requirement (and in our case, app / website / parts of those)

resources = tool, data set, technique, configuration

information = security? performance? reliability? capability?

usability? design? (un)expected behaviour? (be creative!)



Charter Examples

Explore using the TNT app
with a bad connection
to discover if the app still works as expected

Explore the basket api
with burp suite
to discover security issues

Explore the shopping list flyout
with a large number of line items
to discover if it still responsive enough



Charters

- Avoid making the charter too broad, or too narrow
- You can make charters in refinements, planning sessions or when you hear a discussion that makes you go like 'oh, there might be issues in this part of the application, let's explore that!'
- The charters are there to help you avoid turning ET into Monkey Testing



Timeboxing

- Timeboxing is extremely important
- set timebox to a maximum of 30 minutes
 - You probably cannot hold deep focus for longer than that
 - if you can, good for you! You can make the test sessions longer :-)
- 5 min charters —> 30 min testing —> 5 min debrief
- Repeat cycle if desired (and there's concentration left)



Debriefing

- discuss the defects you've found with the team (including PO) and decide which ones need to be solved
- Jonathan Bach uses the acronym **PROOF** to help structure his debriefing:
 - *Past*. What happened during the session?
 - *Results*. What was achieved during the session?
 - *Obstacles*. What got in the way of good testing?
 - *Outlook*. What still needs to be done?
 - *Feelings*. How does the tester feel about all this?



Document what you're doing

- **!!! Documentation: really important !!!**
- document what you are doing to make defects reproducible
- make sure you are able to show others how you have produced a defect
- Which version of the build are you testing? Against which environment?
What device, with which OS? Which browser?



Document what you're doing

- Tools you might find helpful
 - Screenshot / Screen Capture (e.g. Monosnap for Mac)
 - Audio Recorder
 - Scrivener
 - Text Editor



Focus

- A pitfall for beginning Exploratory testers is the **lack of focus**.
- **stick to your charter** in one testing session, stay focussed!
- **document unrelated things** you notice, but don't let them distract you!
- don't start debugging when you've found a defect in the testing session
- concentration is a limited source, so use it wisely



ET with the Team

- We all see things from a different perspective, have different assumptions, different approaches, different experiences ...
- the more different views you mix, the more complete the picture gets
- invite all team members (devs, POs, SMs) and all other stakeholders (UI/UX, Customers / Users, Customer Support, interested colleagues ...)
- each view is valuable, more eyes find more bugs
- everyone gets a better sense for how their software feels like
- good possibility to present your features as well



A typical ET session

1. explain newcomers what ET is (10 min)
2. create, prioritize & assign charters (10 - 15 min)
3. do the testing session (30 min)
4. debrief: discuss the defects and decide with PO which ones need to be solved (5-10 min)
5. optional: go to 2.



Prepare ET Session

- make sure **the thing can be tested**
 - environment stable, release deployed
 - people bring their computers and have all required tools installed
 - calendar entry & everyone who could be interested is invited
 - you have a suitable room / space
 - etc.



Sources

- Talk by *Maaïke Brinkhof* at *Agile Testing Days*
- Document: https://docs.google.com/document/d/1XmrqBjcAdm1k8TROzs8FhXK9X5X4Y6V_0FopTc5pLoY/edit?usp=sharing
- Slides: <https://drive.google.com/file/d/0By2IWBE0cdioVE5yTXpfQWFNVWM/view?usp=sharing>
- Book: „Explore It!“ by Elisabeth Hendrickson
- Personal experience



~ The End ~

- Slides:
 - <https://tmstff.github.io/talks/slides/et-1.0.1.pdf>
- Questions?
- Feedback?

