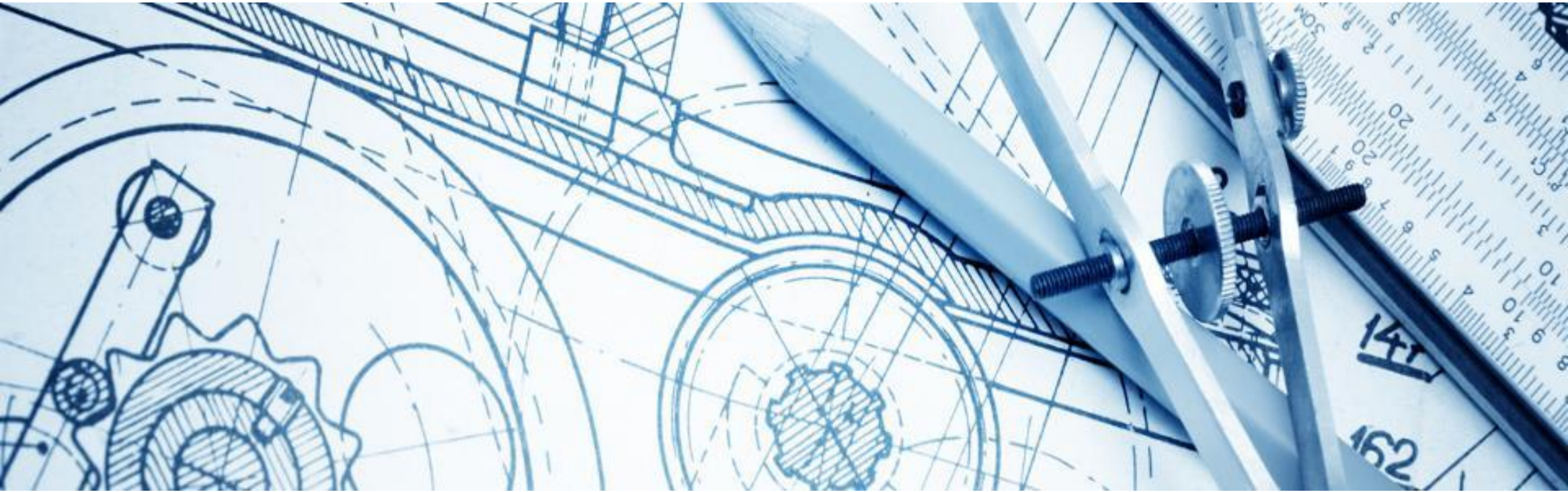
Testopsy:
Dissecting your Testing

Huib Schoots
huib.schoots@improveqs.nl
@huibschoots
www.huibschoots.nl
+31 (0)6 – 24 64 10 33

Improve
QUALITY SERVICES

European Testing Conference

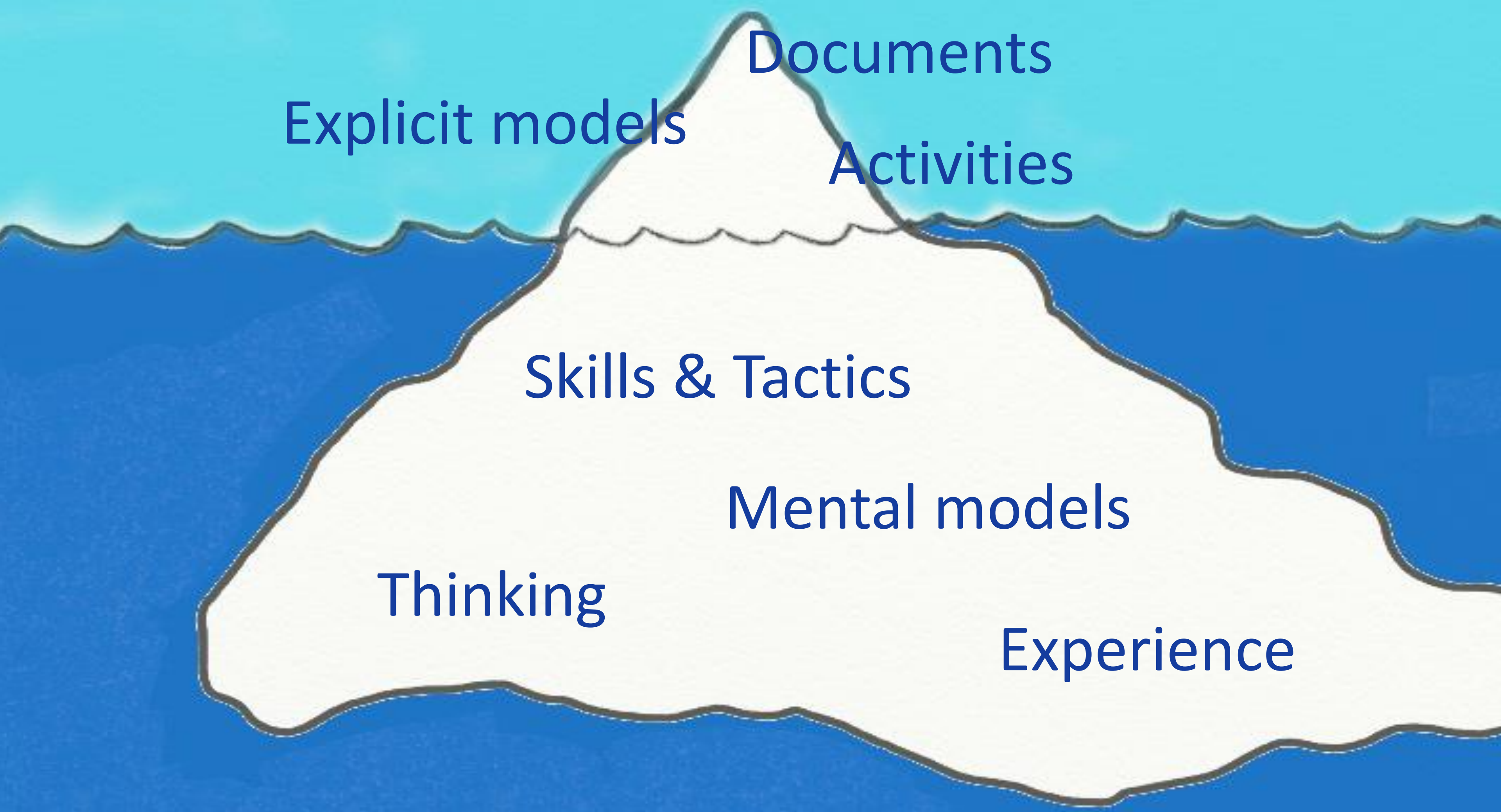
Introduction



Who am I?

- Context-driven software tester
- Rapid Software Testing teacher
- Scrum master & agile expert
- Humanist
- Curious & lifelong learner
- Passionate & energetic people lover
- Trainer, coach, writer, speaker & leader
- Books & Apple gadget collector
- Trombone player
- Gamer
- STAR WARS & LEGO freak
- Beer brewer
- Aspiring Magician

What do you do when you test?



Documents

Explicit models

Activities

Skills & Tactics

Mental models

Thinking

Experience

Unplanned… unanticipated… unspoken work…

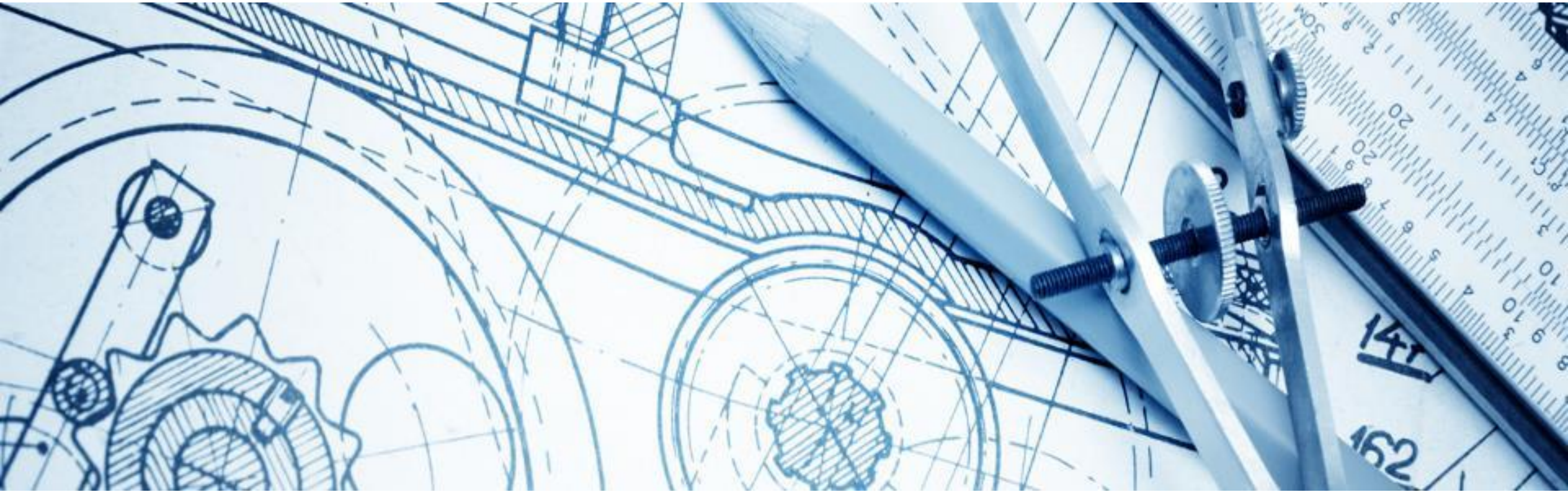
How do you learn?

If you don’t know what activities, tactics or skills you are using (or should use), how can you learn or train them?

Learn by experience:

- Concrete, challenging & achievable tasks
- Realistic application, processing & reflection
- Personal interpretation, exchange with others & constructive feedback
- Safe environment to experiment & make mistakes

Testopsies



What?

autopsy (n): a critical examination or dissection of a subject or work

testopsy (n): an autopsy of a testing session

Testopsy

A testopsy* is an examination of testing work, performed by watching a testing session in action and evaluating it.

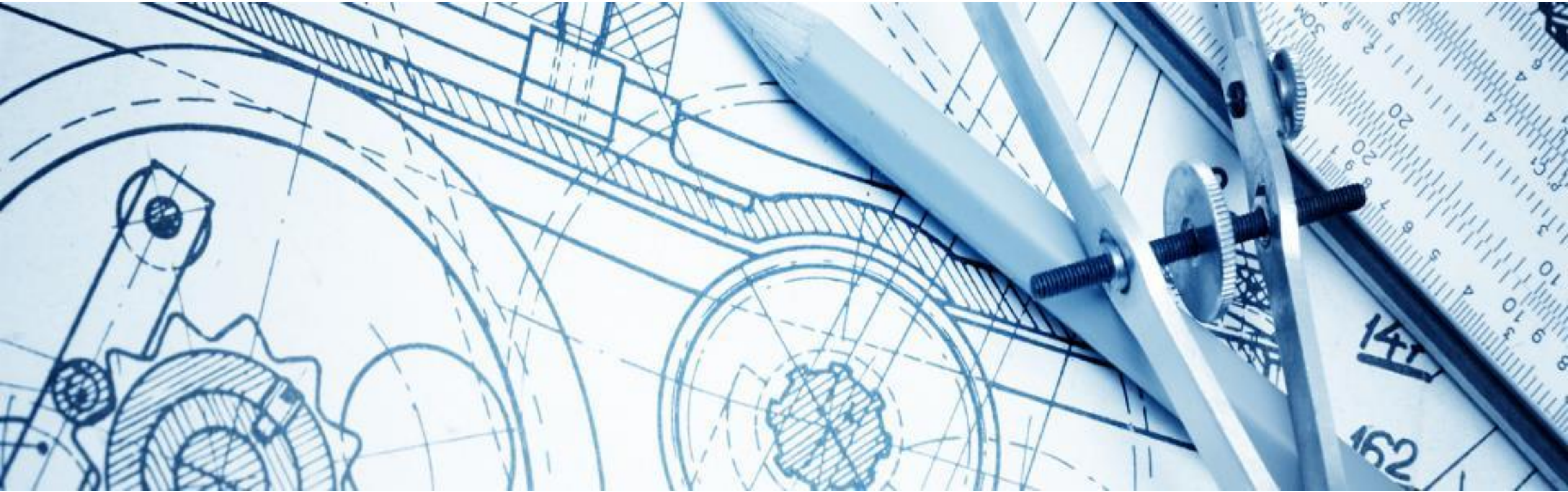
Testopsies can help in training, assessment, and developing testing skill for novices and experienced testers alike.

* The term testopsy is coined by James Bach

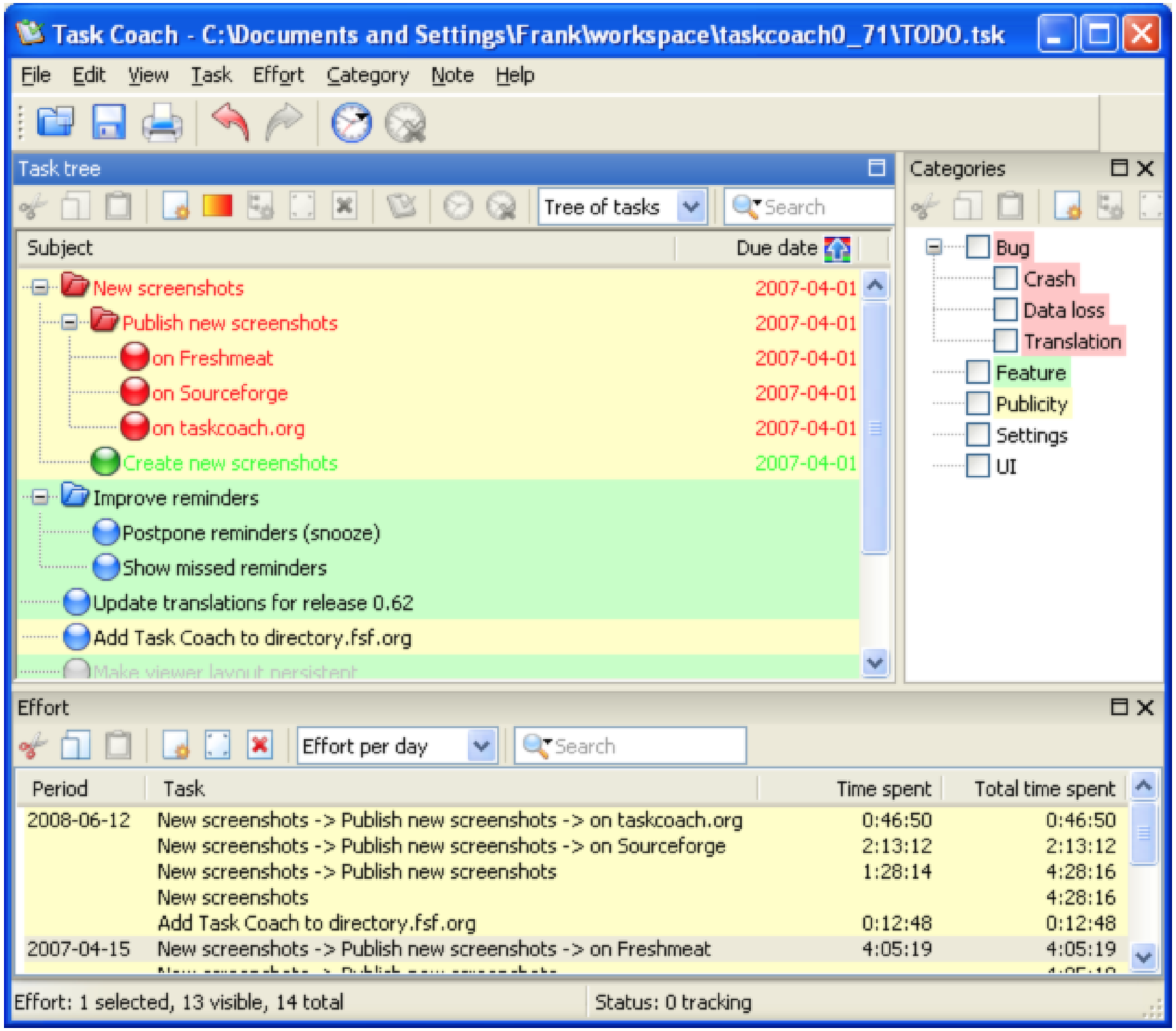
The Basic Idea

- Observe a testing session (your own or somebody else)
- Become aware of something interesting or complicated
- When you're aware of it, name it and make it explicit.
  - “Is there a name for that?”
- Having made it explicit, analyze it
  - When do you need to do it?
  - When do you need to avoid it?
  - Do we like it? Do we want it?
- Close the loop intentionality
  - Intend it
  - Do it
  - Explain it
  - Justify it

Let’s Test!



Task Coach


Task Coach - C:\Documents and Settings\Frank\workspace\taskcoach0_71\TODO.tsk
File Edit View Task Effort Category Note Help
Task tree
Tree of tasks
Search
Categories
Effort
Effort per day
Search

Subject	Due date
New screenshots	2007-04-01
Publish new screenshots	2007-04-01
on Freshmeat	2007-04-01
on Sourceforge	2007-04-01
on taskcoach.org	2007-04-01
Create new screenshots	2007-04-01
Improve reminders	
Postpone reminders (snooze)	
Show missed reminders	
Update translations for release 0.62	
Add Task Coach to directory.fsf.org	
Make viewer layout persistent	

- □ Bug
  - □ Crash
  - □ Data loss
  - □ Translation
- □ Feature
- □ Publicity
- □ Settings
- □ UI

Period	Task	Time spent	Total time spent
2008-06-12	New screenshots -> Publish new screenshots -> on taskcoach.org	0:46:50	0:46:50
	New screenshots -> Publish new screenshots -> on Sourceforge	2:13:12	2:13:12
	New screenshots -> Publish new screenshots	1:28:14	4:28:16
	New screenshots		4:28:16
	Add Task Coach to directory.fsf.org	0:12:48	0:12:48
2007-04-15	New screenshots -> Publish new screenshots -> on Freshmeat	4:05:19	4:05:19

Effort: 1 selected, 13 visible, 14 total

Status: 0 tracking

Product Coverage Outline

- is an artifact (a map, list, diagram, sketch, table…) that identifies the dimensions or elements of a product that might be relevant to testing it
- The Product Elements section of the Heuristic Test Strategy Model (SFDIPOT) provides a point of departure for creating a coverage outline

What I did (and why)

- I explored the application with the intention to create an overview which is helpful for further testing
- I created a mind map based on set of ideas: SFDIPOT heuristics from the Heuristic Test Strategy Model)
- I learned rapidly about the application by interacting with it and recording this in my mind map
- Creating a map has 2 goals:
  1. explore the application in a systematic way
  2. learn about useful aspects of the application

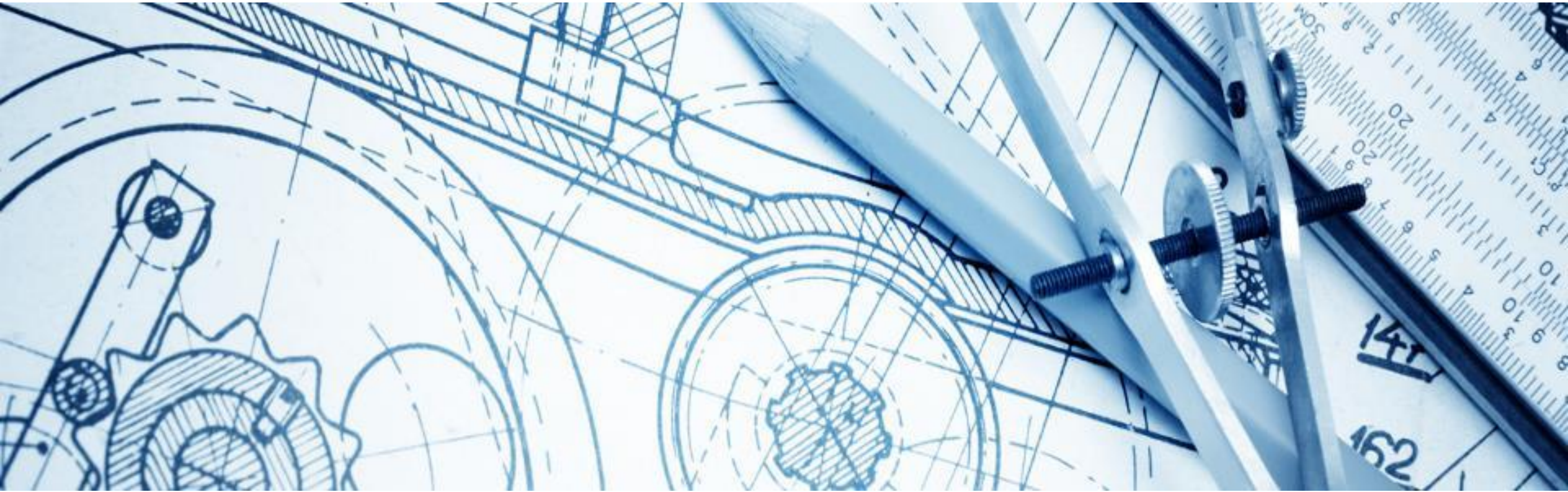
Watch the video

Product coverage outline

Huib Schoots
www.huibschoots.nl/blog

http://youtu.be/NUojNfDjIjw

Demo
Test Session



Preparation

- Work in pairs and create a list in 5 minutes of activities, tactics and skills you expect me to do when testing “Task Coach”

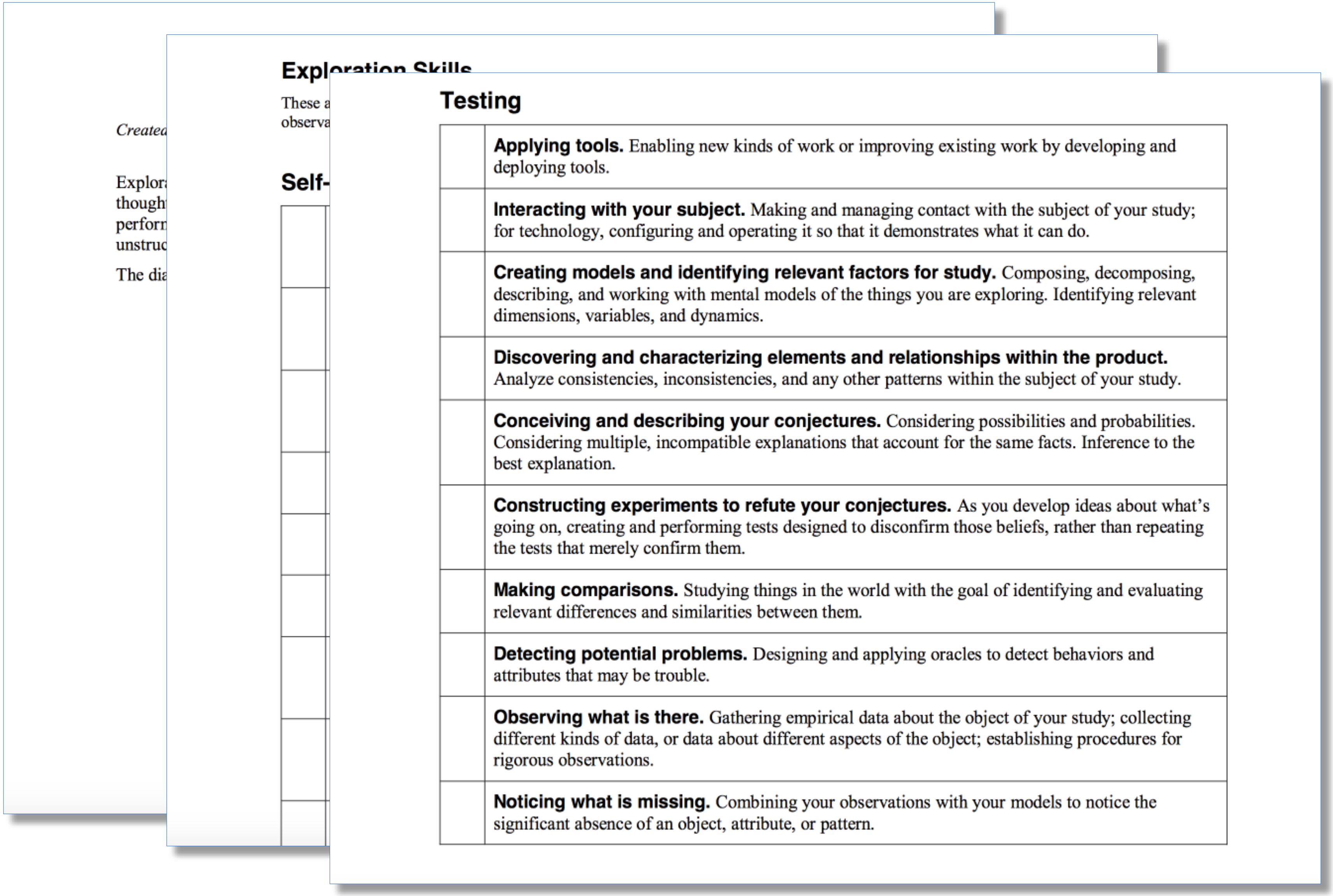
- When you see me test later:
  - Tick off the things on your list you see me do
  - Add new activities, tactics and skills you discover

Coding system

A coding system, a map of out the activities that testers perform and the skills and tactics they apply, helps a in observing and analyzing the work.

Use the coding system to guide observation of a testing session. Record what happens, and discuss the activity. Finally refine the coding system.

Example Coding system



Created

Explor
though
perforr
unstruc

The dia

Exploration Skills

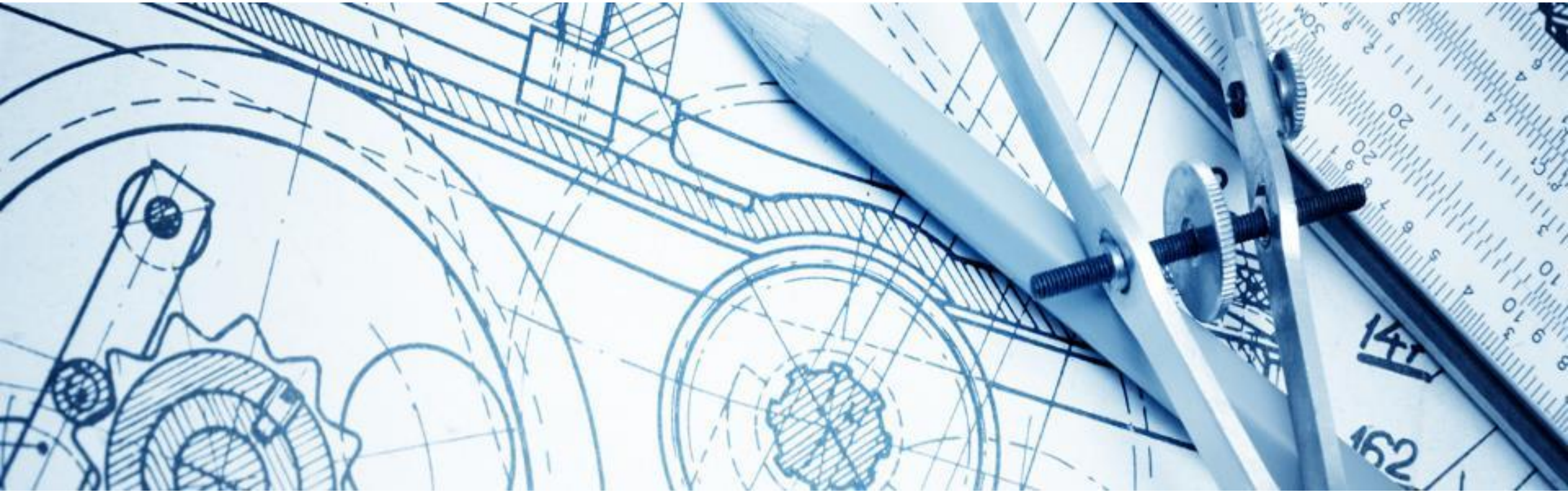
These a
observa

Self-

Testing

	Applying tools. Enabling new kinds of work or improving existing work by developing and deploying tools.
	Interacting with your subject. Making and managing contact with the subject of your study; for technology, configuring and operating it so that it demonstrates what it can do.
	Creating models and identifying relevant factors for study. Composing, decomposing, describing, and working with mental models of the things you are exploring. Identifying relevant dimensions, variables, and dynamics.
	Discovering and characterizing elements and relationships within the product. Analyze consistencies, inconsistencies, and any other patterns within the subject of your study.
	Conceiving and describing your conjectures. Considering possibilities and probabilities. Considering multiple, incompatible explanations that account for the same facts. Inference to the best explanation.
	Constructing experiments to refute your conjectures. As you develop ideas about what's going on, creating and performing tests designed to disconfirm those beliefs, rather than repeating the tests that merely confirm them.
	Making comparisons. Studying things in the world with the goal of identifying and evaluating relevant differences and similarities between them.
	Detecting potential problems. Designing and applying oracles to detect behaviors and attributes that may be trouble.
	Observing what is there. Gathering empirical data about the object of your study; collecting different kinds of data, or data about different aspects of the object; establishing procedures for rigorous observations.
	Noticing what is missing. Combining your observations with your models to notice the significant absence of an object, attribute, or pattern.

Now you!



How to do a Testopsy?

1. Record a session of your testing.
2. Go through the recording and note every single activity that you did. Put specific words to each activity.
3. Explain why you did what you did.

You can do this for a 10 minute session or a two-hour session.
I personally feel that very short sessions that are rich in product learning and test design are the most interesting to study.

QUESTIONS, YOU HAVE, HMM?

References Testopsy

- Exploratory Testing Skills & Dynamics (in RST Appendices) – http://www.satisfice.com/rst-appendices.pdf
- Skills mind map - http://goo.gl/VCQ0IN
- Podcast explaining Testopsy - http://www.qualitestgroup.com/The-Testing-Show/testopsies/
- Report of a Testopsy - http://patternsofproof.wordpress.com/2015/03/07/on-performing-an-autopsy/
- Reort of a Testopsies workshop - http://www.brendanconnolly.net/testopsies/

Other references

- Tacit and Explicit Knowledge and Exploratory Testing - http://steveo1967.blogspot.nl/2013/06/tacit-and-explicit-knowledge-and.html
- Shapes of Actions - http://www.developsense.com/blog/2011/12/shapes-of-actions/
- Testing Unexplained - http://goo.gl/nG0RZ6

References PCO

- Task Coach - http://taskcoach.org/
- Heuristic Test Strategy Model - http://www.satisfice.com/tools/htsm.pdf
- Rapid Software Testing - http://www.satisfice.com/info_rst.shtml
- Presentation on Test Coverage Outline - http://www.stickyminds.com/conference-presentation/test-coverage-outline-your-testing-road-map
- Experience report on using a Product Coverage Outline - http://prairietester.blogspot.nl/2013/09/monday-product-coverage-outlines.html
- Testing Story - http://www.developsense.com/blog/2012/02/braiding-the-stories/

Acknowledgements

- Some slides shown are taken from Rapid Software Testing and are used with permission.
- Rapid Software Testing is developed by James Bach and Michael Bolton. Also see: http://www.satisfice.com/info_rst.shtml

Rapid Software Testing (RST) is a mind-set and a skill-set focused on performing testing more quickly and less expensively while still completely fulfilling the mission of testing.

RAPID SOFTWARE TESTING

RST

A CONTEXT-DRIVEN METHODOLOGY

huib.schoots@improveqs.nl
@huibschoots
www.huibschoots.nl/blog

Improve
QUALITY SERVICES