

9th
UCAAT *User Conference on
Advanced Automated Testing*

Complementing GUI Testing Scripts with Smart Monkey Testing

Dr. Pekka Aho



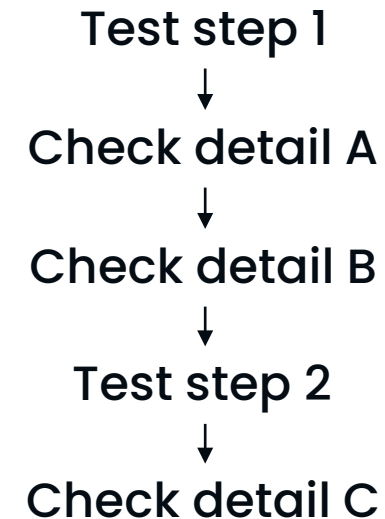
15 September 2022



Automated GUI testing scripts

Traditionally GUI testing scripts are...

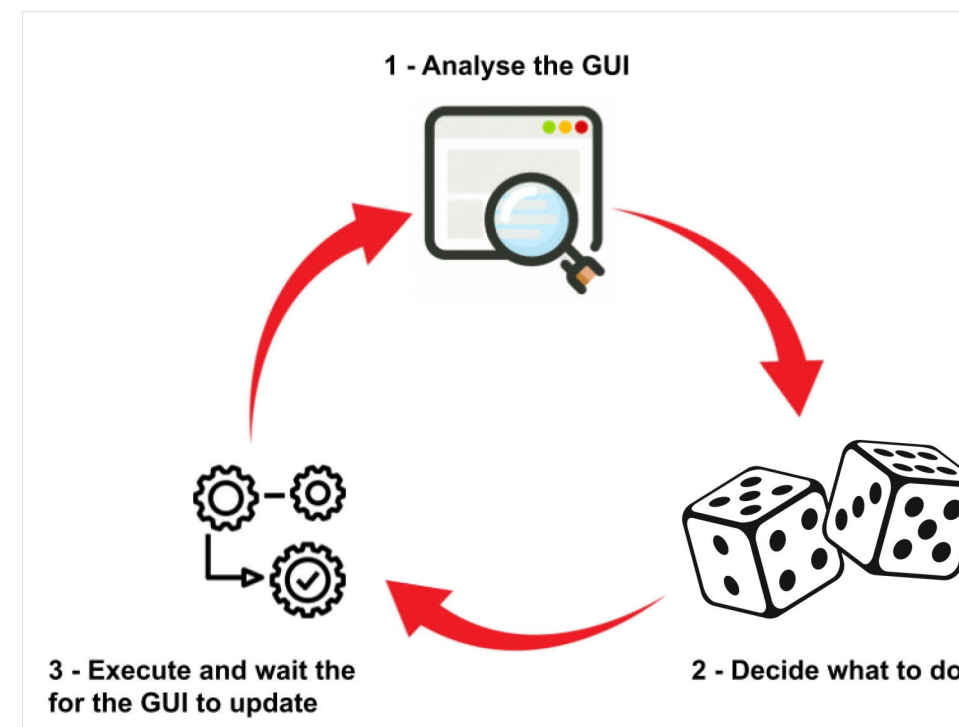
- designed manually
 - captured with C&R tools, or
 - manually written test scripts
- static until maintenance required
 - maintained manually
- for regression testing
 - checking the correctness of a selected set of functionality and properties



Monkey testing

In monkey testing, the tests are...

- generated during run-time
 - no test scripts to maintain
- dynamic, randomness often included
 - multiple executions could yield different results
- closer to robustness testing than functional testing
 - trying to find incorrectness anywhere in the system under testing



Monkey testing complements scripts

GUI testing scripts and monkey testing...

- find different kind of failures
- cover different parts of code

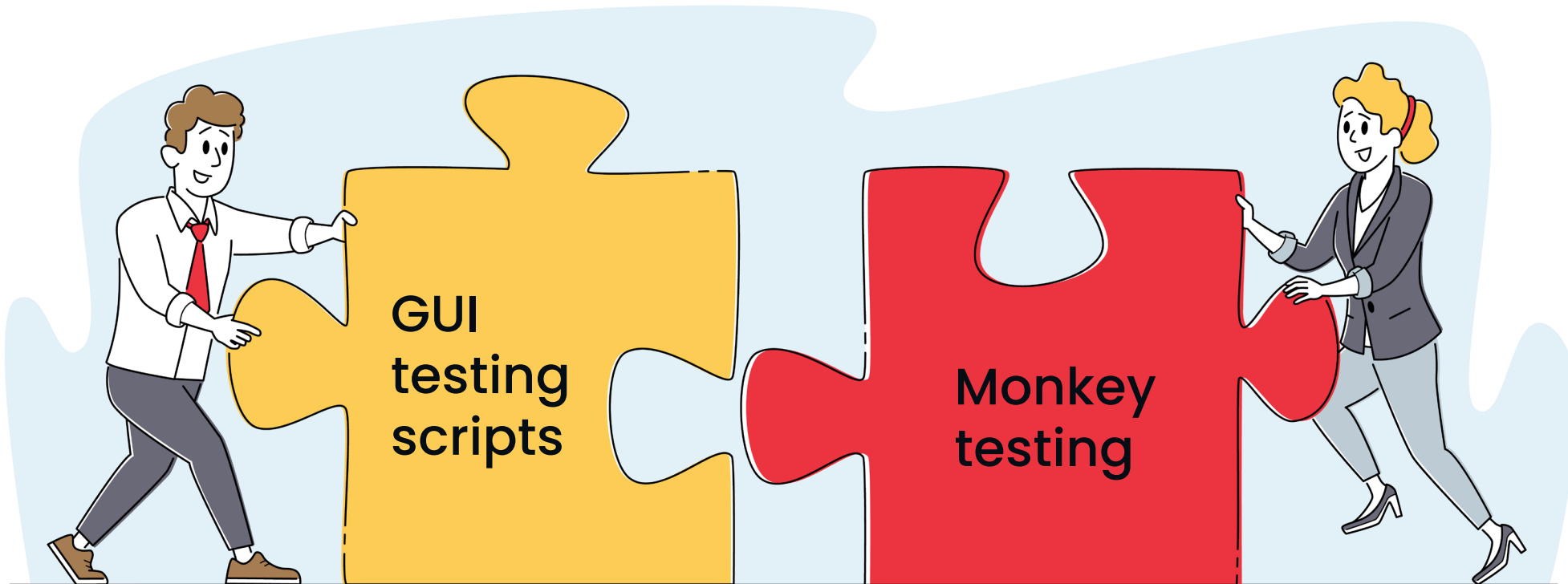


Tool	New bugs	Known bugs
Selenium	2 high	1 high
TESTAR	2 high 4 low	-



Tool	% IC	% LC	% MC
Espresso	43,9	43,4	45,9
TESTAR	41,0	40,7	40,8
combined	52,3	52,1	52,3

Monkey testing complements scripts



Random monkey testing is effective



Case study: Cap Gemini/ProRail

- Academia/industry cooperation
 - Open source TESTAR tool with random action selection and default test oracles
- Web application: Java based (JEE6) with 12,263 LOC
- System for managing the assignment of train platforms
- Existing test approach: manual with 100 test cases.
- After a change in the application: all 100 test cases were executed as regression tests

	TESTAR	Manual
Preparation	44 hour	43 hour
Testing	(51 hour)	6 hour
Post testing	5 hour	2 hour
Critical faults	4	0
Functional coverage	80%	73%

How to make the monkey smarter

The monkey testing tool can be enhanced with...

- a memory (model) to remember where it has been and what it has done
- pre-defined information about the SUT
 - how to reach in every part of the SUT (e.g., login)
 - SUT-specific test inputs or test oracles
- improved action selection algorithms
 - AI or ML to learn how to reach a certain goal
 - using the memory (model) to systematically explore
- self-coordinating parallel GUI exploration



Open source TESTAR tool

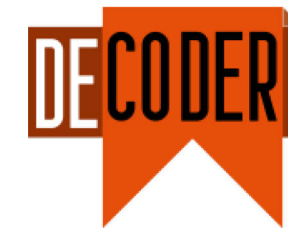
Smart monkey testing tool

- state model inference based on observed GUI behaviour
- various action selection algorithms (random, systematic, ML)
- supports SUT-specific information and configuration
- distributed GUI exploration with independent TESTAR agents and shared state model database

Open source

- https://github.com/TESTARtool/TESTAR_dev
- <https://testar.org/>

Many ongoing research projects with industrial collaboration



Any further questions?

More details from:

pekka.aho@ou.nl or <https://testar.org/>

