



EMBRACING NON-DETERMINISM IN TESTING

Andreas Ulrich, Stefan Dorsch

Non-determinism in testing – A Phantom Menace?

Martin Fowler (“loud-mouthed pundit on software development”) on Eradicating Non-Determinism in Tests

“Non-deterministic tests have two problems, firstly they are useless,
secondly they are a virulent infection that can completely ruin your entire test suite.”

<http://martinfowler.com/articles/nonDeterminism.html>

Jason Polites: The Tao of Testing v1.1 – A Field Manual for SW Engineers

Rule #2 Unit tests should be deterministic

“If a test relies on non-deterministic components in order to succeed
then failures in these components will cause a failure in the test.”

<http://jasonpolites.github.io/tao-of-testing/ch4-1.1.html>

Vojta Jina (Node.JS developer) on Make Your Tests Deterministic

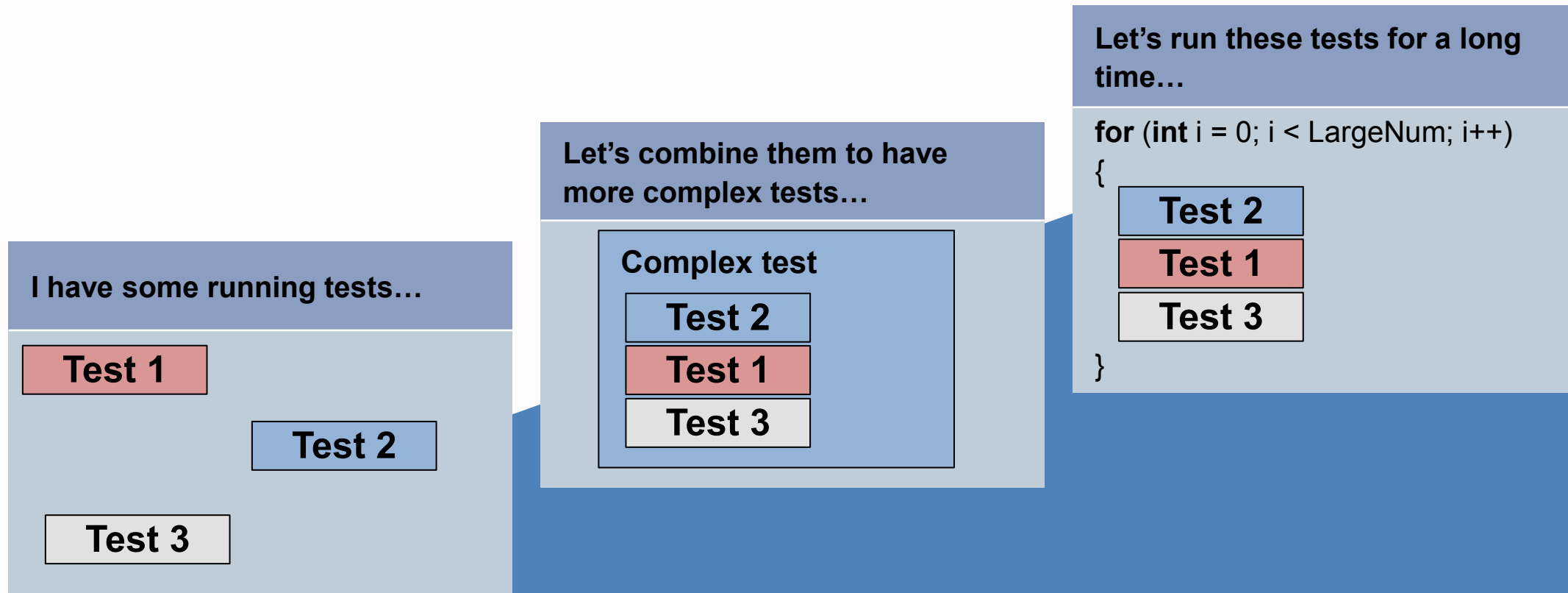
“In order to guarantee correct behavior of our code, we need to be sure that it
handles correctly all [...] situations. The best way to do that is by
simulating these situations in a fully controlled – a deterministic way.”

<https://howtonode.org/make-your-tests-deterministic>

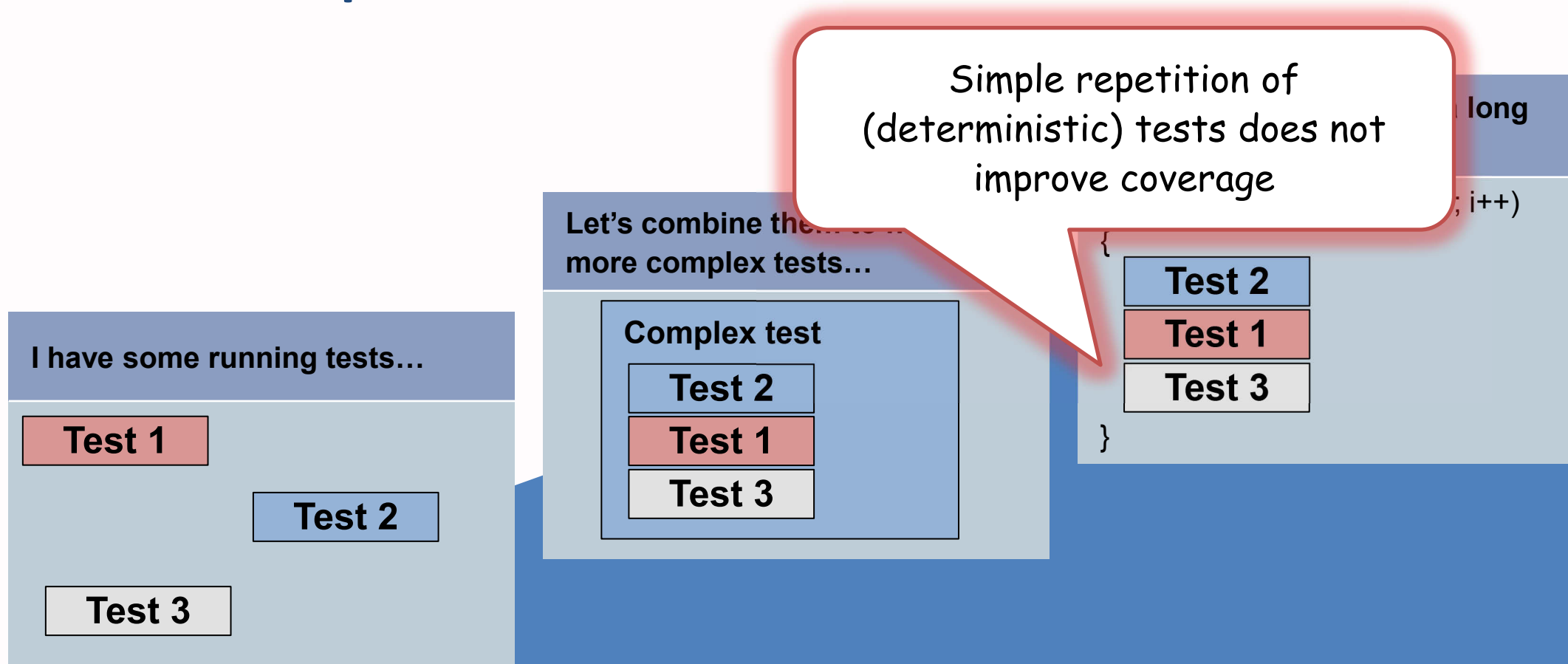
Non-determinism in testing – The light side

- **Non-determinism in testing is there already** – mostly unknowingly!
 - Test selection strategies follow their own rules, hidden in tools
 - Examples
 - Different tools on pair-wise test generation produce different output
 - In fact, all MBT tools produce different test cases for the same SUT
- **Lifting non-determinism at test execution level**
 - Why? – Improving test coverage!
 - Example: exploratory testing
 - How to avoid its negative side effects?
 - Repeatability, fault analysis

The quest for higher test coverage – The typical practice

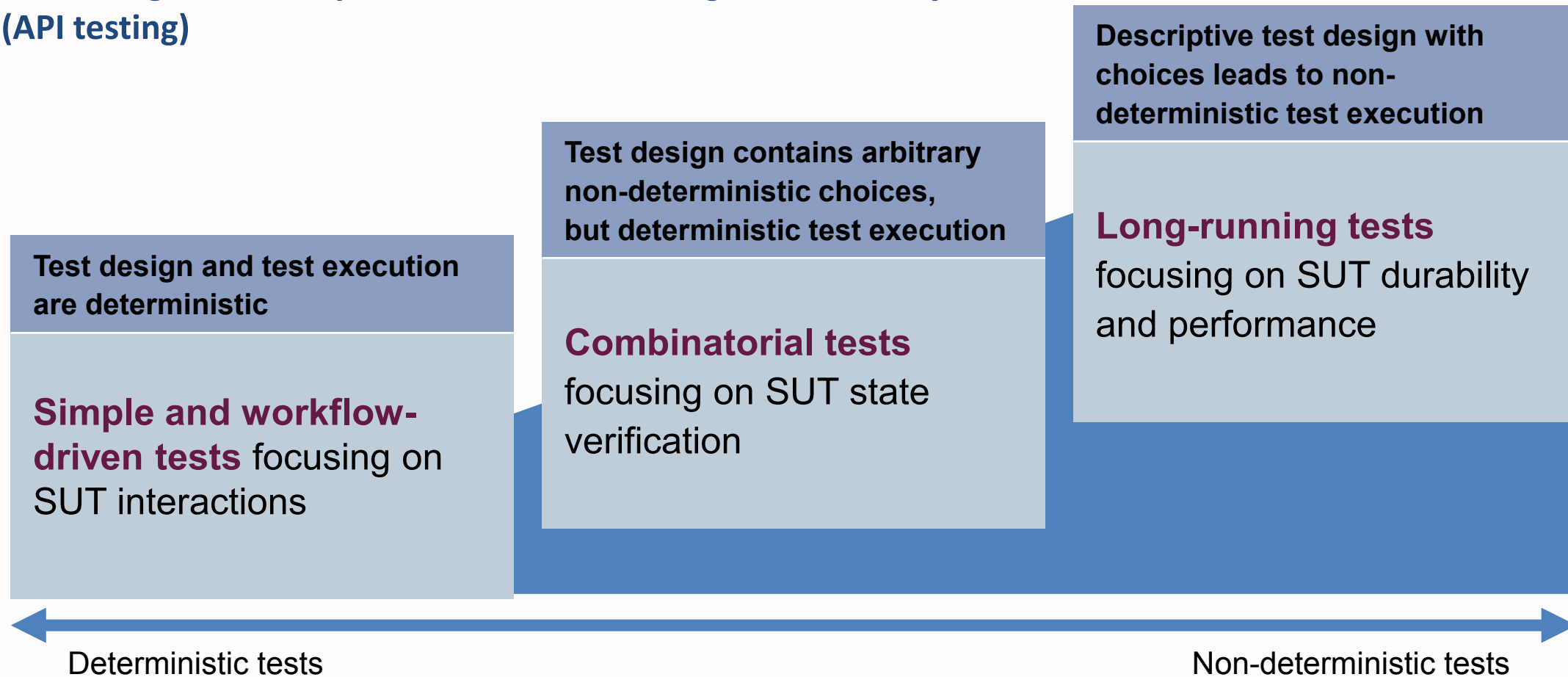


Limits of this practice



A proven strategy for non-deterministic testing

For testing software systems, black-box testing at the SUT façade
(API testing)

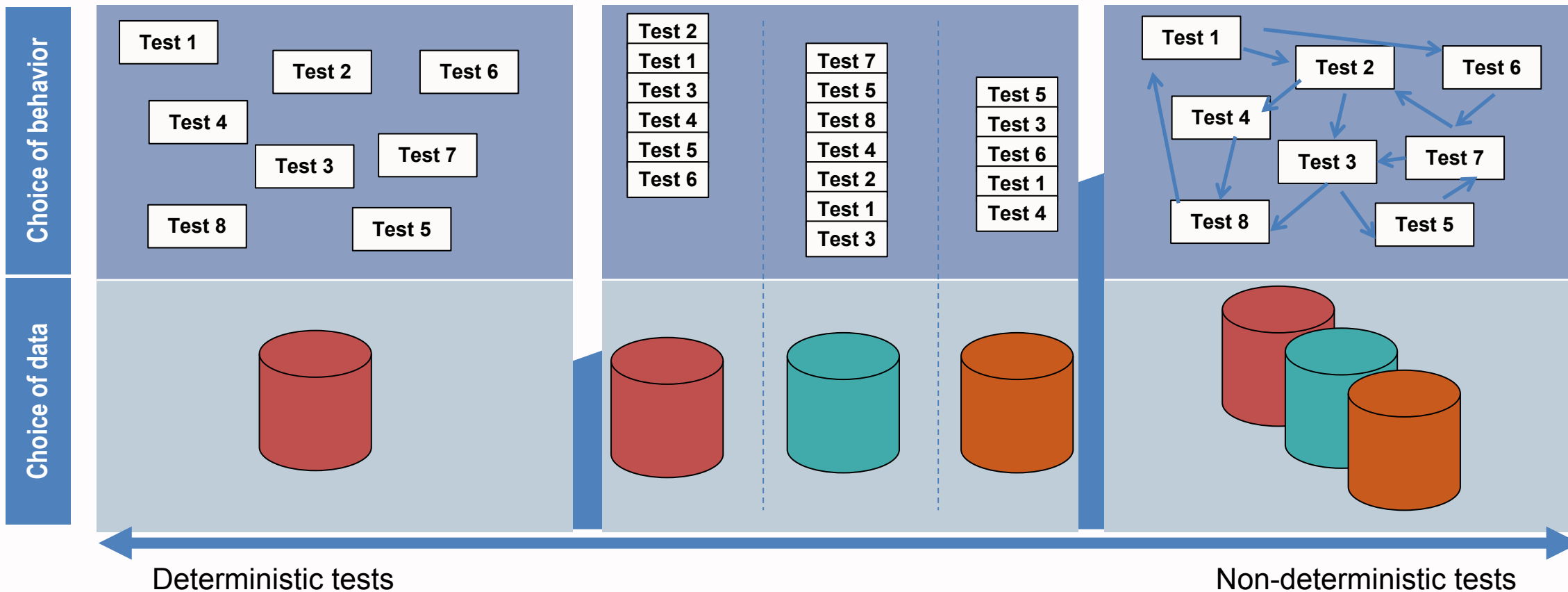


Step-wise test strategy visualized

Simple tests

Workflow & combinatorial tests

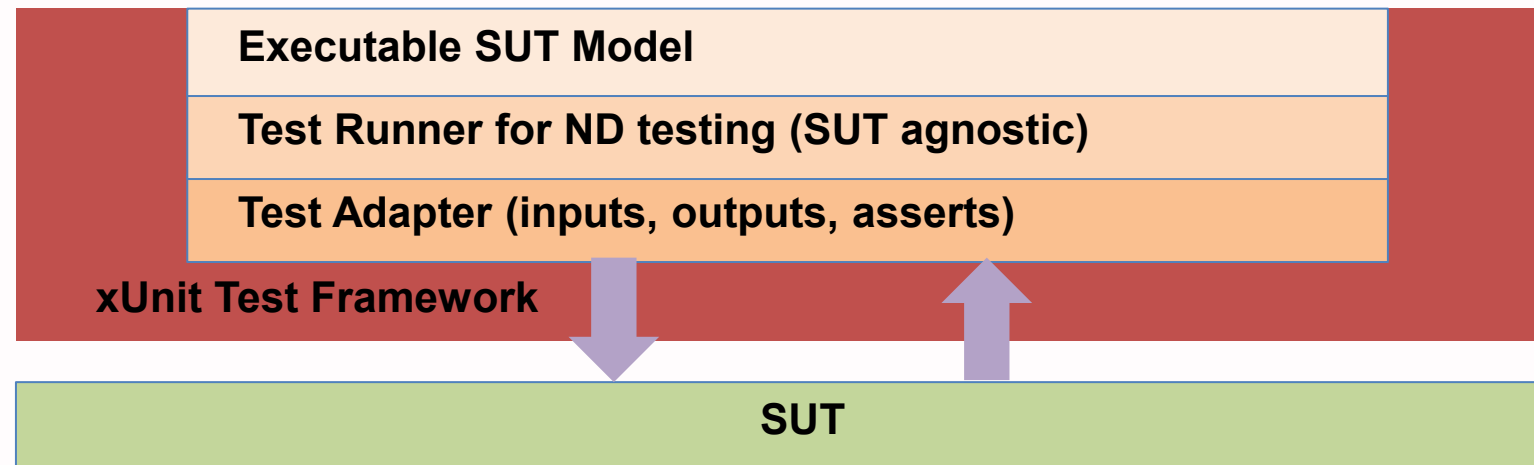
Long-running tests



Prerequisites for non-deterministic test execution

- **Modeling of SUT façade behavior**
 - Provide an abstract model as basis for test automation
- **Test automation layer for accessing SUT façade**
 - Build abstraction of concrete SUT behavior
 - Implement assertions for SUT responses here
- **Test runner tool for on-the-fly test execution supporting**
 - Non-deterministic choice of behavior (ND-Beh)
 - Non-deterministic choice of data (ND-Data)

Possible test architecture:



Available tools for non-deterministic testing

Tool	Type	Technology	Notation	ND-Beh	ND-Data
fMBT	OSS	Python	ASM	✓	—
GraphWalker	OSS	Java	EFSM	✓	—
JTorX	OSS	Java	IOTS	✓	✓
Modbat	OSS	Scala	EFSM	✓	✓
ModelJUnit	OSS	Java	FSM	✓	—
OSMO	OSS	Java	ASM	✓	—
PyModel	OSS	Python	ASM/FSM	✓	—
TestOptimal	Commercial	Java	EFSM	✓	✓

Legend:

ASM... Abstract State Machine
 (E)FSM... (Extended) Finite State Machine
 IOTS... Input/Output Transition System
 ND... Non-deterministic choice
 OSS... Open Source Software

Disclaimer: List created on best knowledge, not exhaustive.

Selection based on Zoltán Micskei's collection of MBT tools, http://mit.bme.hu/~micskeiz/pages/modelbased_testing.html

Putting non-deterministic testing into practice

Siemens TIA Portal

<http://www.industry.siemens.com/topics/global/en/tia-portal/>

The **engineering platform** for the Totally Integrated Automation of complex production processes



User Conference on
Advanced Automated Testing

SIEMENS

26-28/10/2016

© All rights reserved

Obtained results and achievements

Use of non-deterministic testing during integration test phase of TIA Portal development

- Non-deterministic tests introduced as **last test step**, i.e. nothing to start with!
 - Using our in-house test tool based on C#
 - Ensure that all known test scenarios are covered in deterministic test executions before
- Testing based on **non-deterministic choice of behavior and data** proved very successful!
 - Creates test scenarios not tested before, because they are hard/impossible to anticipate
 - Realistic faults found that would slip through testing otherwise
 - Tests are suitable in regression testing as part of Continuous Integration
- Managing **repeatability** of test results
 - Ensure that non-deterministic choices are made from a random number generator using a **seed** value
 - Non-determinism inside the SUT cannot be addressed by any approach
- **Fault analysis** from long test execution runs
 - Can be alleviated with proper test logging and Design for Testability measures

Contact details

Andreas Ulrich
Siemens AG, Corporate Technology
München / Germany
E-mail:
andreas.ulrich@siemens.com

Stefan Dorsch
Siemens AG, Digital Factory
Amberg / Germany
E-mail:
stefan.dorsch@siemens.com