SOFTWARE ENGINEERING RETHOUGHT

Human-Machine Mutualism over Parasitism

Towards Human-Machine Symbiosis

Robert Feldt,
(Robert.Feldt@gmail.com)

Emil Alégroth, PhD (Leader)
(emil.alegroth@bth.se)

SERL Sweden
LEADING SOFTWARE ENGINEERING

KK-stiftelsen

Background/introduction

(Emil Alégroth)

→ Engineer / Problem Solver / Researcher

- Associated Senior-lecturer, PhD (Tekn. Dr.) Software Engineering, M.Sc. Software Engineering, B.Sc. Computer science
- 4 years in industry (CEO, Developer, Consultant, Technical and educational Advisor)
- 7 years as a researcher
  - Automated verification and validation, GUI-based testing, technical debt in testing, model-based testing, educational research, human factors in software engineering.
  - Steering committee member of ITHS, EC utbildningar (Higher Vocational Education) and SAST Väst.

Software testing

Purpose: Verify and/or validate software

→ Manual

Step	Input	Expected output	Res.
1	Click on button X	Button X changes color	
2	Click on button Y	Button Y changes color	
..	...	...	...
N	Click on button Z	Button Z changes color	

Costly

Tedious

Error-prone

→ Automated

```
//Basic unit test that the constructor can instant
@Test
public void testConstructor(){
    assertNotEquals (null, main);
}
//Basic test of a method's output based on i
@Test
public void testAdd(){
    assertEquals(2, main.add(1, 1));
}
//Test instantion of custom class OperatorClass
@Test
public void testOperatorClass(){
    OperatorClass oc = new OperatorClass(1);
    assertEquals("ADD", oc.getType());
}
```

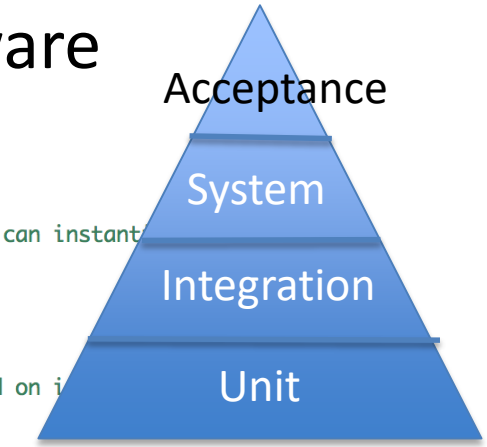


Figure: Test-pyramid

Prerequisites

Regression test

Technical difficulties

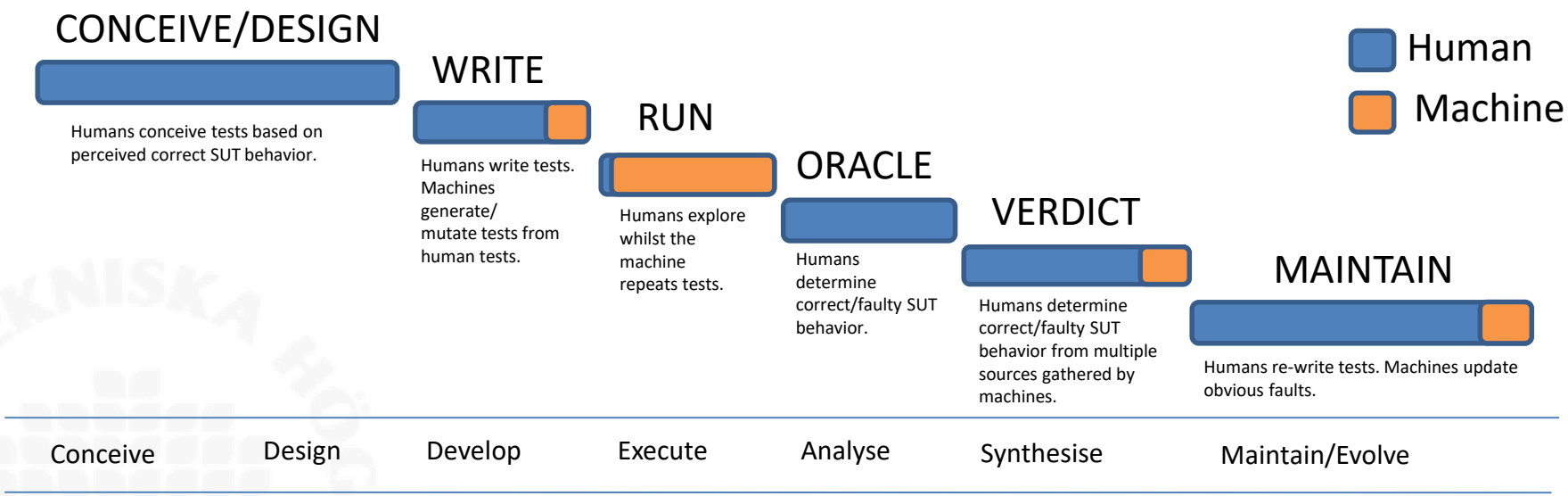
Symbiosis, Parasitism and Mutualism

- **Symbiosis** (from Greek συμβίωσις "living together", from σύν "together" and βίωσις "living")[2] is any type of a close and long-term biological interaction between two different biological organisms, be it mutualistic, commensalistic, or parasitic
- **parasitism** is a relationship between species, where one organism, the **parasite**, lives on or in another organism, the host, causing it some harm, and is adapted structurally to this way of life.
- **Mutualism** or **interspecific cooperation** is the way two organisms of different species exist in a relationship in which each individual benefits from the activity of the other.

Automated Software Testing is Parasitic

→ Humans give instructions
  → E.g. Write test code
→ Machine obeys
  → E.g. Run test code

→ Human responsible for all cognitive tasks



But what if…

→ …the machine learned from the user?
→ …gave suggestions and feedback?
→ …automatically tested new scenarios?

Human
Machine

CONCEIVE (with support)

Humans conceive tests but are supported by suggestions given by the machine learnt by:
- Test data generators
- Crowd observation
- Historical data
- User interaction

INSTRUCT

Parts of the test design is learnt. (E.g. neural net)

RECORD

Learnt behaviors are transformed into executable tests.

OBSERVE

Humans explore together with the machine and the machine repeats tests.

TEACH

Domain knowledge allow machine to determine test result correctness.

JUDGE

Analysis of large test-data sets enable easier decision making and more machine-based decisions.

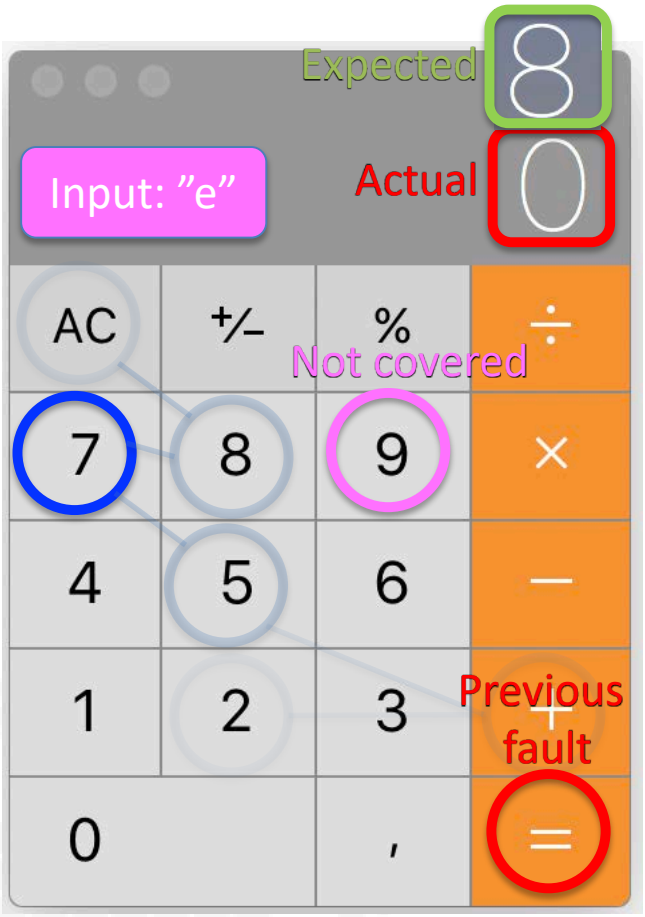
RE-INSTRUCT/ RE-RECORD

Learning enables the machine to make better suggestions how to maintain tests.

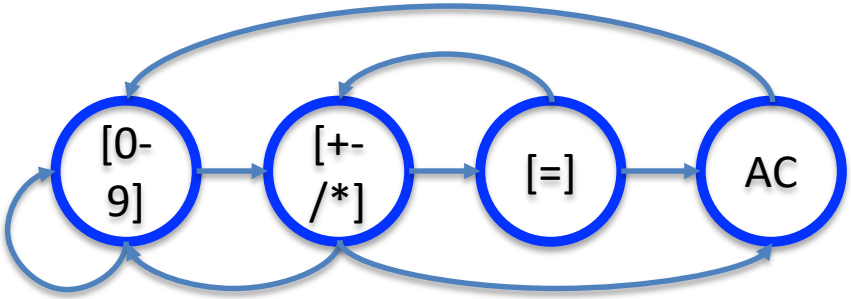
Conceive	Design	Develop	Execute	Analyse	Synthesise	Maintain/Evolve

Challenge: Communication!

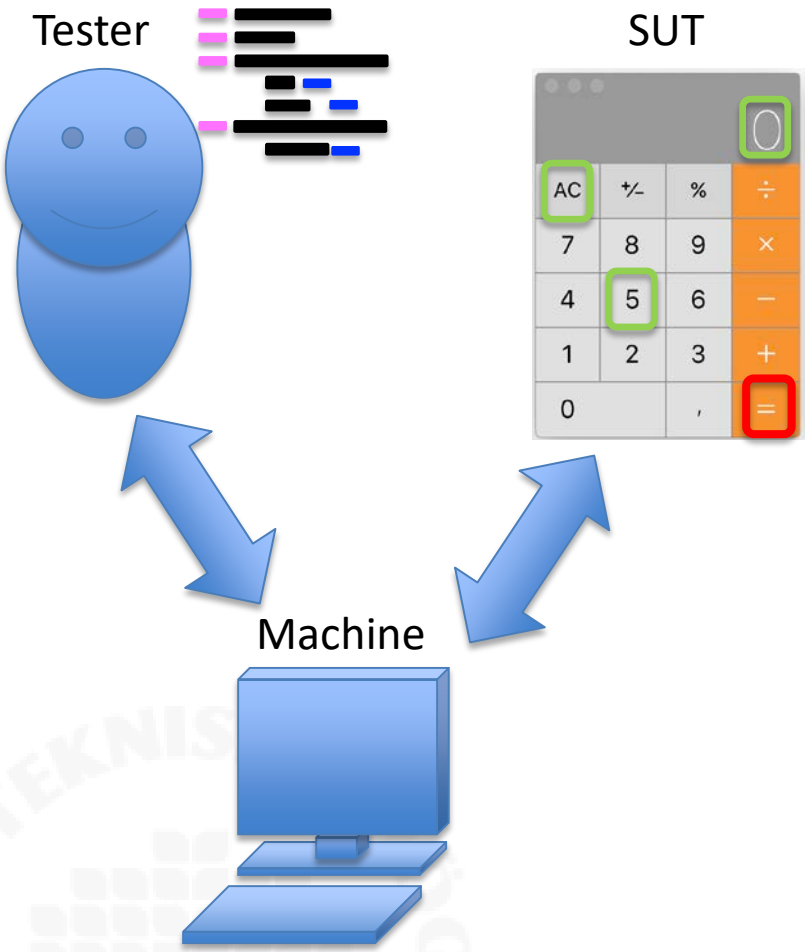
One possibility is: User-Interface Augmentation



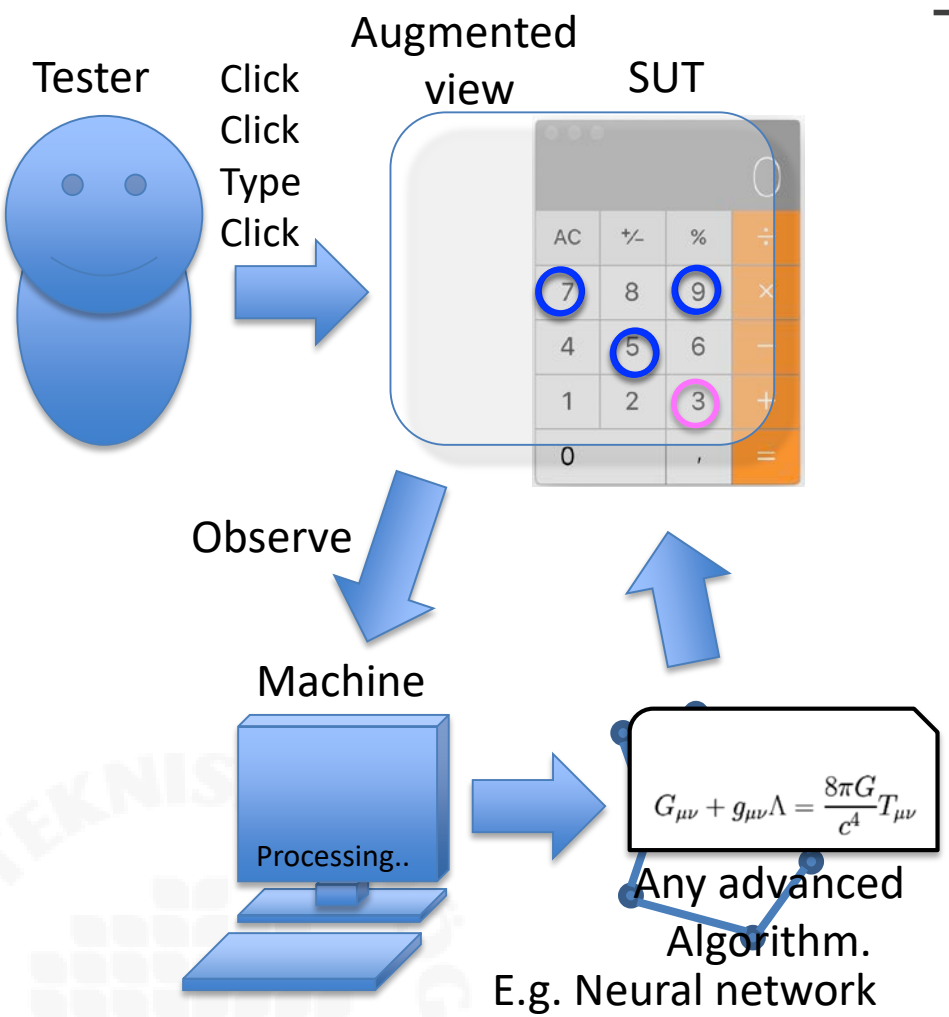
→ Modified view of system under test
- → Visualize tests
- → Visualize test improvements
  - →E.g. Action for function testing
  - →E.g. Input for quality testing
- → Guide exploration towards likely faults
- → Abstract user-patterns
  - →Learn to automatically test



Enable utilization of advanced technology



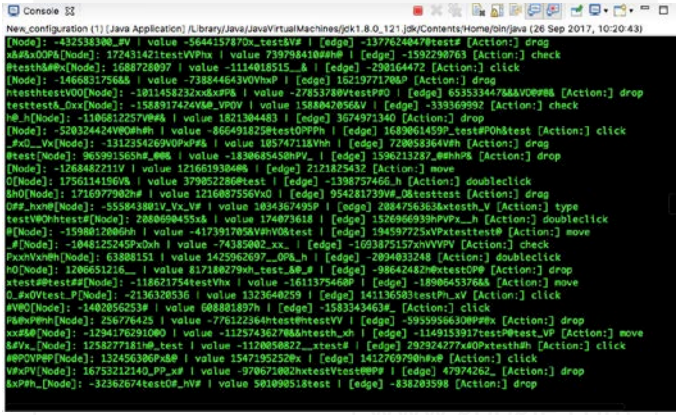
Enable utilization of advanced technology

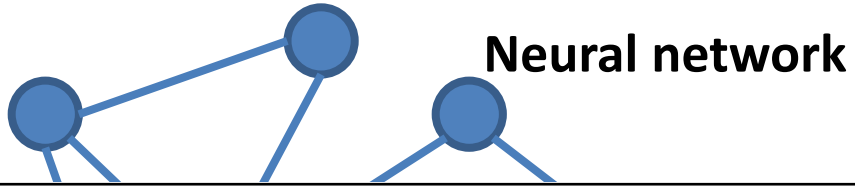


- → **Generate test cases and test data**
  - → **Meta-Heuristics/Search-based**
    - →Genetic/Evolutionary algorithm
    - →Tabu search
    - →Stochastic local search
  - → **Machine-learning**
    - →Artificial neural nets
    - →Reinforcement learning
    - →Clustering
      - →Crowd-sourcing
      - →User pattern analysis
    - →Bayesian networks
    - →Rule-based learning
  - → **Decision-trees**
    - →Game theory
  - → **And more…**

Manual tests transformed into automated tests

Input	Expected output
	www.google.com is shown
Write "Hello World"	Search results for hello world are shown





Neural network

BLEKINGE INSTITUTE OF TECHNOLOGY

NEURALT NÄTVERK

ARTIFICIAL INTELLIGENCE (AI)

..#.H3llo... Any0ne ther3?.. (Y/N)

AI-BASED TESTNING

New Technology, New Challenges

HOW DO YOU **CONTROL** AN AI?

CAN WE **TRUST** AN AI?

WHO IS RESPONSIBLE?

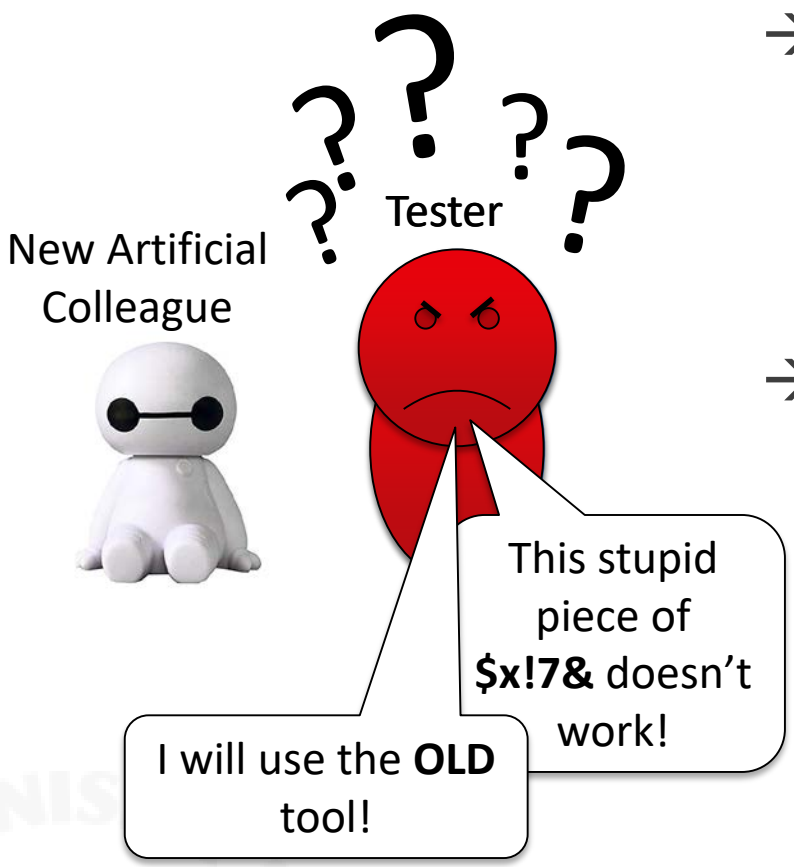
WHAT HAPPENS WHEN THE AI DOES SOMETHING WRONG?

WHAT SHOULD THE AI LEARN?

DO PEOPLE **WANT TO** WORK WITH AI?

…AND MUCH MORE!

Very technical! Remember the tester!



- → **User-centric guidelines**
  - → Why test?
  - → How to test?
  - → When to test?
  - → Who should test?
- → **What if the machine makes a misstake?**
  - → Human factors
    - → Trust
    - → Enjoyment
    - → Stress
    - → Teamwork
    - → Etc.

Human-Machine Mutualism (HMM)

Costly

Tedious

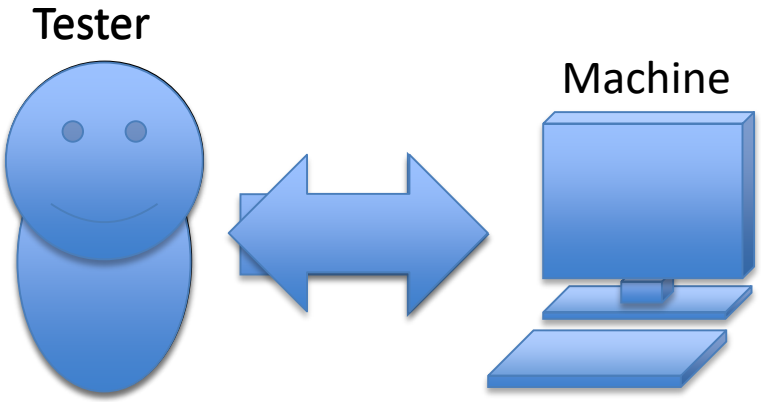
Error-prone

Prerequisites

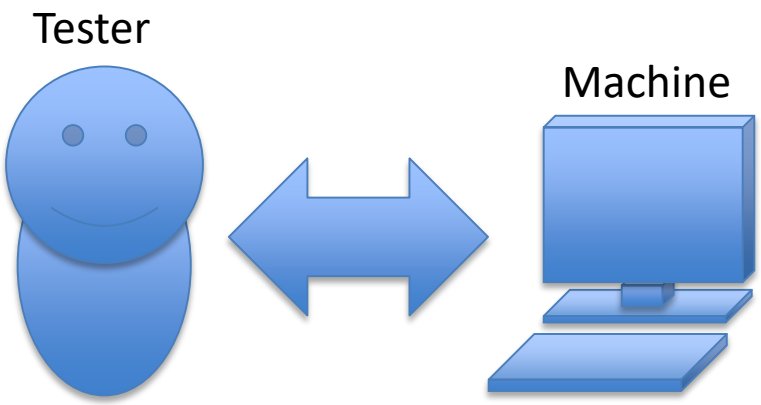
Regression test

Technical difficulties

- → **The goal:**
  - → A new approach to automated testing based on human-machine mutualistic collaboration
- → **The purpose:**
  - → More effective and efficient test automation
  - → More enjoyable testing
- → **Our need:**
  - → Collaborators that share this goal



Questions?

Contact:
Emil Alégroth (Sub-project leader): Emil.Alegroth@bth.se

Michel Nass: Michel.Nass@bth.se
Robert Feldt: Robert.Feldt@gmail.com
Michael Unterkalmsteiner: Michael.Unterkalmsteiner@bth.se