

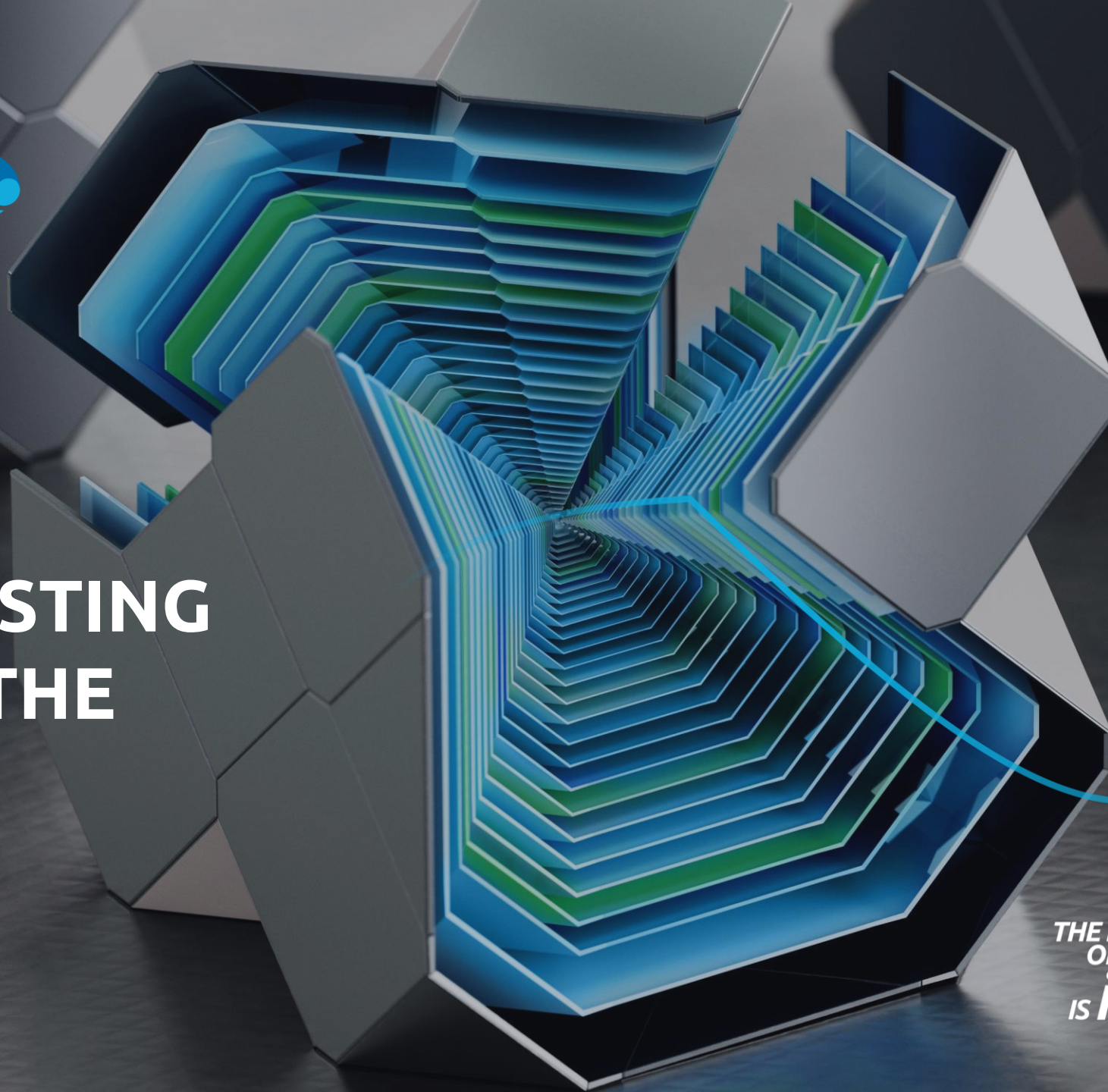


**HISC**  
HIGH INTEGRITY SOFTWARE  
**CONFERENCE**  
OCT 17, 2023

# EXPERT TESTING WITHOUT THE EXPERTS

Thomas Wilson  
HISC 2023

THE FUTURE  
OF YOUR *industry*  
IS *intelligent*





01

**ABOUT ME**

02

**EARLY MOTIVATIONS**

03

**OUR APPROACH**

04

**RELATED WORK**

05

**CASE STUDIES**

06

**CONCLUSIONS**





# THOMAS WILSON

High-Integrity  
Software Engineer



With **Capgemini**  **engineering**  
formerly, **ALTRAN**  
for **16** years

- 10 years client project engineering
- 6 years software engineering process improvement research



PhD in Applied Formal Methods



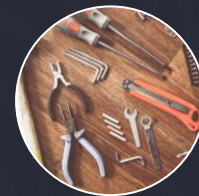
Senior Architect for Test for High-Integrity Expertise Centre of Capgemini Engineering in Bath



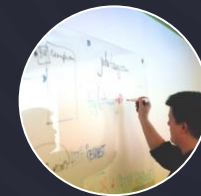
## IMPROVEMENT ENGINE



Research & Technology



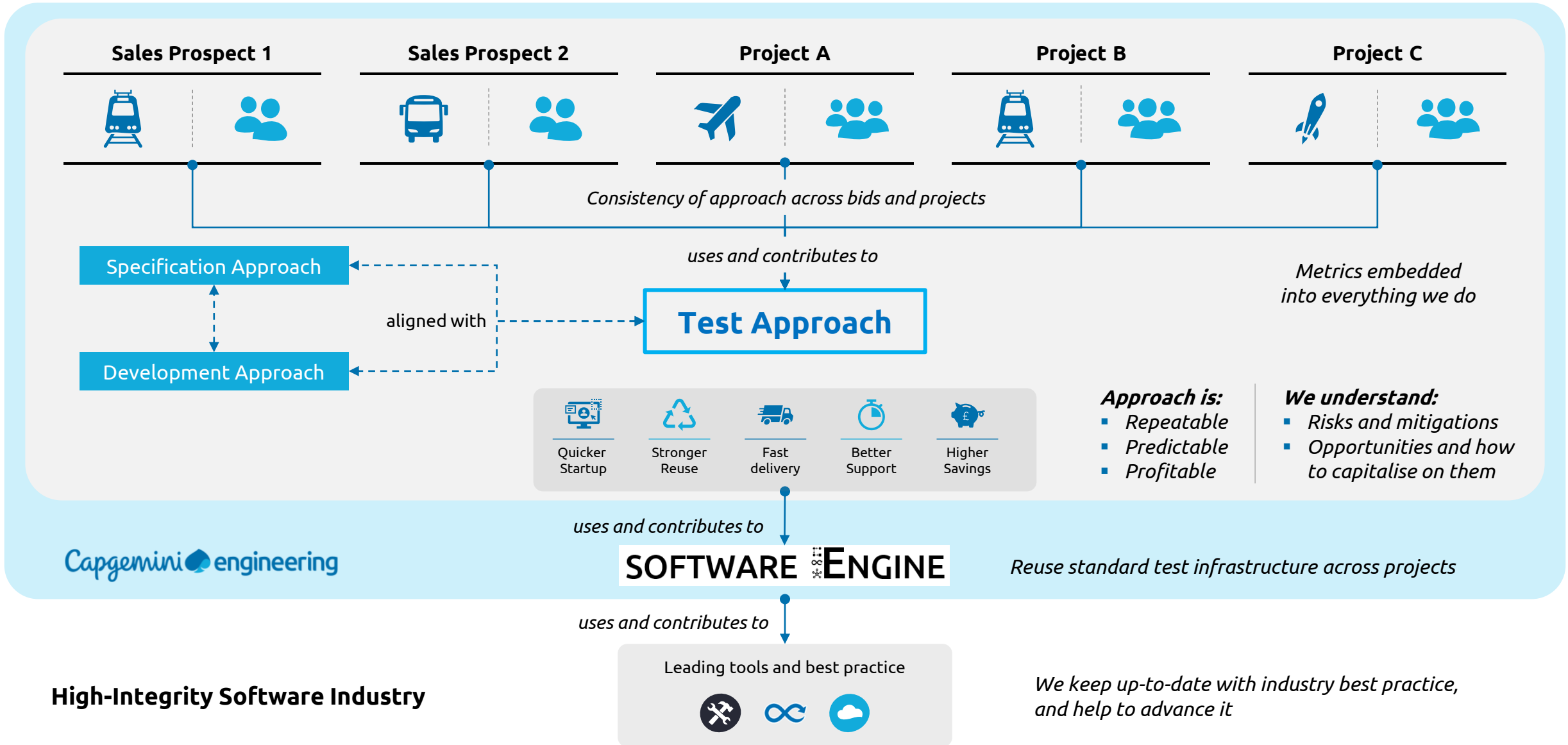
Production Environment



Special Interest Groups



# IMPROVING MODEL-BASED TESTING



**High-Integrity Software Industry**

# EARLY MOTIVATIONS



SOFTWARE  
FOR SMALL  
BUSINESSES

“BUSINESS  
MADE  
SIMPLE”



16 YEAR OLD  
PROGRAMMER

CAREER-  
SHAPING  
EVENTS

BUSINESS

contact  
management  
database

accountancy  
software

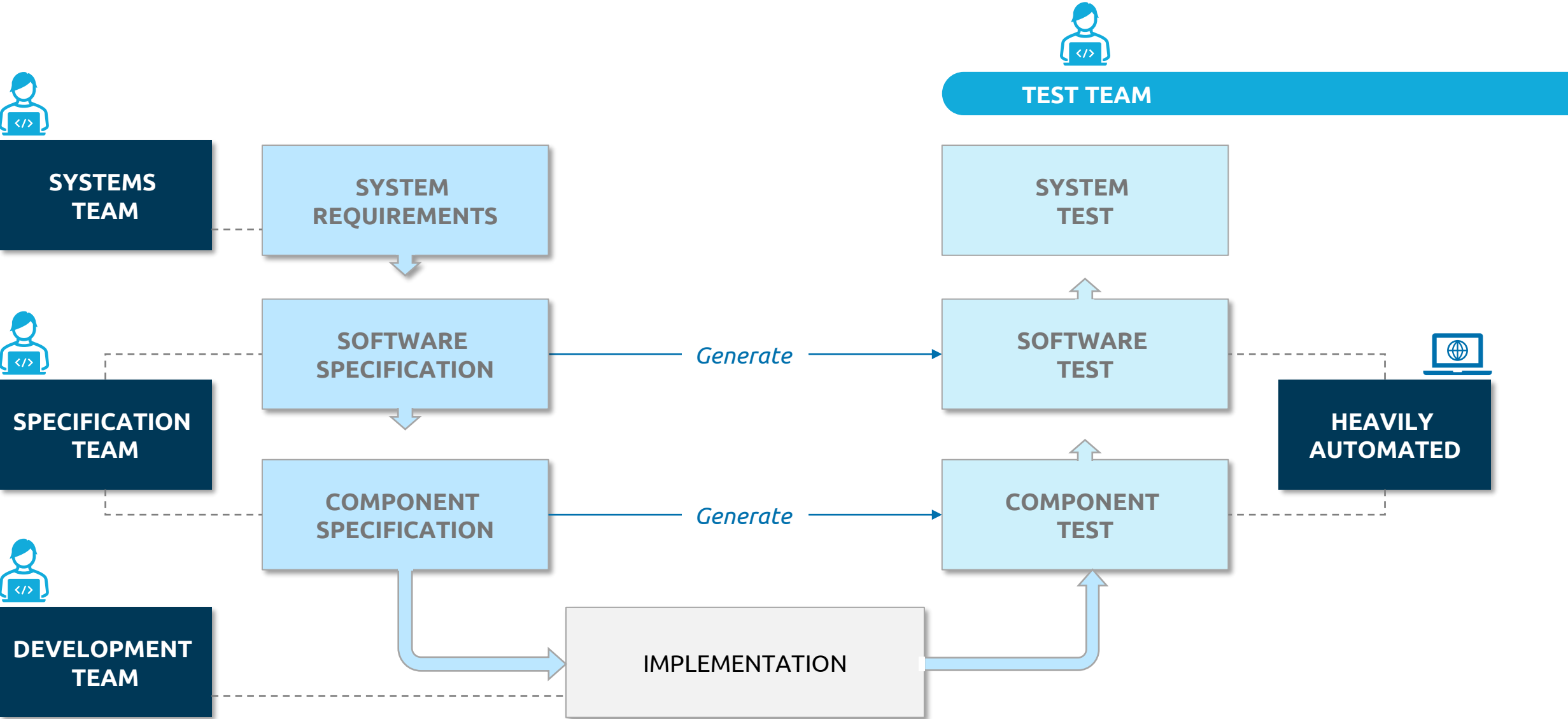
MATHS

Can we use maths  
more?

Programming is  
mathematical

Can we exploit  
that more?

# OUR APPROACH



# MODEL-BASED SPECIFICATION



## DEFINE CONTEXT AND EXTERNAL INTERFACE(S)



- Model of the world needed to describe the machine's interface
- Define interface(s) to operations of the machine, with their inputs and outputs

## SPECIFY DETAILED BEHAVIOUR



- Specify persistent state required for system/packages
- Specify detailed behaviour for (partial) operations

## DECOMPOSE AS REQUIRED

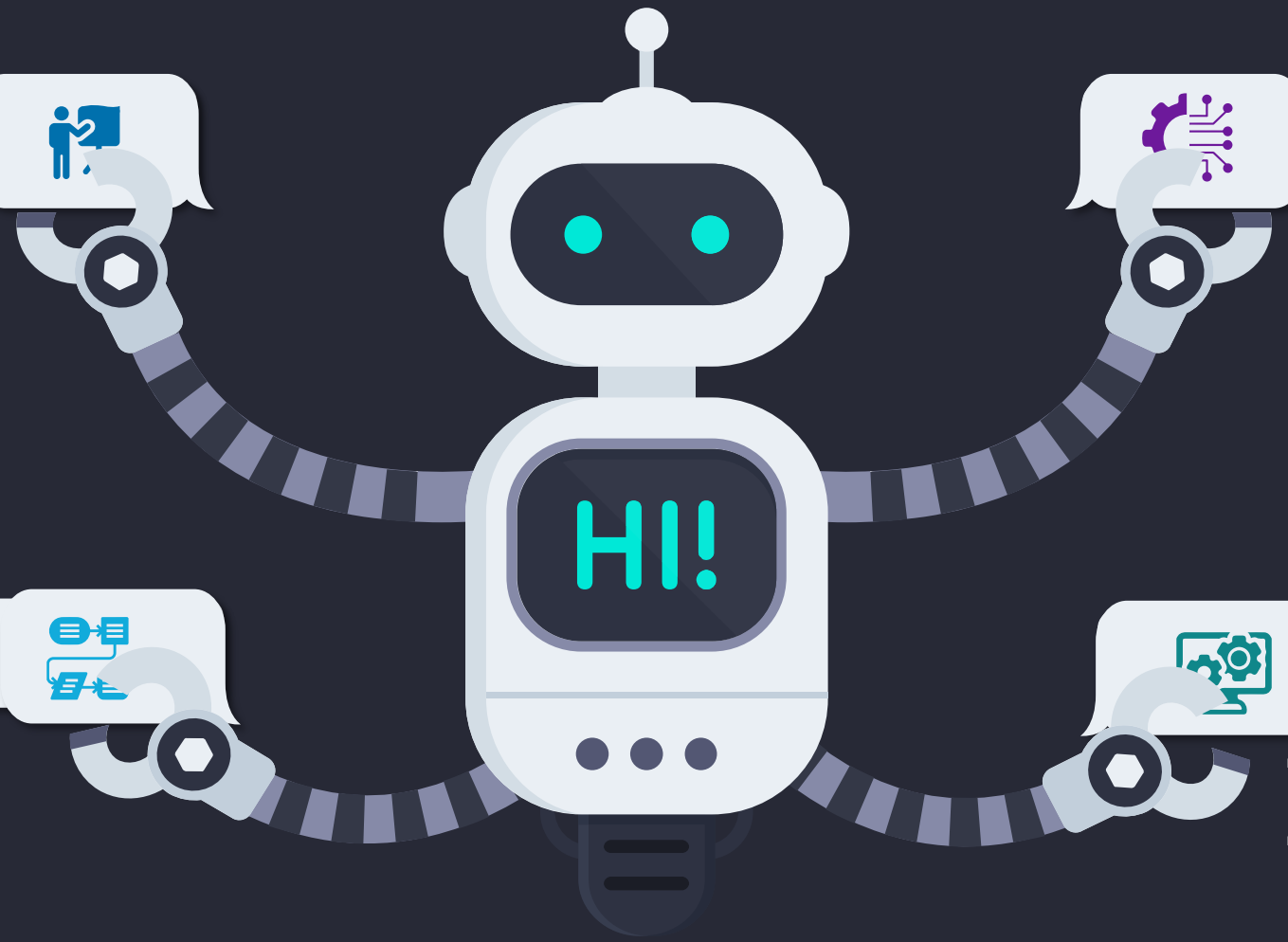


- Where required, identify suitable packages to divide specification into and decompose operations into partial operations that update the separate packages

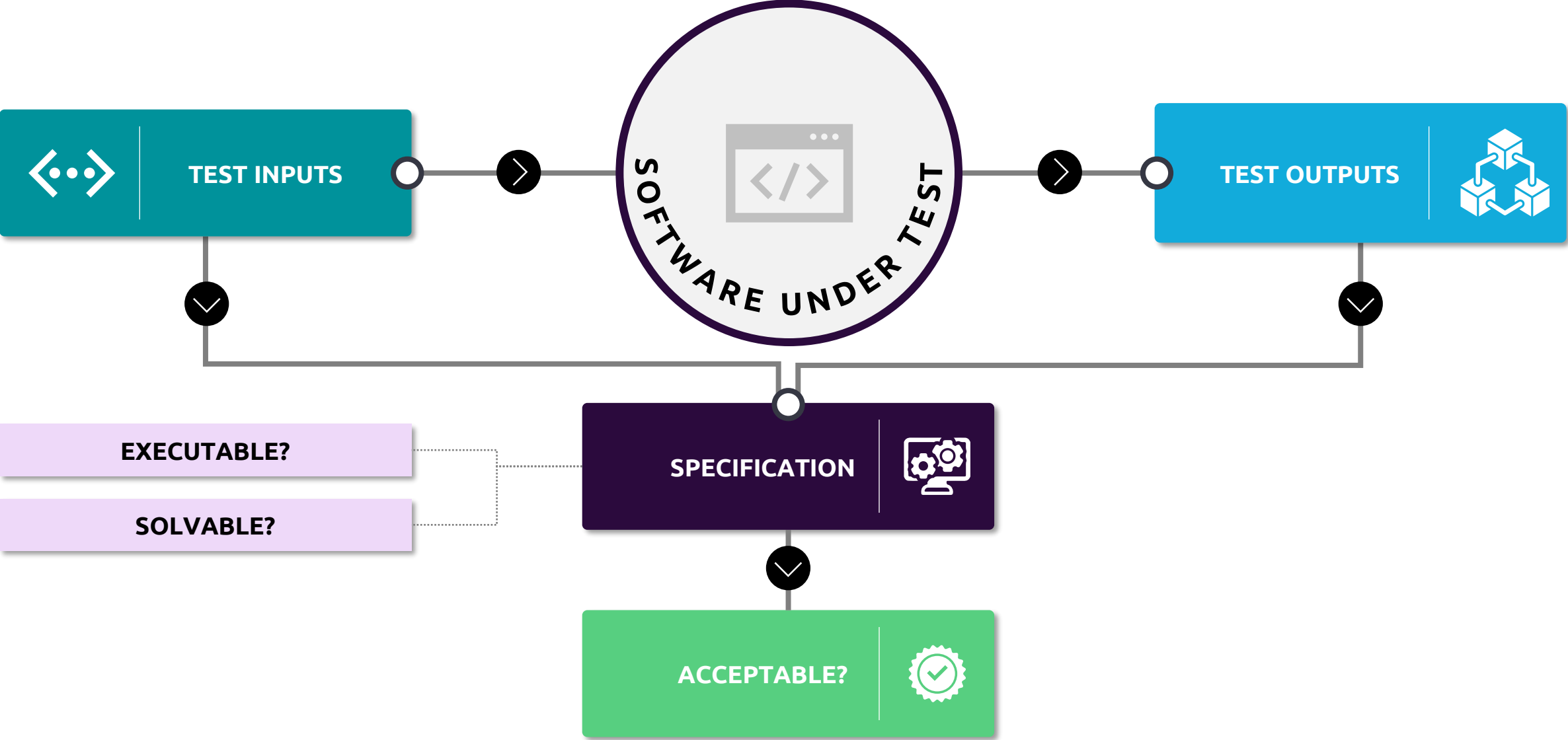
## CHECK CORRECTNESS AND USEFULNESS



- Check unambiguous, consistent, complete
- Check feasible, refutable, traceable, usable



# DETERMINING WHAT THE TESTS SHOULD CHECK FOR





# COVERAGE CRITERIA FOR RULE-BASED TESTING



When processing a file containing:

```
pred p1 [ a, b : Bool] {  
  a or b  
}
```

If we had an expression:

```
p1[a1, b1] or c1
```

Then we'd generate conditions:

a or b

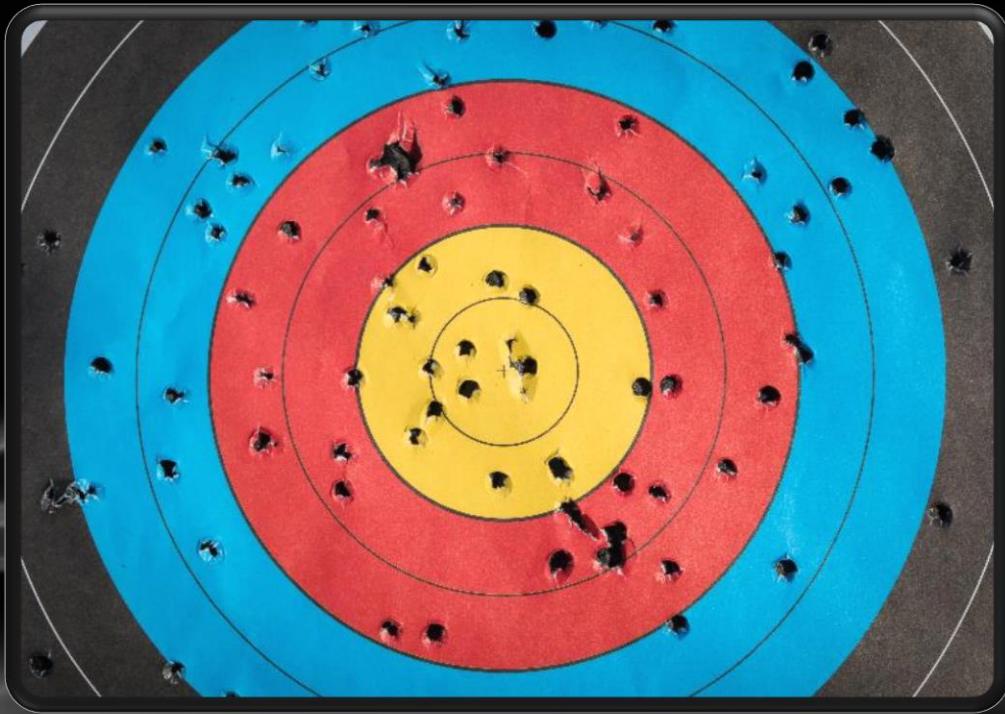
	1 (T)	2 (T)	3 (F)
a	●	○	○
b	○	●	○

- 1 (T)
  - p1[a1, b1]
    - **let** a = a1, b = b1 |
      - a
      - **not** b
    - **not** c1
- 2 (T)
  - p1[a1, b1]
    - **let** a = a1, b = b1 |
      - **not** a
      - b
    - **not** c1
- 3 (T)
  - **not** p1[a1, b1]
    - **let** a = a1, b = b1 |
      - **not** a
      - **not** b
    - c1
- ...

# GETTING COVERAGE FOR RULE-BASED TESTING



## BLACK BOX



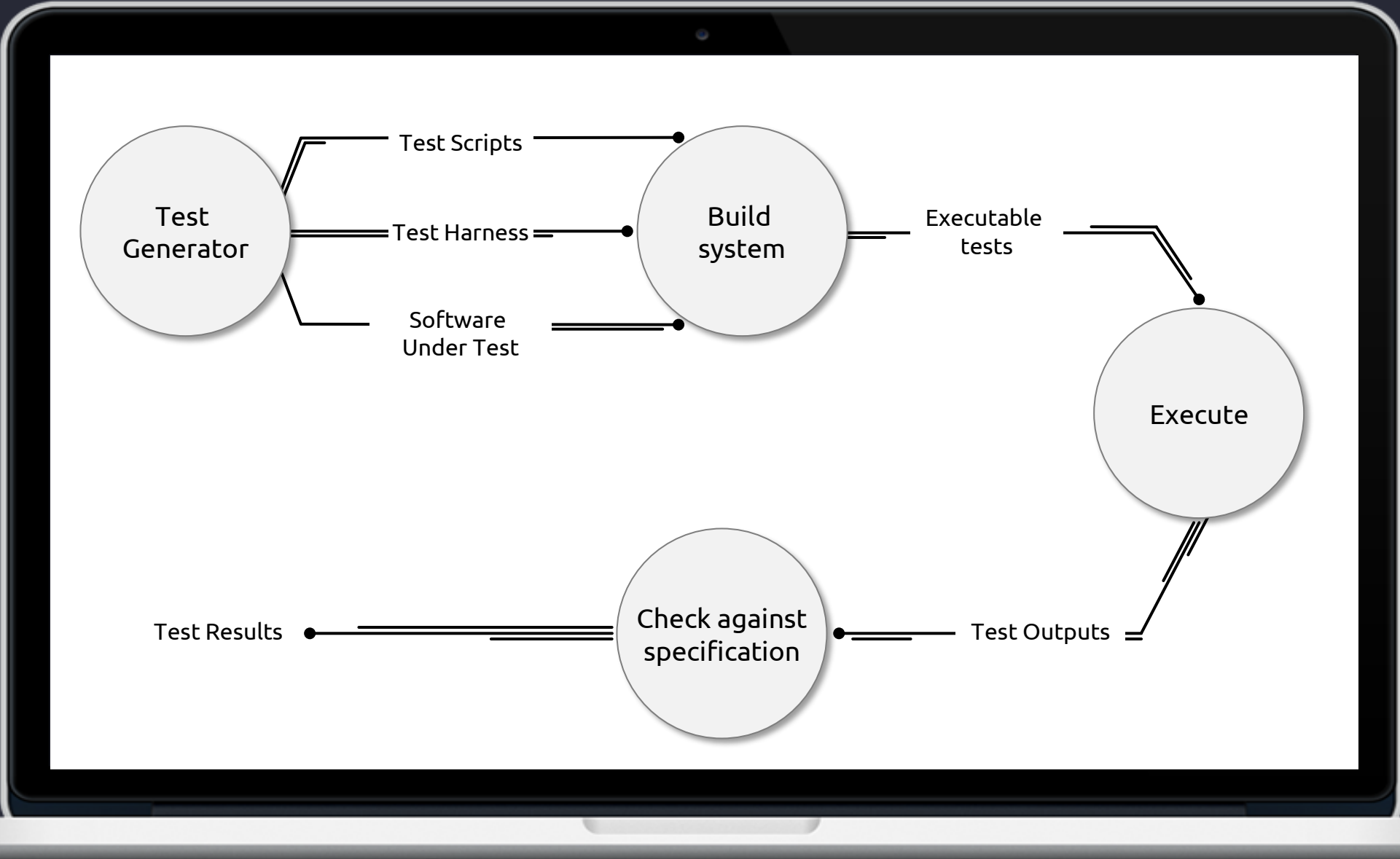
**CONSTRAINED RANDOM  
ML-BASED  
SOLVER-BASED ON CODE**

## GREY BOX

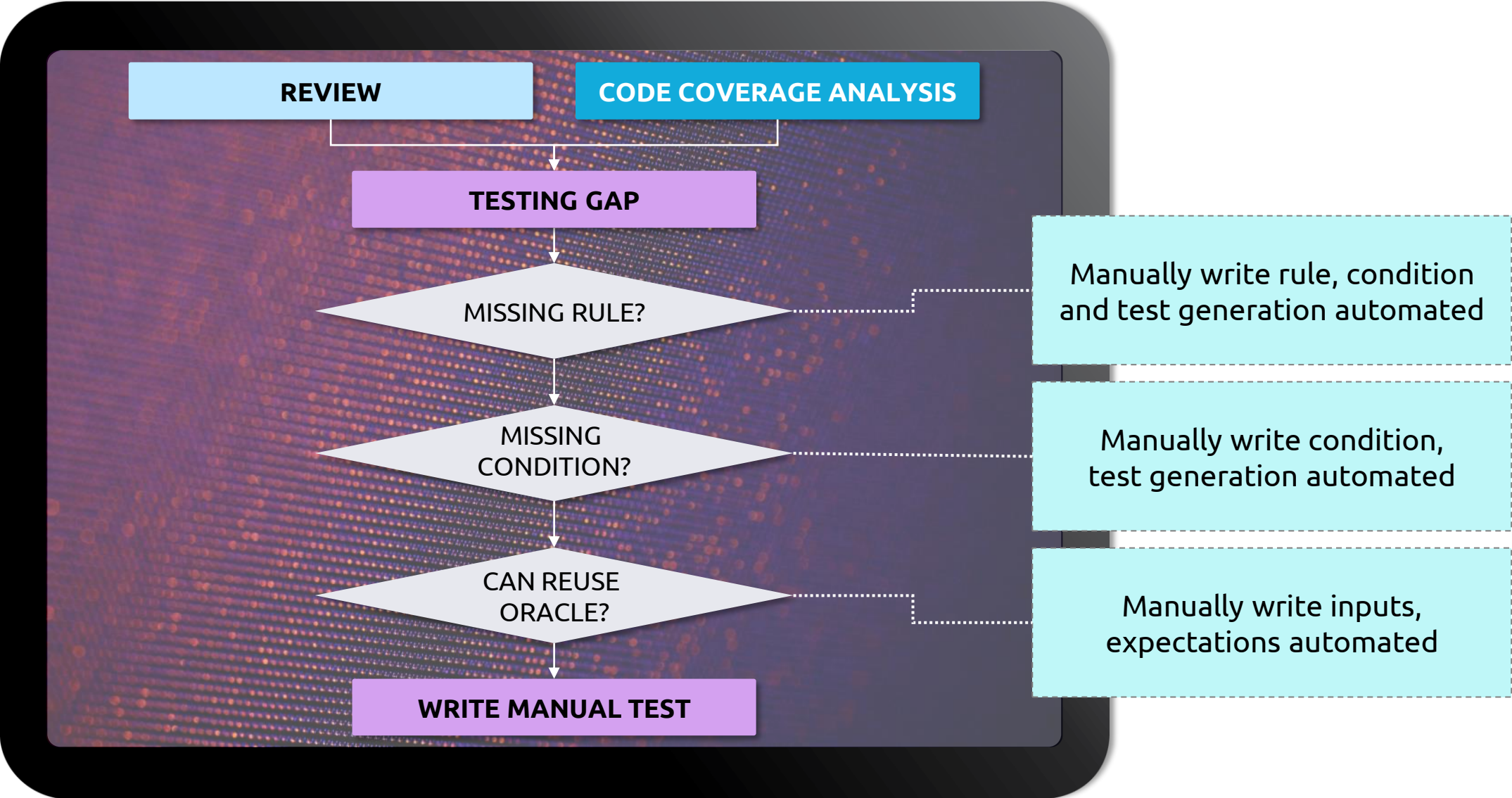


**SOLVER-BASED ON SPECIFICATION**

# RUNNING TESTS AGAINST AN IMPLEMENTATION

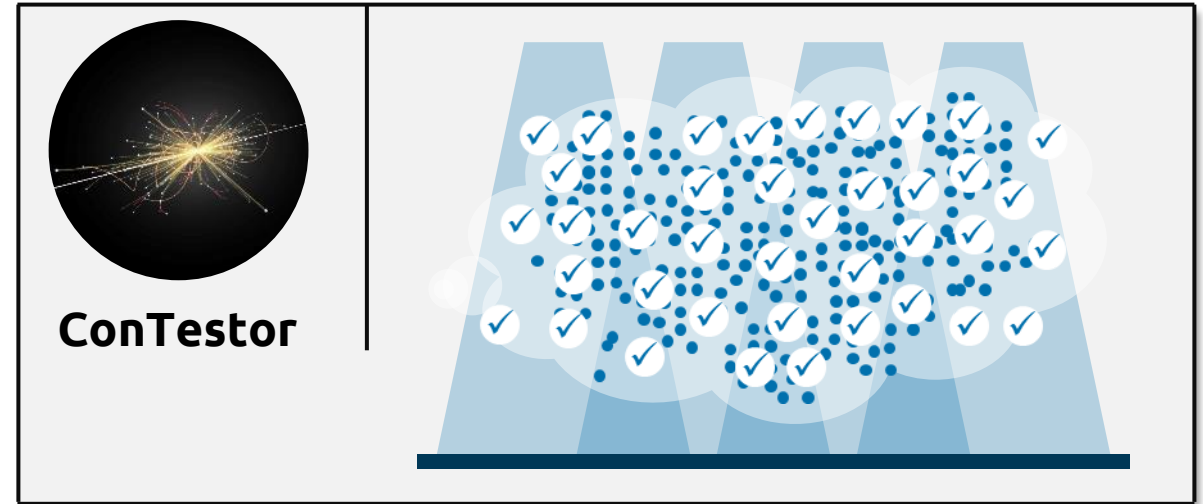
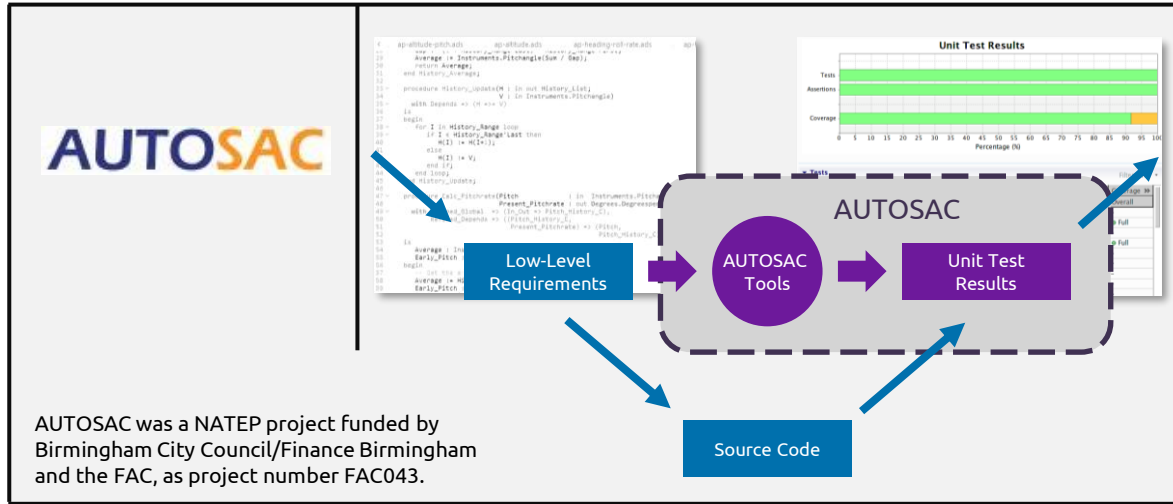


# CREATIVE TESTING





# RELATED WORK



**CRUCIBLE**

The development of Crucible was supported by the HICLASS project, funded by the Aerospace Technology Institute and Innovate UK, as project number 113213.

**Crucible - Test Generation Dashboard**

Total Time: 03:58:03  
Condition Generation: 00:00:26  
Solving: 03:57:37

**Verification Conditions**

Number of VCs vs Solving Time / hours

**Path Conditions**

Number of PCs vs Solving Time / hours

**test\_1**

Tested in 39 seconds

**Tracing**

- call pred.pre3
- comparison.eq.pred1
- comparison.in.pred1
- comparison.c.pred3
- logical.and.pred1
- logical.block.pattern.3
- logical.impraise.pred.1

**Path Condition**

```
test1(cycles, testCycle)
+ ((testCycle.(vc0.prev)) & cycles)
+ operation!(testCycle.pre), (testCycle.post), (testCycle.inputs), (testCycle.outputs)
```

**Solution**

- Solution
- cycles = (8 entries)
- testCycle = cycles5

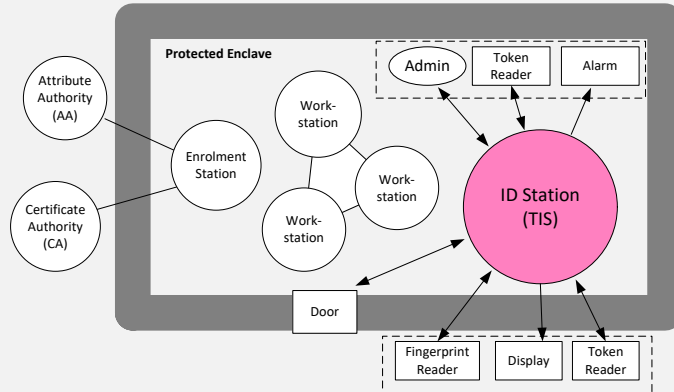
**Code Snippet:**

```
with Types: use Types;
with Alloy_Interface: use Alloy_Interface;

procedure Test_1 is
  Cycles : constant Set_Cycle :=
    Set_Cycle {
      Length => 6,
      Values => Set_Cycle_Values {
        1 => Cycle'
      }
    }
  Pre => Null_Optional_Steam_Boiler,
  Inputs => Null_Optional_Inputs,
  Outputs => Null_Optional_Outputs,
  Post => Steam_Boiler {
    C => 50,
    M_1 => 10,
    M_2 => 40,
    N_1 => 20,
    N_2 => 30,
    Normal_Range => Set_Integer {
      Length => 11,
      Values => Set_Integer_Values {
        1 => 20,
        2 => 21,
        3 => 22,
        4 => 23,
        5 => 24
      }
    }
  }
end Test_1;
```



## TOKENEER



- Alloy specification from original Z specification
- Generate unit tests against original SPARK
- Animation to validate

## STEAM BOILER

Input	Value
stop	F
sbw	T
pur	T
ps	[T,T,T,T]
pcs	[T,T,T,T]
level	240L
steam	0L
pr	[F,F,F,F]
pcr	[F,F,F,F]
lr	F
sr	F
pla	[F,F,F,F]
pcfa	[F,F,F,F]
lfa	F
sfa	F

Cycle4 State: Normal Mode

Output	Value
mode	Normal
pr	F
vo	F
po	[T,T,T,F]
pfd	[F,F,F,F]
pcfd	[F,F,F,F]
lfd	F
sfd	F
pra	[F,F,F,F]
pcra	[F,F,F,F]
lra	F
sra	F

- Alloy specification from original requirements
- Generate test sequences against SPARK implementation
- Manual testing via animator

## PROJECT X

Retrospectively applied to past client project

- Alloy specification from original requirements
- Generate unit tests against original SPARK
- Use of CVC4 as external solver

## PROJECT Y

Being applied to live client project

- Alloy specification from software requirements
- Generate component tests against SPARK implementation
- Animation to validate

# CONCLUSIONS

## CAN WE PERFORM EXPERT TESTING WITHOUT THE EXPERTS?

### 3 . F U T U R E



Automate even more

### 2 . P R E S E N T



Deployment on more live client projects



Deployment on first live client project

### 1 . P A S T



Proof of concept developed





**GET THE  
FUTURE  
YOU WANT**

[capgemini.com](https://www.capgemini.com)





## About Capgemini

Capgemini is a global leader in partnering with companies to transform and manage their business by harnessing the power of technology. The Group is guided everyday by its purpose of unleashing human energy through technology for an inclusive and sustainable future. It is a responsible and diverse organization of over 360,000 team members in more than 50 countries. With its strong 55-year heritage and deep industry expertise, Capgemini is trusted by its clients to address the entire breadth of their business needs, from strategy and design to operations, fueled by the fast evolving and innovative world of cloud, data, AI, connectivity, software, digital engineering and platforms. The Group reported in 2022 global revenues of €22 billion.

Get The Future You Want | [www.capgemini.com](http://www.capgemini.com)



This presentation contains information that may be privileged or confidential and is the property of the Capgemini Group.

Copyright © 2023 Capgemini. All rights reserved.