RAPITA Systems Ltd
A DANLAW Company

Something you didn't know about

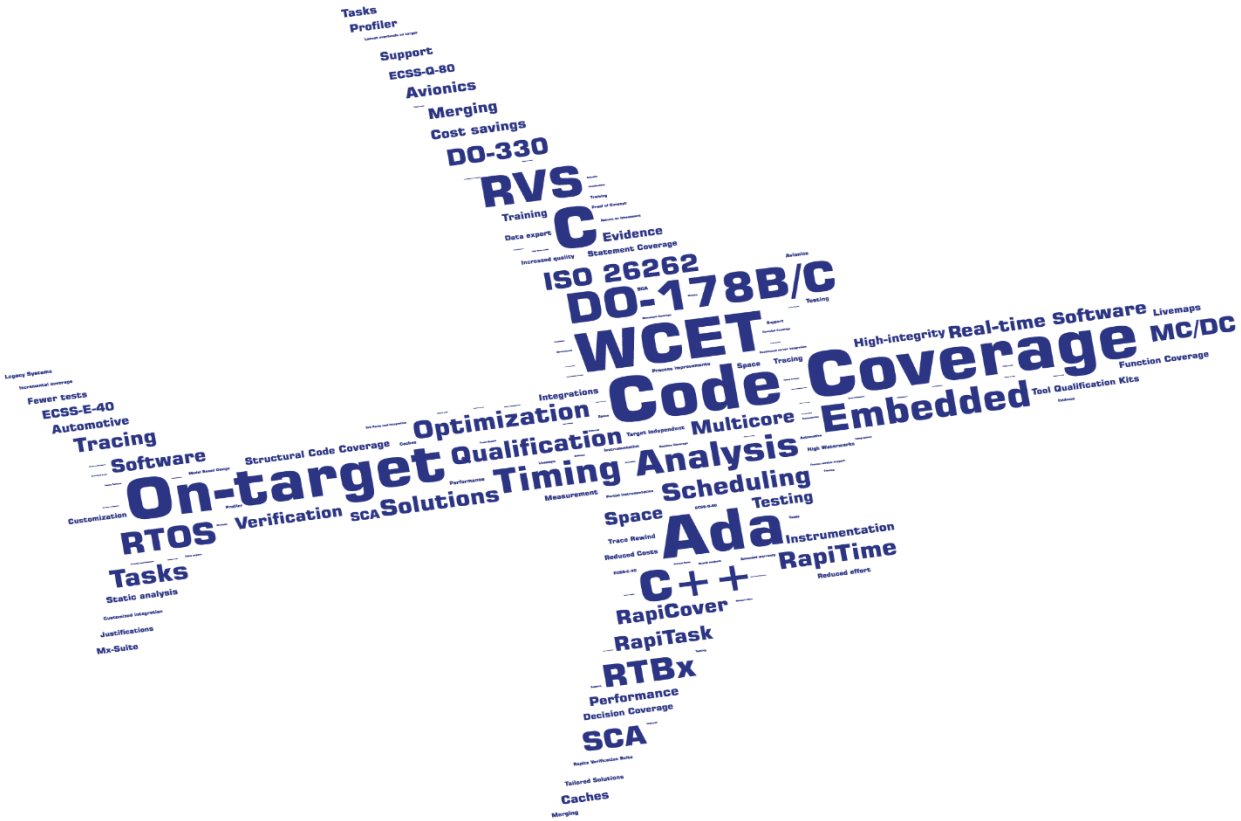
...

Rapita/Ada/Software

Ian Broster

ianb@rapitasystems.com

What does Rapita do?



Advanced technologies for reducing V&V costs

Revolutionize your software verification

RVS: Software verification tools and hardware

- Manage **tests** from system to unit level
- Apply and **run** tests on-target and on-host
- Significantly reduce testing times

RapiTest Framework

RapiCover

- Measure **code coverage** to **MC/DC** level
- Lowest **on-target** overheads on the market
- Merge coverage from multiple tests and builds

- Calculate **WCET** and high water mark times
- Identify where to focus **optimization**
- Single & multi-core analysis

RapiTime

RapiTask

- Visualize system **scheduling** graphically
- Highlight rare timing events e.g. race conditions
- Identify system capacity issues

- **Test simulation models** and software code
- Provide evidence that code meets requirements
- Test on target ECU

MX SUITE™

RTBx

- Trace 100+ million events per second for days
- Minimal **instrumentation overheads**
- Target-independent tracing

Talk to us about…

- Commercial projects (tools and services)
- Custom V&V tools
- Academic use
- Partnerships
- R&D projects
- Ada!

Supporting Ada & Aerospace – recent things

RapiTest

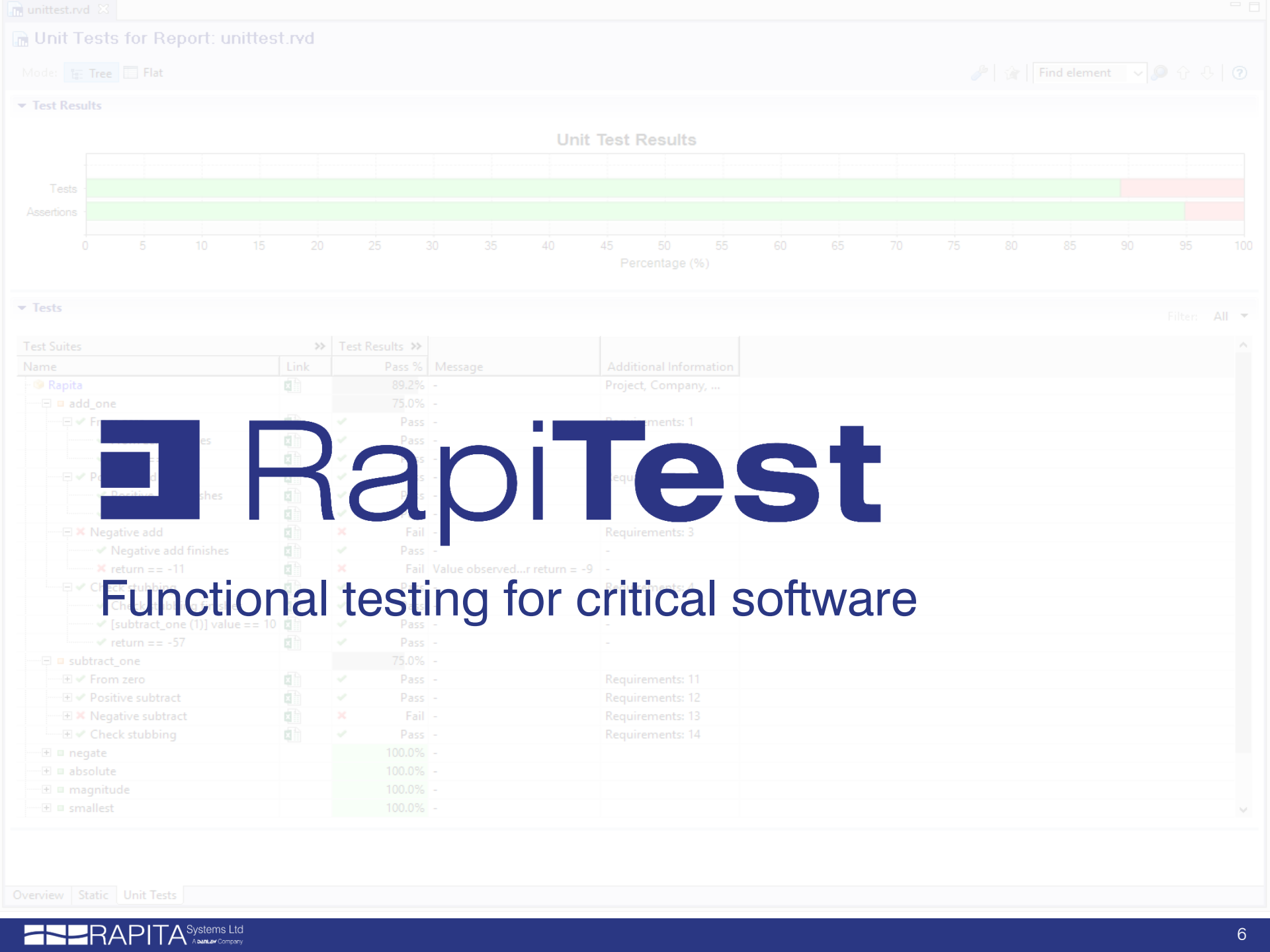
- Major product investment (Ada/C/C++)

Multi-core timing analysis/WCET

- Industrial solution for safety-critical MCP

Automatic test generation

- For RBT, coverage and formal modelling

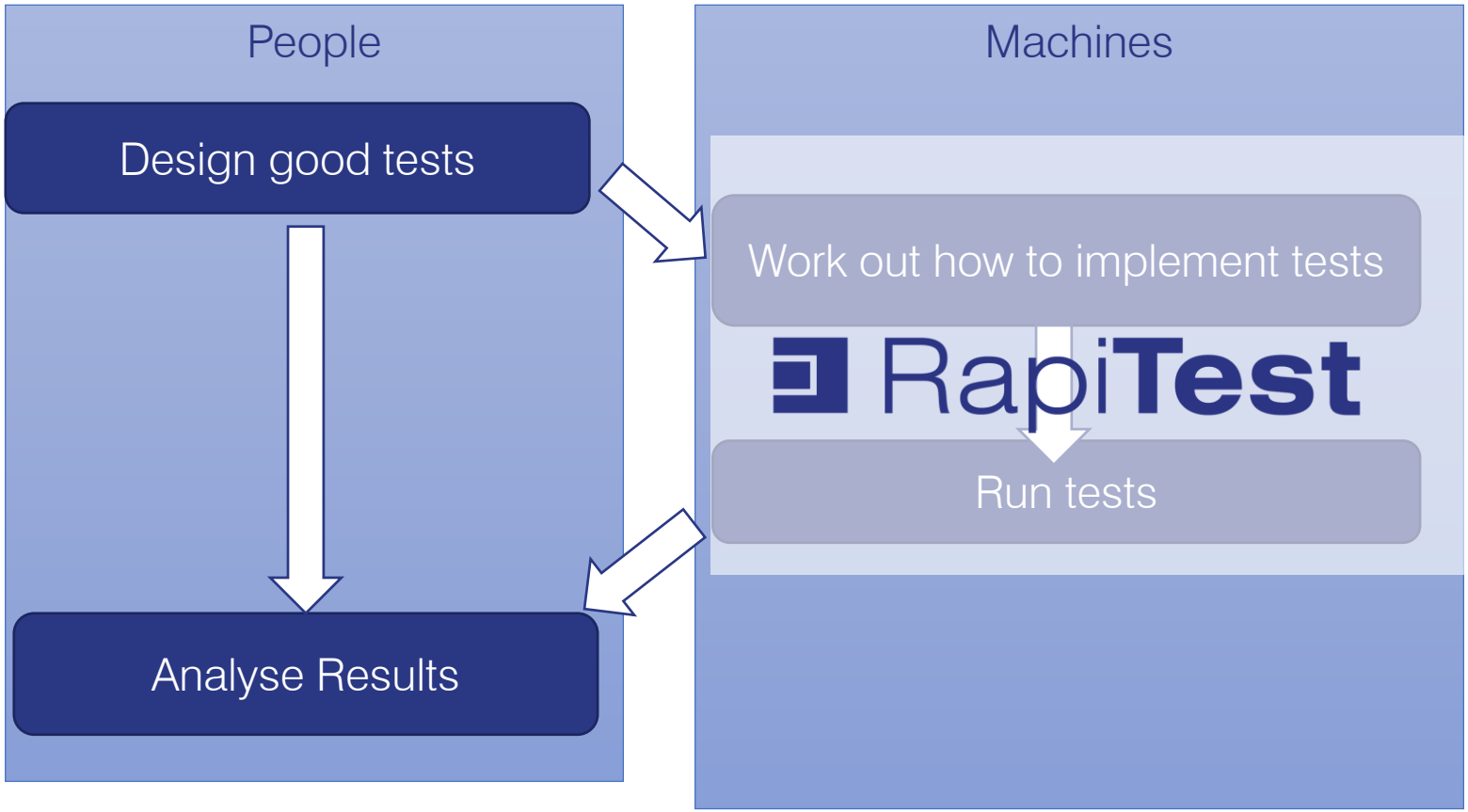


RapiTest

Functional testing for critical software

RapiTest Philosophy: Efficient People

Unit and system testing tool (Ada, C, C++)
Designed for aerospace and Ada.



In many DO-178C projects, testers cannot even "see" the source

No "User code" – tester should not have to write any code to test Ada

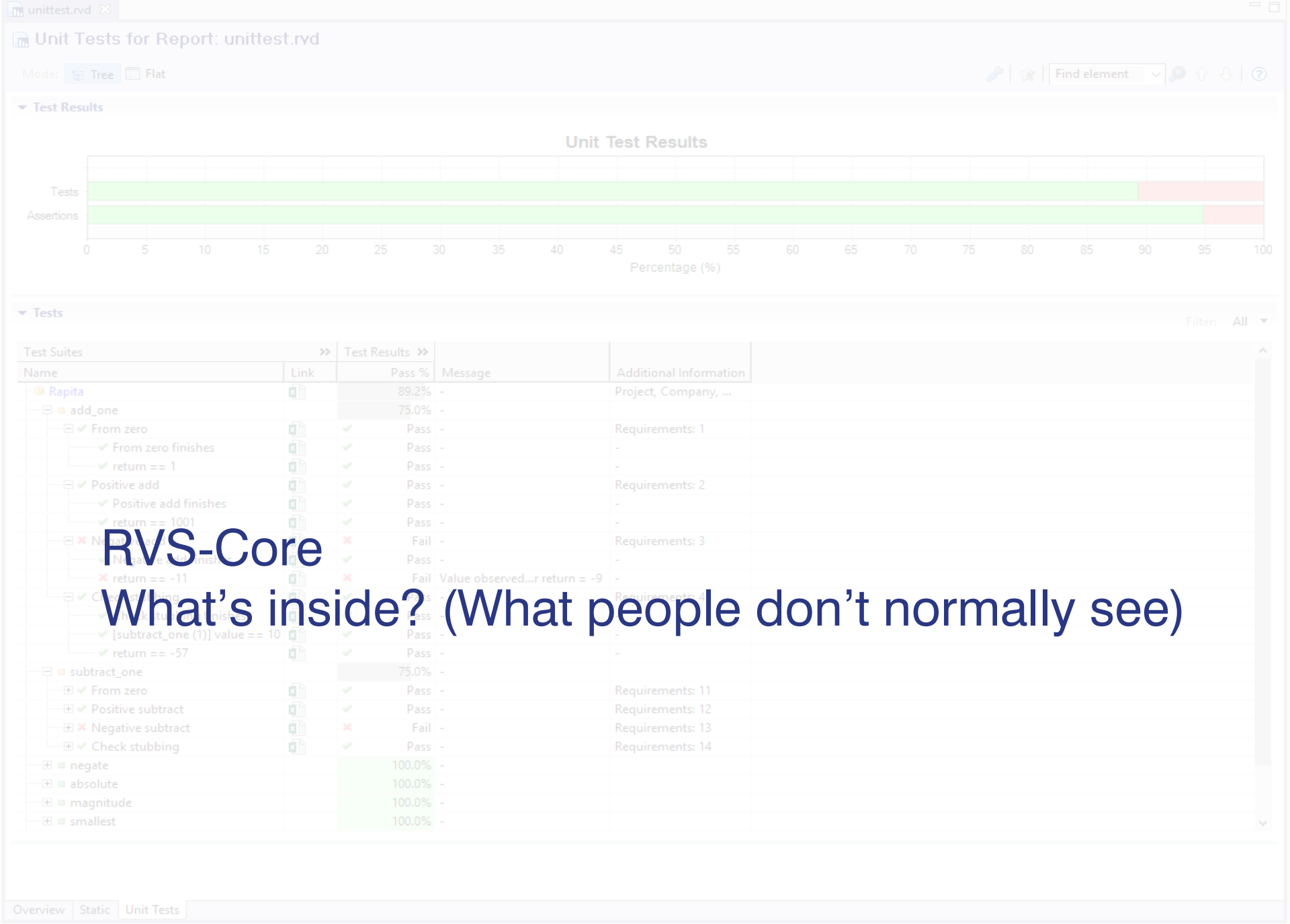
After 3 years of development and test

RapiTest
RapiCover

- RapiTest version 1.4a
  - C, Ada, C++
  - DO-178C Qualification
  - Lots of really good feedback from first users

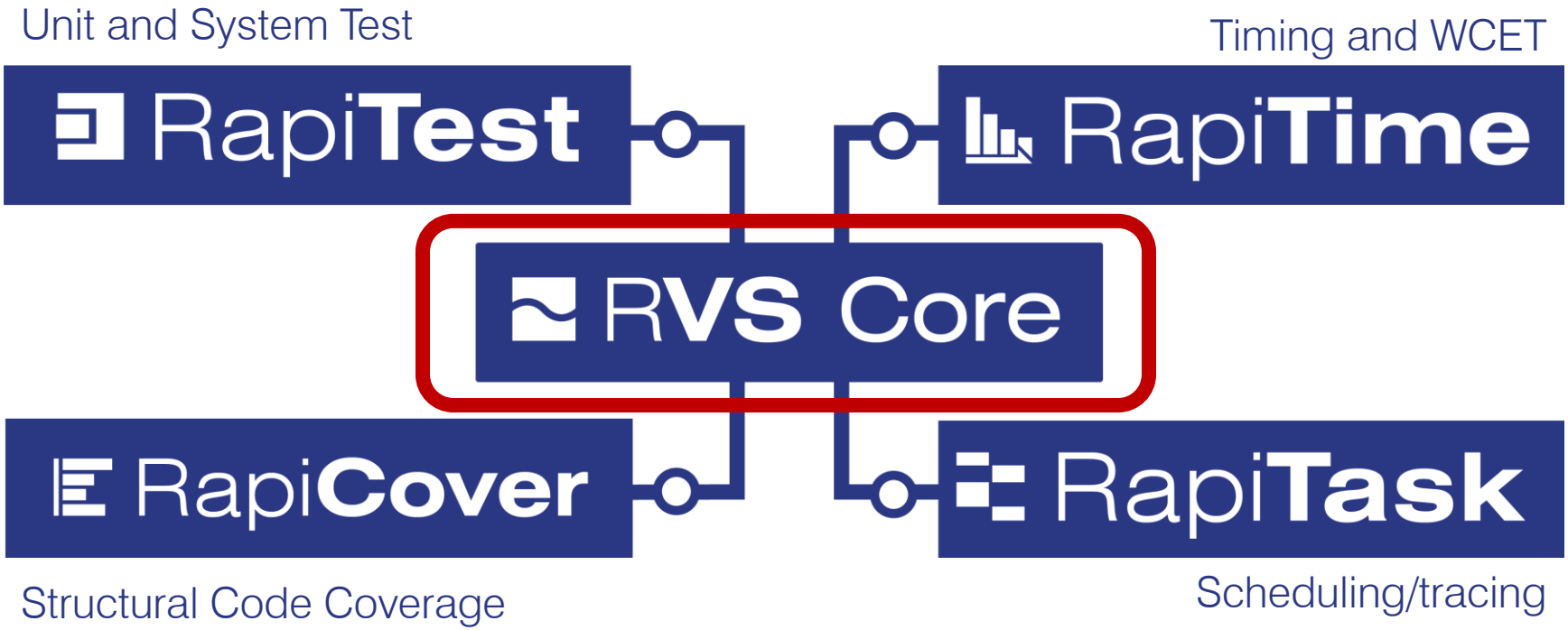
- Would you like to try it?
  - We welcome your ideas and feedback. What more it could do?

- Technology maturity:
  - Successfully completed 3 V&V projects
    - DO-178C DAL A – software accepted by OEM/DER
  - In use on several DO-178C DAL A projects in US & UK
  - Further pilots/eval ….



RVS-Core
What's inside? (What people don't normally see)

Rapita Verification Suite

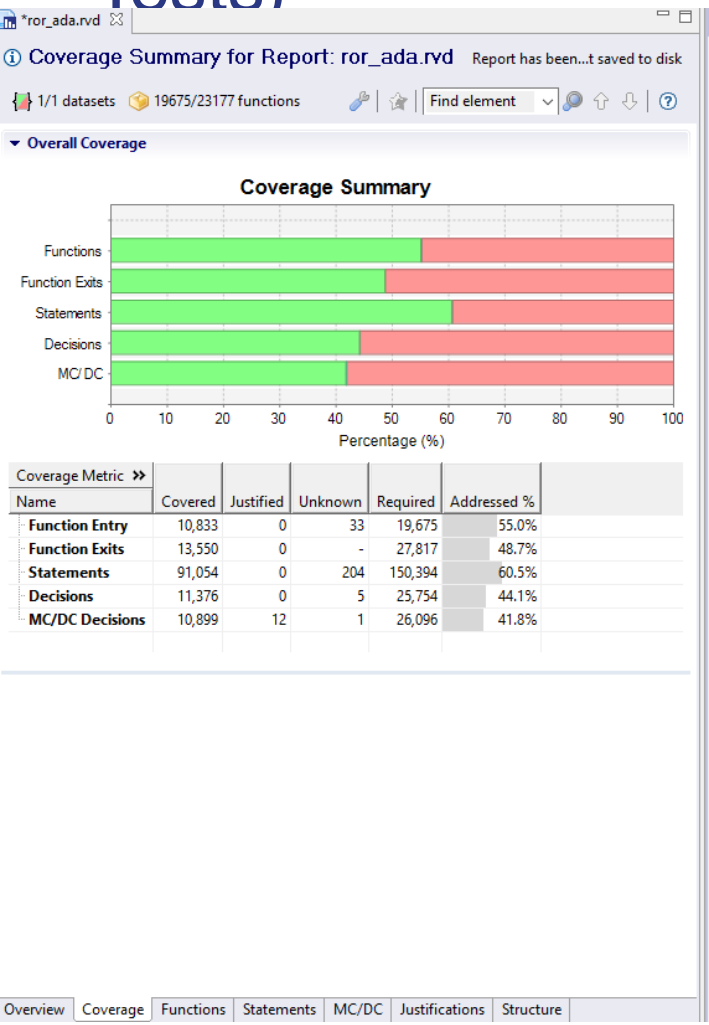


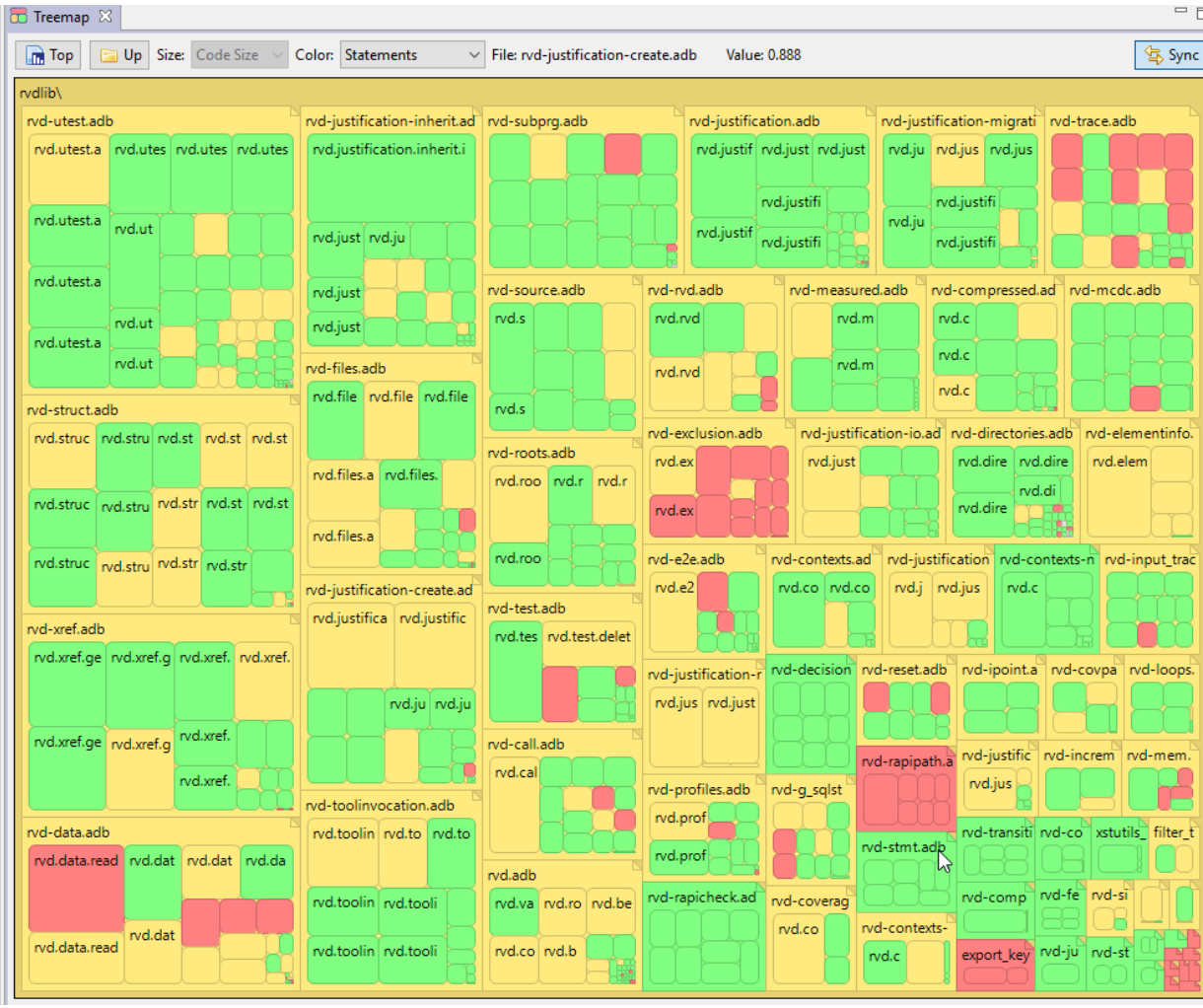
RVS-Core is..

- A powerful set of language parsing libraries
  - Ada 83-2012, C, C++
- Many Target/embedded integration libraries
- Reporting/GUI
- Witten in Ada (mostly)
  - Except the bits in Java, Python, JavaScript, Perl, C and 3 DSLs
- Tested with RVS

AdaCore

RVS on RVS (Statement Coverage from System Tests)



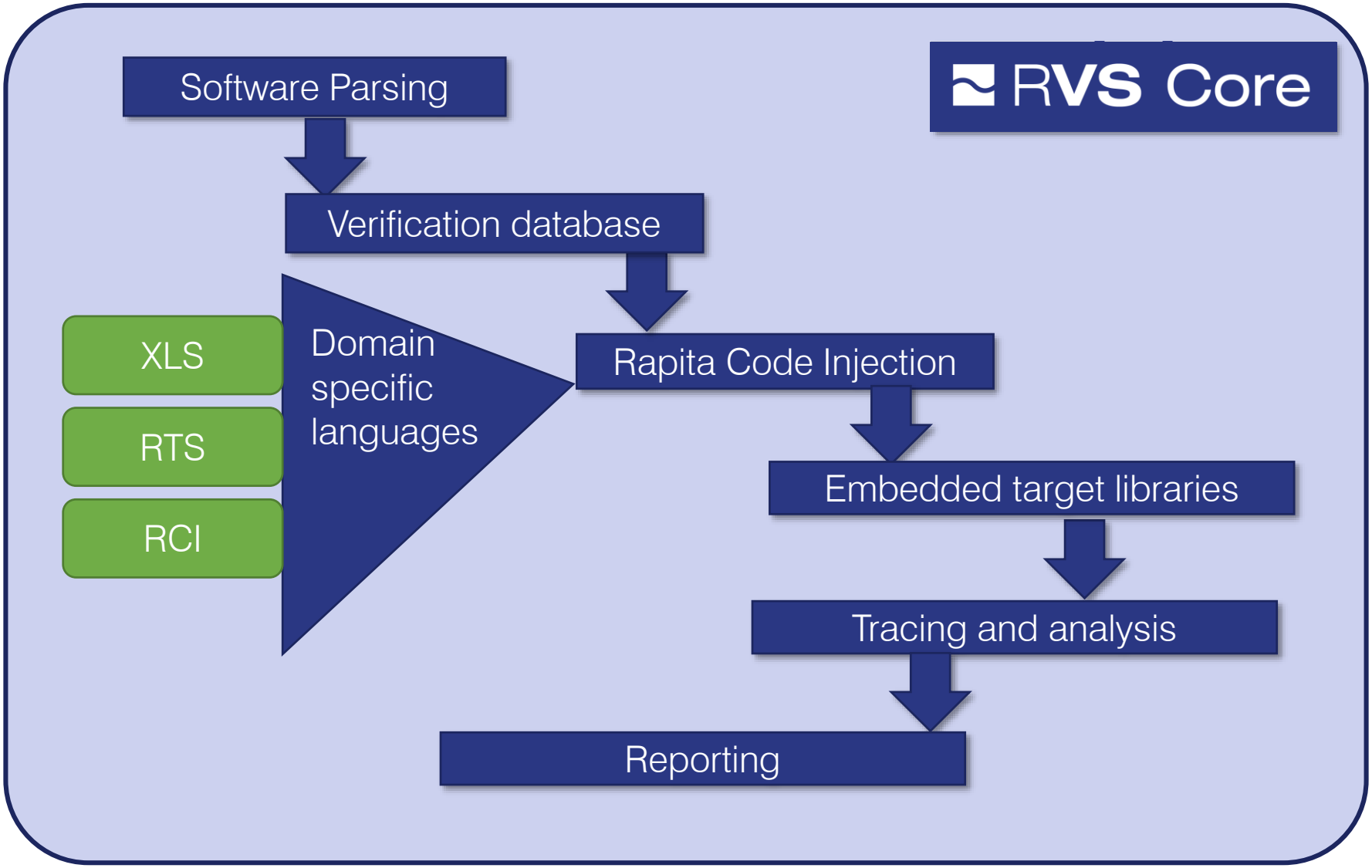


RVS Core can…

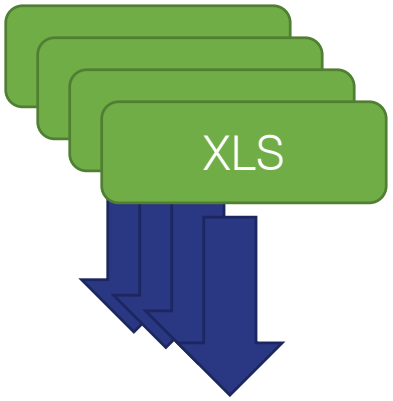
- Do “anything“ to source code
  - Code injection/modification
  - Analysis
  - Tracing
  - Data/Control coupling verification
  - Auto test-generation

- Q: How else can we use the core?

Technologies and dataflow inside RVS-Core



Domain Specific languages in RVS



[Set of] spreadsheet-based languages for writing tests (including legacy combability converters)

Rapita Test Script
Text-language for writing tests (e.g. unit tests, system tests, etc)

Rapita Code Injection
Text-based language for source code manipulation

RCI Example: dump function call arguments at run-time

- At run-time, create a “trace” of all subprogram calls and their parameters

- Simple Implementation is 28 lines of RCI language not including target tracing code.

```
      newmode : MODE_FILLCIRCLE
<  FUNC_SETMODE
>  FUNC_GETMODE
<  FUNC_GETMODE
>  FUNC_SETMODE
      newmode : MODE_FILLRECT
<  FUNC_SETMODE
>  FUNC_GETMODE
<  FUNC_GETMODE
>  FUNC_SETMODE
      newmode : MODE_DRAWRECT
<  FUNC_SETMODE
>  FUNC_GETMODE
<  FUNC_GETMODE
>  FUNC_SETMODE
      newmode : MODE_DRAWCIRCLE
<  FUNC_SETMODE
>  FUNC_GETMODE
<  FUNC_GETMODE
>  FUNC_DRAWCIRCLE
      x :  92
      y :  62
      r :  8
      rgb :  65280
>    FUNC_PIXEL
        x :  99
        y :  62
        rgb :  65280
<    FUNC_PIXEL
>    FUNC_PIXEL
        x :  92
        y :  69
        rgb :  65280
<    FUNC_PIXEL
>    FUNC_PIXEL
```

RCI code injection at start and end of subprograms

```
append.unique.at Package.body_loc.head
[::
   type FunctionNames_e is (
[[
for Target Package.locals {
  if Target.is_func [::
    Func_[[Target.leafname]],
  ::]
}
]]
      UndefinedFunction
   );

   GuardNesting : Integer := 0;
   type Guard(NameId : FunctionNames
      Nesting : Integer := 0;
   end record;
   procedure Initialize(Obj : in out
   procedure Finalize  (Obj : in out
```

```
for Target Package.locals {
  if Target.is_func {
    append.unique.at Target.body_loc.head [::
      tracker : Guard(Func_[[Target.leafname]]);
    begin
    [[
      for param Target.params {
        [::IndentPrint("[[param.leafname]] : " & [[param.leafname]]'Img);
        ::]
      }
    ]]
    declare
    ::]
```

```
   ---------------
   -- Simple vertical line render
   -- Draws Y <= p < Y + YSz
   procedure VLine(X, Y, YSz : Integer ; RGB : Natural) is
  tracker : Guard(Func_vline);
begin
IndentPrint("x : " & x'Img);
IndentPrint("y : " & y'Img);
IndentPrint("ysz : " & ysz'Img);
IndentPrint("rgb : " & rgb'Img);

declare

   begin
      GfxPrim.VLine(X, Y, YSz, RGB);

end;
end VLine;
```

Example: test context switching

- Do some robustness testing of context switches by forcing far more to happen than normal?
  - RCI code injection to force a context switch every line

Example: RapiTest - Unit test using private types

```
1  ---------------
2  -- Low level access functions
3
4  package Conversions is
5
6     type Speed_Data is private;
7
8     function Get_Speed(Velocity : Float)
9              return Speed_Data;
10
11 private
12
13    type Motion_Type is ( Stationary,
14                          Forward,
15                          Backward
16                        );
17
18    type Speed_Data is record
19       Direction : Motion_Type;
20       Speed     : Float;
21    end record;
22
23 end Conversions;
```

```
1  ---------------
2  -- Low level access functions
3
4  package body Conversions is
5
6     function Get_Speed(Velocity : Float)
7              return Speed_Data is
8        result : Speed_Data;
9     begin
10       if Velocity < 0.0 then
11          result.Speed := -Velocity;
12          result.Direction := Backward;
13       elsif Velocity > 0.0 then
14          result.Speed := Velocity;
15          result.Direction := Forward;
16       else
17          result.Speed := Velocity;
18          result.Direction := Stationary;
19       end if;
20       return result;
21    end Get_Speed;
22
23 end Conversions;
```

Test Strategy (ideas)

1. Make Speed_data a non-private type?
2. Create hook functions to read/write components of speed_data?
3. Create user-code to do the test?

Example: RapiTest Unit test using private types

That's it. No hooks, probes, test points, user code
RapiTest adds any code necessary to "see" private types

```
1  ------
2  -- Lo
3
4  packa
5
6     type Speed_Data is private;
7
8     function Get_Speed(Velocity : Float)
9              return Speed_Data;
10
11 private
12
13    type Motion_Type is ( Stationary,
14                          Forward,
15                          Backward
16                        );
17
```

```
5
6     function Get_Speed(Velocity : Float)
7              return Speed_Data is
8        result : Speed_Data;
9     begin
10       if Velocity < 0.0 then
11          result.Speed := -Velocity;
12          result.Direction := Backward;
13       elsif Velocity > 0.0 then
14          result.Speed := Velocity;
15          result.Direction := Forward;
16       else
```

Variable	Operation	Value	Value	Value
		2001	2002	2003
		Going Backwards	Going Forwards	It's stopped
Velocity	set	-4.1	5.2	0
return.speed	check	4.1	5.2	0
return.direction	check	Backward	Forward	Stationary

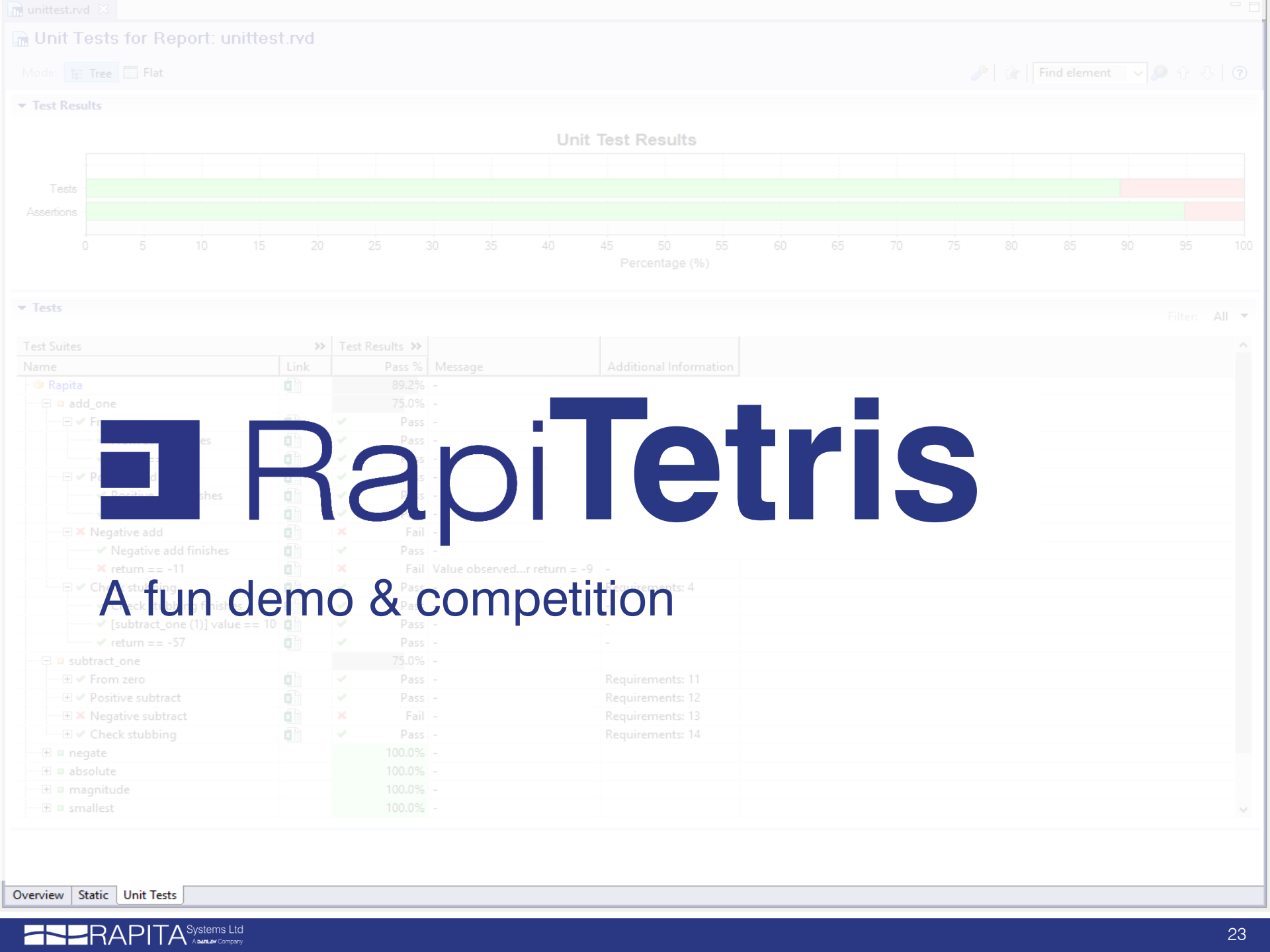
RAPITA Systems Ltd
A DANLAW Company

Multi-core Timing Analysis

An *industrial approach* for multi-core airborne software timing verification using DO-178C/CAST-32A

Ian Broster, Christos Evripidou (Rapita)
Francisco Cazorla, Enrico Mezetti, Suzana Milutinovic (BSC)


Practical DO-178C/ED-12C training workshops
Reduce the cost of compliance
San Diego, USA
25-26 Jul 2018
Bristol, UK
13-14 Nov 2018
Register »



RapiTetris

A fun demo & competition

RapiTetris

- Test your software testing skills
- Prize for highest structural code coverage
- Try on our stand
- Download and play
  - www.rapitasystems.com/TetrisComp