Specification by Example for Hardware Design and Verification

Jussi Mäkelä, Verranto AS

Introduction

- Specification by Example in SW development is documented
  - significantly reduce development iterations
  - less rework
  - higher product quality
- How issues and solutions from SW development map to HW?
- Can practices from SW development be applied to HW development?

Specification

- ”an act of identifying something precisely or of stating a precise requirement”
- A requirement specification
- A functional specification
- An implementation specification

Common Issues in Specifications

- Time consuming:
  - 20-40% of the project schedule
- Incomplete and changing:
  - The 2nd and 3rd largest factor for challenged projects
  - The largest factor for impaired projects
- Specifies Implementation but not Intention
- Missing Important Details
- Use of Pseudocode
- Outdated

Example: Long Division

"The design implements a long division for 2 unsigned integers. It divides a number (N) with a second number (D) to give the result (Q), the remainder (R), and a rounding bit (U). The rounding bit (U) is set to 1 if the fraction of the division is greater or equal to 0.5, otherwise it is set to 0."

- Exact statement of long division:
  - No Requirements / No Intention
- Missing details:
  - What if D is 0, or N and D are 0?

Use of Pseudocode

- Natural Language is ambiguous

  "The long division shifts gradually from the left to the right-hand side of the dividend, subtracting the largest possible multiple of the divisor at each stage. The multiples become the digits of the quotient and the final difference is the remainder."

- Pseudocode is not tested or verified
- Easy to copy
- Is it any easier to get right?

Example: pseudo-code vs. Python

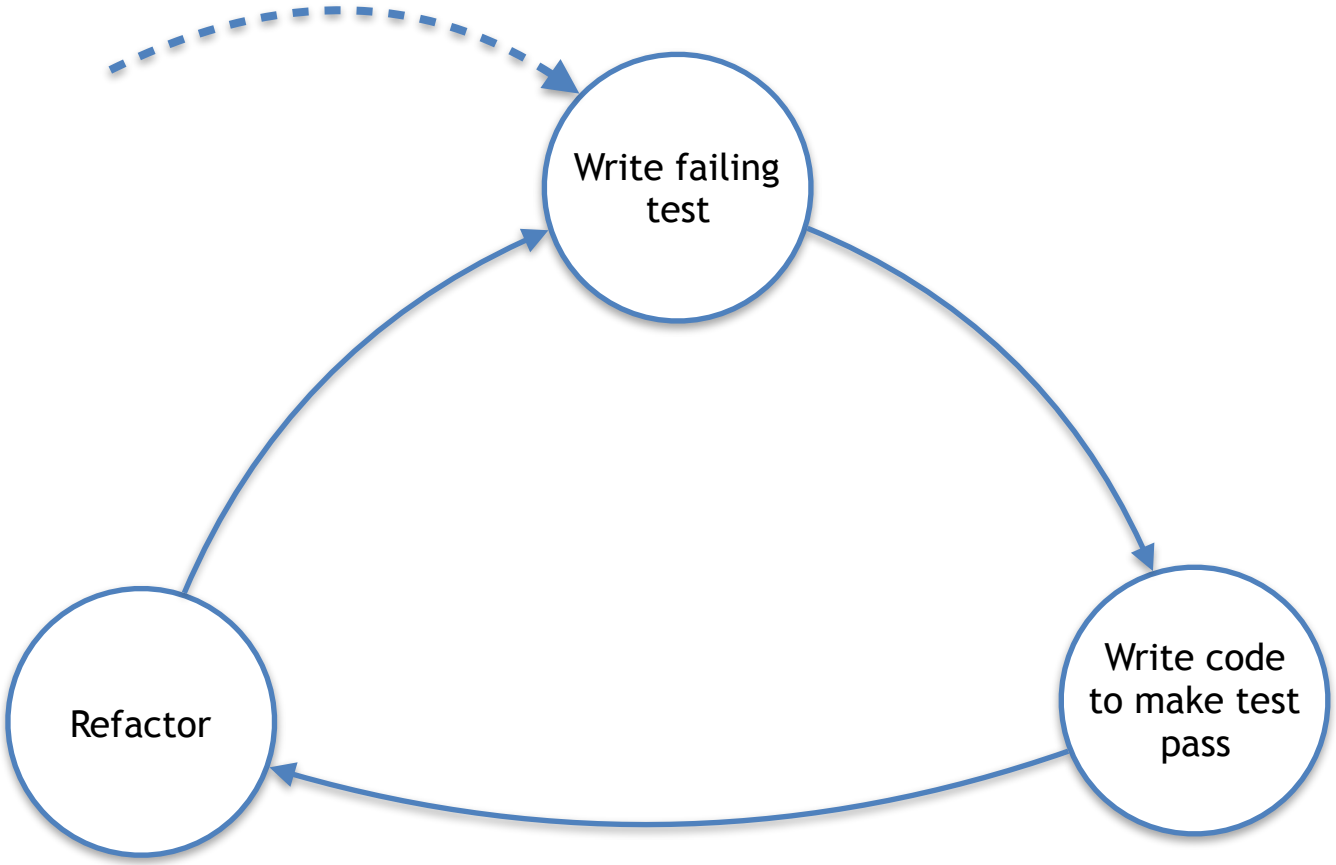
```
degree(P):
  return the index of the last non-zero element of P;
         if all elements are 0, return -∞

polynomial_long_division(N, D) returns (q, r):
  if degree(D) < 0 then error
  q ← 0
  while degree(N) ≥ degree(D)
    d ← D shifted right by (degree(N) - degree(D))
    q(degree(N) - degree(D)) ← N(degree(N)) /
d(degree(d))
    d ← d * q(degree(N) - degree(D))
    N ← N - d
  endwhile
  r ← N
  return (q, r)
```

```
def LongDivision(N, D):
  Q = 0
  R = 0
  U = 0

  for i in range(self.WIDTH-1, -1,
  -1):
    R = R << 1
    R = setBitTo(R, 0, ((N>>i) & 1))

    if R >= D:
      R = R - D
      Q = setBit(Q, i)

  if (R<<1) >= D:
    U = 1
  return Q, R, U
```

Agile Processes

- Working software over comprehensive documentation
- Collaboration over all functions
- Face-to-face conversation
- Test Driven Development
- Acceptance Test Driven Development
- Specification by Example

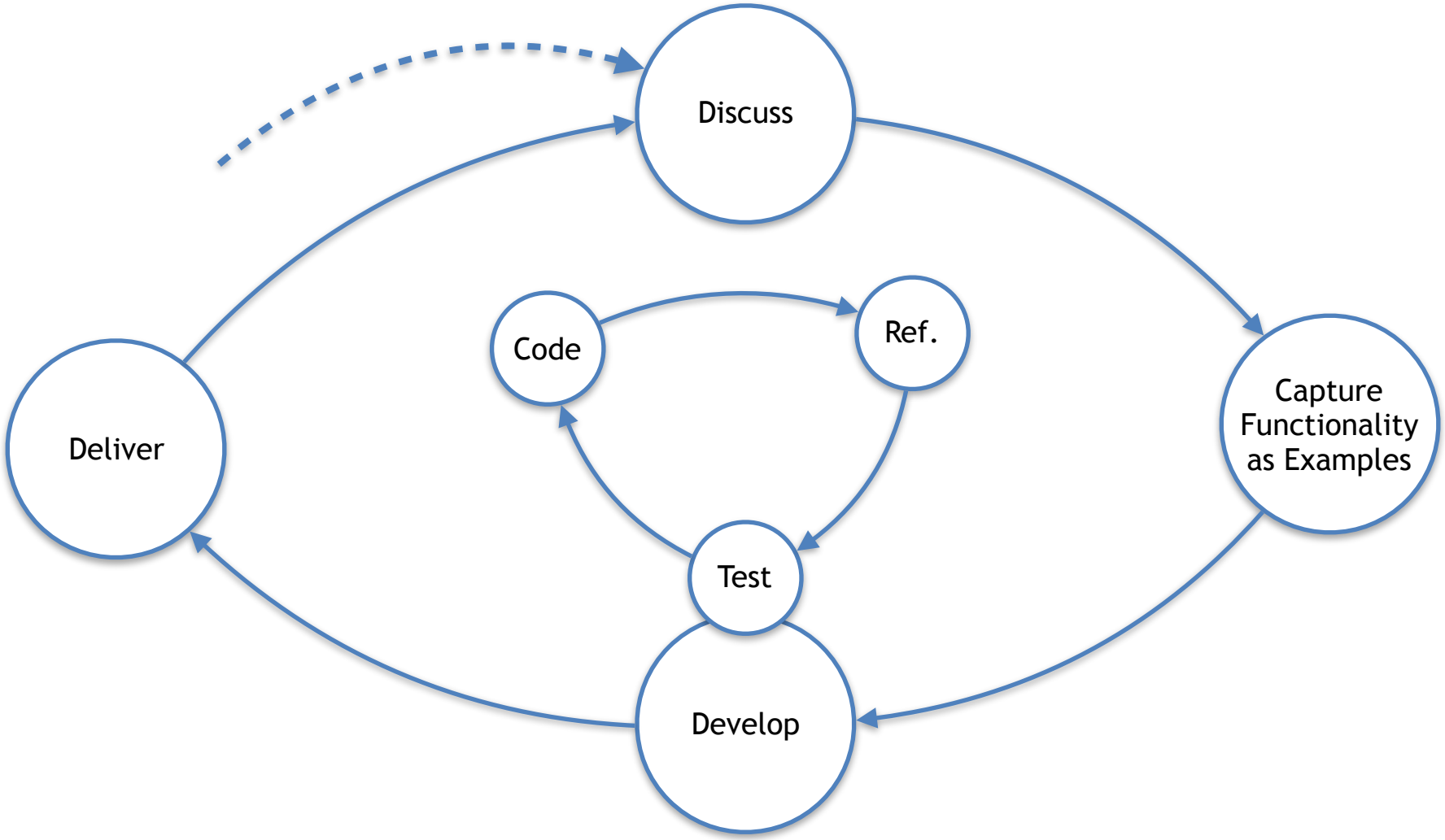
Test Driven Development



Test Driven Development for HW

- In Software Development documented to:
  - achieve better productivity
  - more consistent quality results
- No documented results in HW development, but anecdotal evidence shows similar results
- SVUnit is a unit test framework for ASIC and FPGA developers

Specification by Example



Principles of Specification by Example

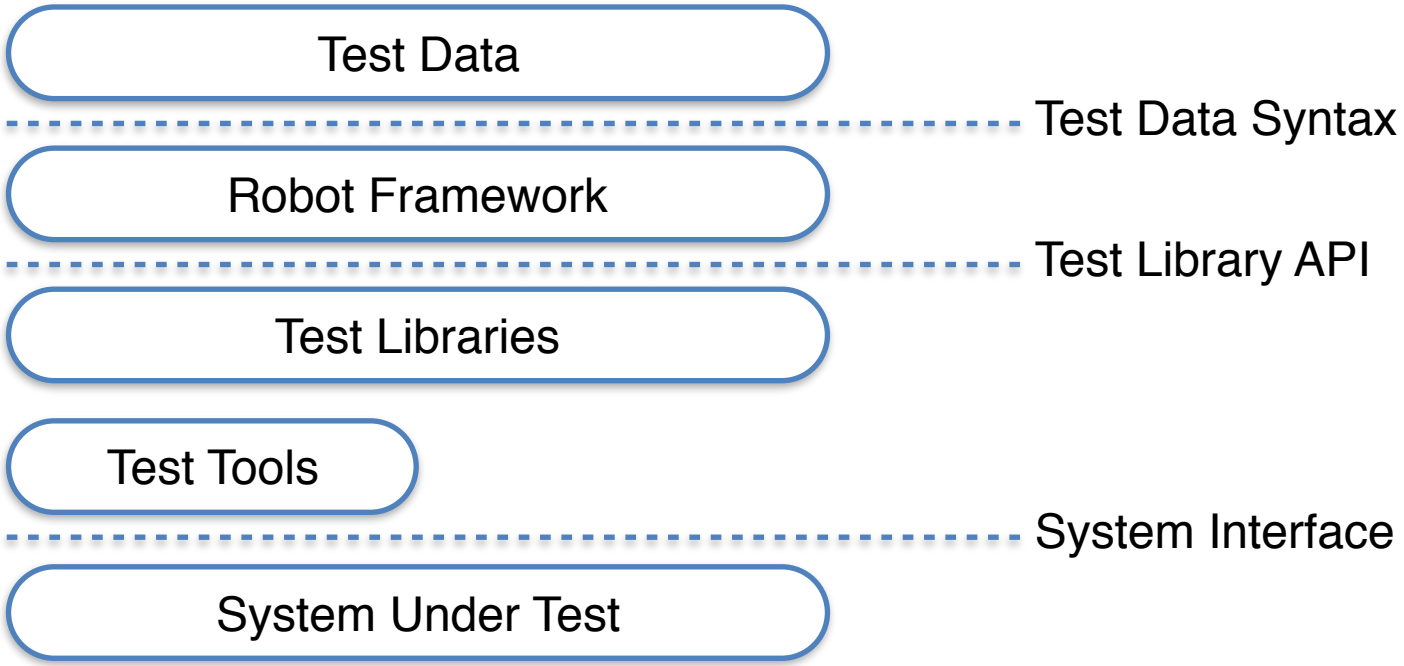
- Collaboration:
  - engage multiple team members with different background
  - encouraged to challenge
- Capture all information into a specification in form of examples
- Examples in executable form
- Automate specification validation

Automation Frameworks

- Uses High level close to natural language
  - Engages non-technical people
  - Human readable
  - Can be rendered to different formats
- Separate test data from implementation
- Executable specification
  - validate real code
- Living always up-to-date documentation

Robot Framework

- Generic test automation framework for acceptance testing
- Robot Framework Architecture



Test Format: Keywords

- Symbolize a functionality to be tested
- Captures how the data is used

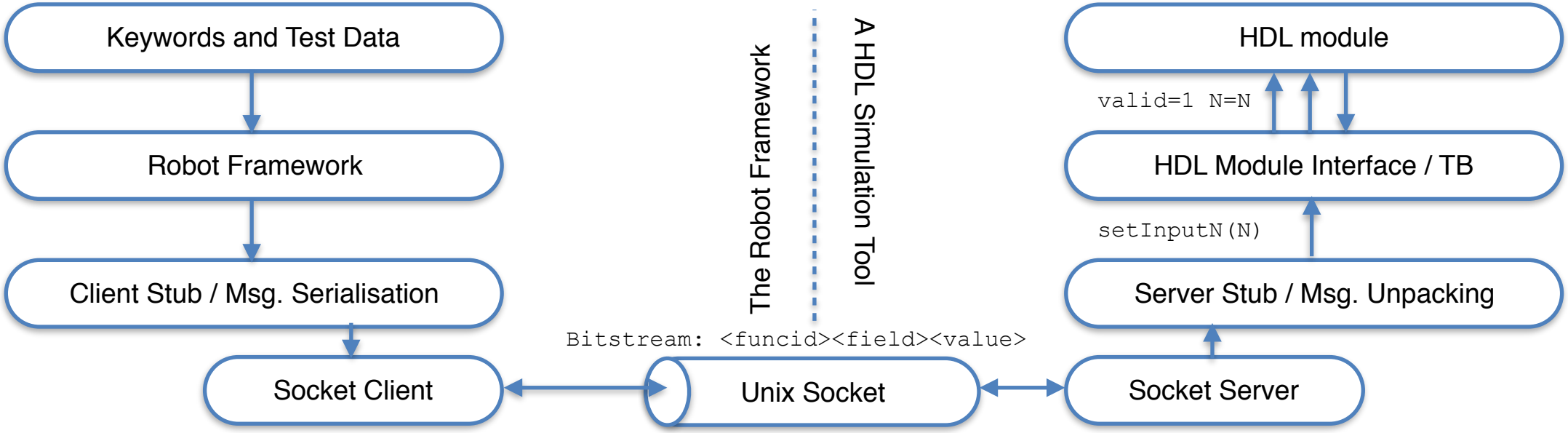
```
*** Keyword ***
Divide
   [Documentation]   Divide operation divides number N with number D which both
   ...               are whole numbers and gives result (Q), remainder (R) and
   ...               round up information (U). U is 1 if fraction is 0.5 or
   ...               larger and otherwise it is 0.
   [Arguments]       ${N}  ${D}  ${Result}  ${Remainder}  ${RoundUp}
   Set input N to ${N} and input D to ${D}
   Perform operation
   The output result should be ${Result}
   And output remainder should be ${Remainder}
   And output round up should be ${Roundup}
```

Test Format: Data Tables

- Test inputs and verifiable outputs
- Holds the functional detail

```
------------------------------------------------------------------------
Examples
------------------------------------------------------------------------

When number N is fully divisible by number D then the result should be N/D and
there should be no reminder nor roundup set.

========================  =======  =======  ========  ==========  ============
  Test Case                N        D         Result    Reminder    Roundup
========================  =======  =======  ========  ==========  ============
 No Fraction               255      255       1         0           0
========================  =======  =======  ========  ==========  ============
```

Integration with HDL Simulator



Achievements

- Use of generic test automation framework - no need for proprietary framework
- Can connect any simulator and/or language which:
  - can implement socket server
  - can implement serialisation protocol
- Implementable with DPI and VPI - simulator independent solution for Verilog/SystemVerilog

Issues

- Not (necessary) well suited for end-to-end testing
  - HW often some form of end to end testing
  - Best suited for unit-level verification of features
  - Not replacement for traditional verification methods
- For efficient use the integration must be automated:
  - Autogenerate Remote Procedure Calls (RPC) library

Conclusions And Questions

- Specification by Example is proven to be successful on SW development
- Successful results similar to results in SW are achievable in HW development:
  - Issues in SW development are very similar to issues in HW development.
  - HW can be specified with examples
  - Generic Test Automation Framework can be utilised with RTL and simulator
- Is there demand for specification by example framework for HW development?

Questions