Digital hardware design
What can we learn from software development – and what not?

Philipp Wagner @ FOSDEM 2017
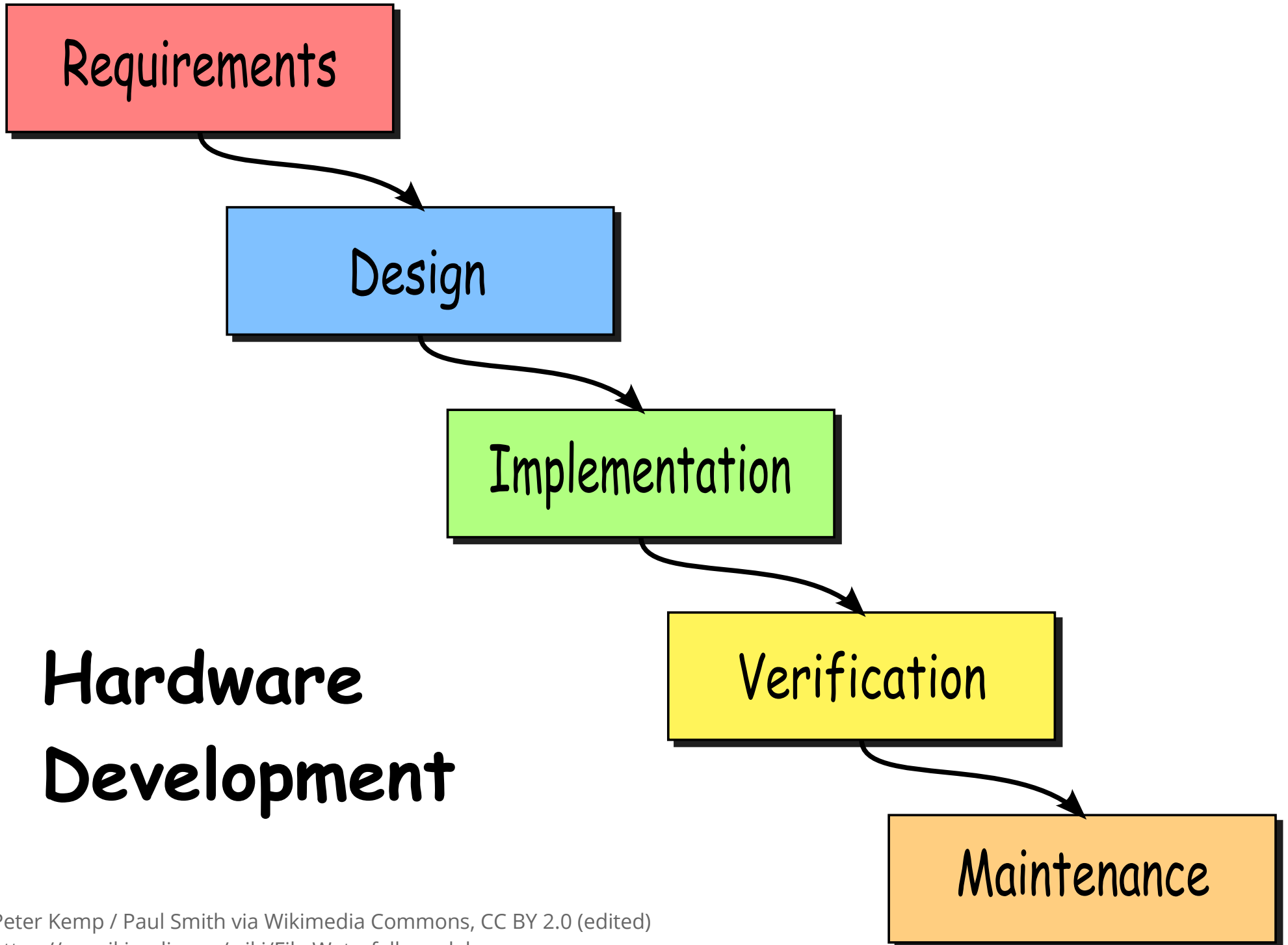
LibreCores

Free and Open Digital Hardware

FOSSi
Foundation

Hardware Development

Software Development

Can we treat FPGA-targeting hardware design like software design?

Today we look at ...

- Be less confident in quality.

- Iterate! Iterate! Iterate!

- Differentiate where it matters most.

DONT worry

Be HAPPY

confidence in quality
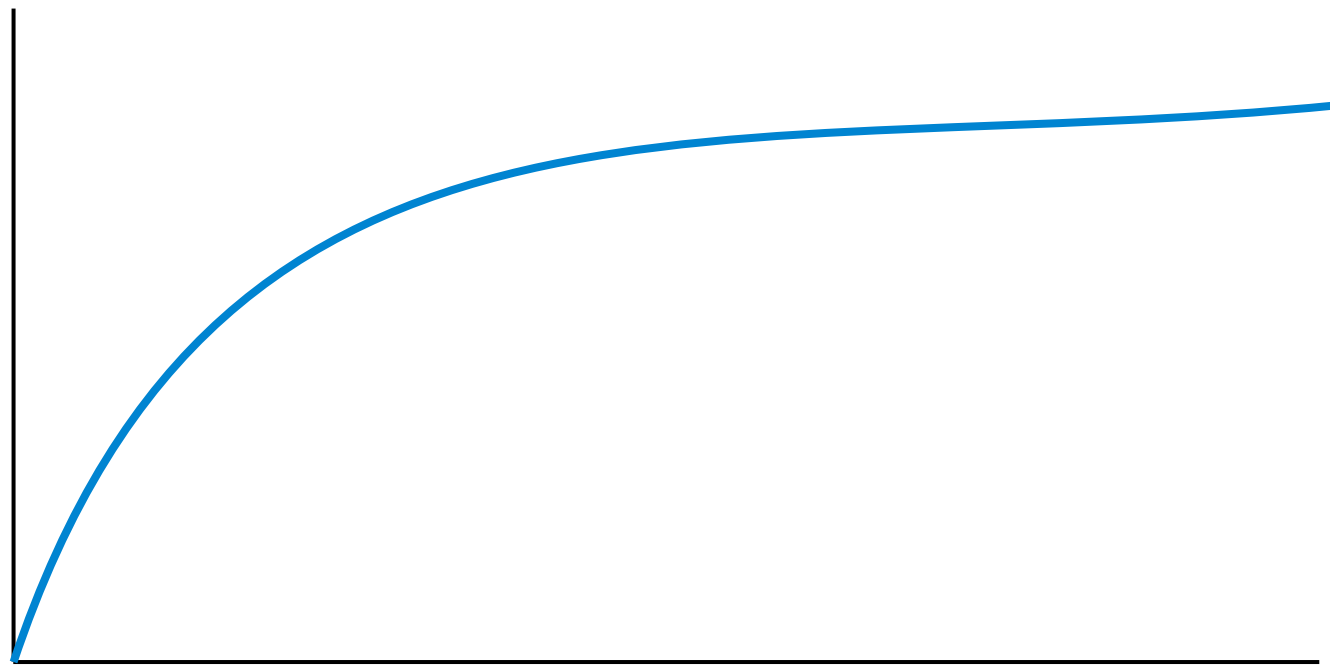
testing effort

Source: gut feeling

Be less confident in quality

- FPGAs enable cheap respins

- You don't need to get it right the first time. Win big going 80:20!

- **What does an FPGA-targeting testing and verification flow look like?**

- Prerequisite for most software development approaches.

Iterate! Iterate! Iterate!
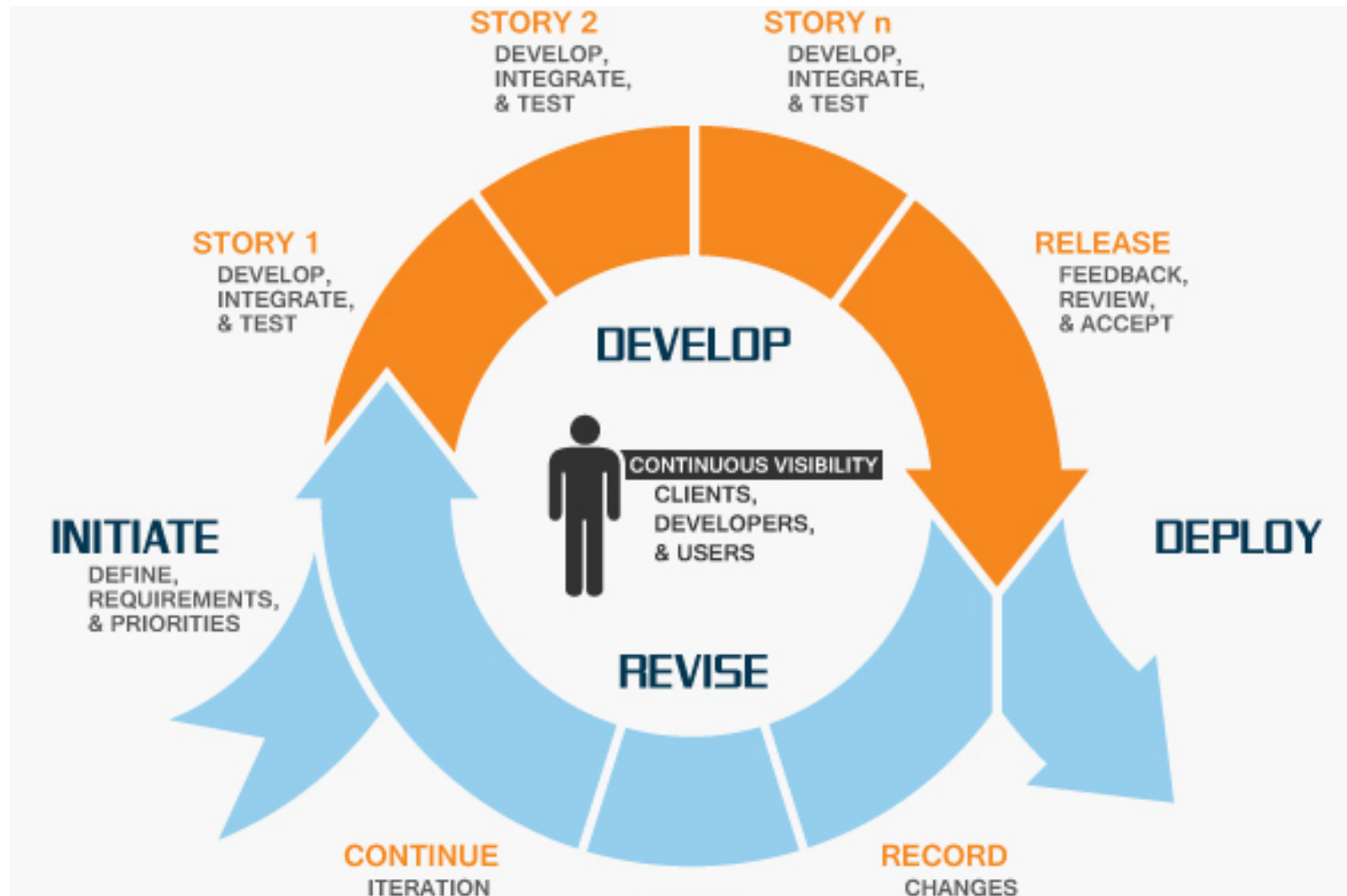
productivity = ƒ (feedback time)

GOOD THINGS
come to those

WHO
WAIT

Please

Iterate! Iterate! Iterate!

- FPGAs enable **cheap** iterations.

- But do we have the tools for **fast** iterations?

- Intelligent IDEs

- Linting and static analysis

- Automation

  – Continuous Integration, Continuous Delivery

  – Very nice side effect: reproducible results

HDS Setup Assistant Wizard

- Language
  - Verilog 2005 (Graphics)
  - VHDL 2008 (Text VHDL)
  - SystemVerilog (Text Verilog)
- Type Of Design
  - FPGA
    - Xilinx
- Simulator
  - MODEL_SIM
- Synthesis
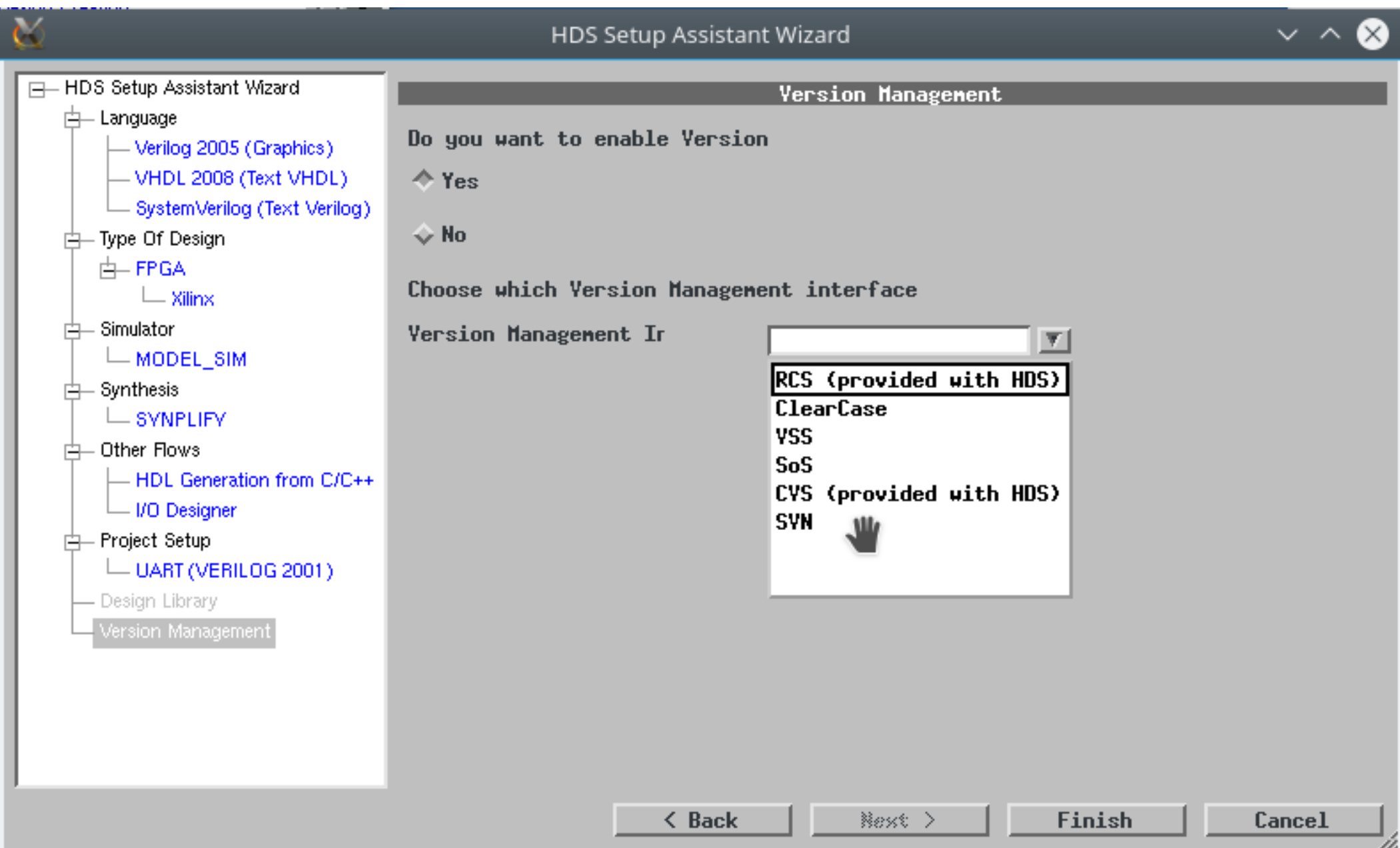 - SYNPLIFY
- Other Flows
  - HDL Generation from C/C++
  - I/O Designer
- Project Setup
  - UART (VERILOG 2001)
  - Design Library
  - Version Management

Version Management

Do you want to enable Version

◆ Yes

◇ No

Choose which Version Management interface

Version Management Ir [   ] ▼

RCS (provided with HDS)
ClearCase
VSS
SoS
CVS (provided with HDS)
SVN

< Back   Next >   Finish   Cancel

Screenshot from Mentor Graphics HDL Designer v2016.1 (own work)
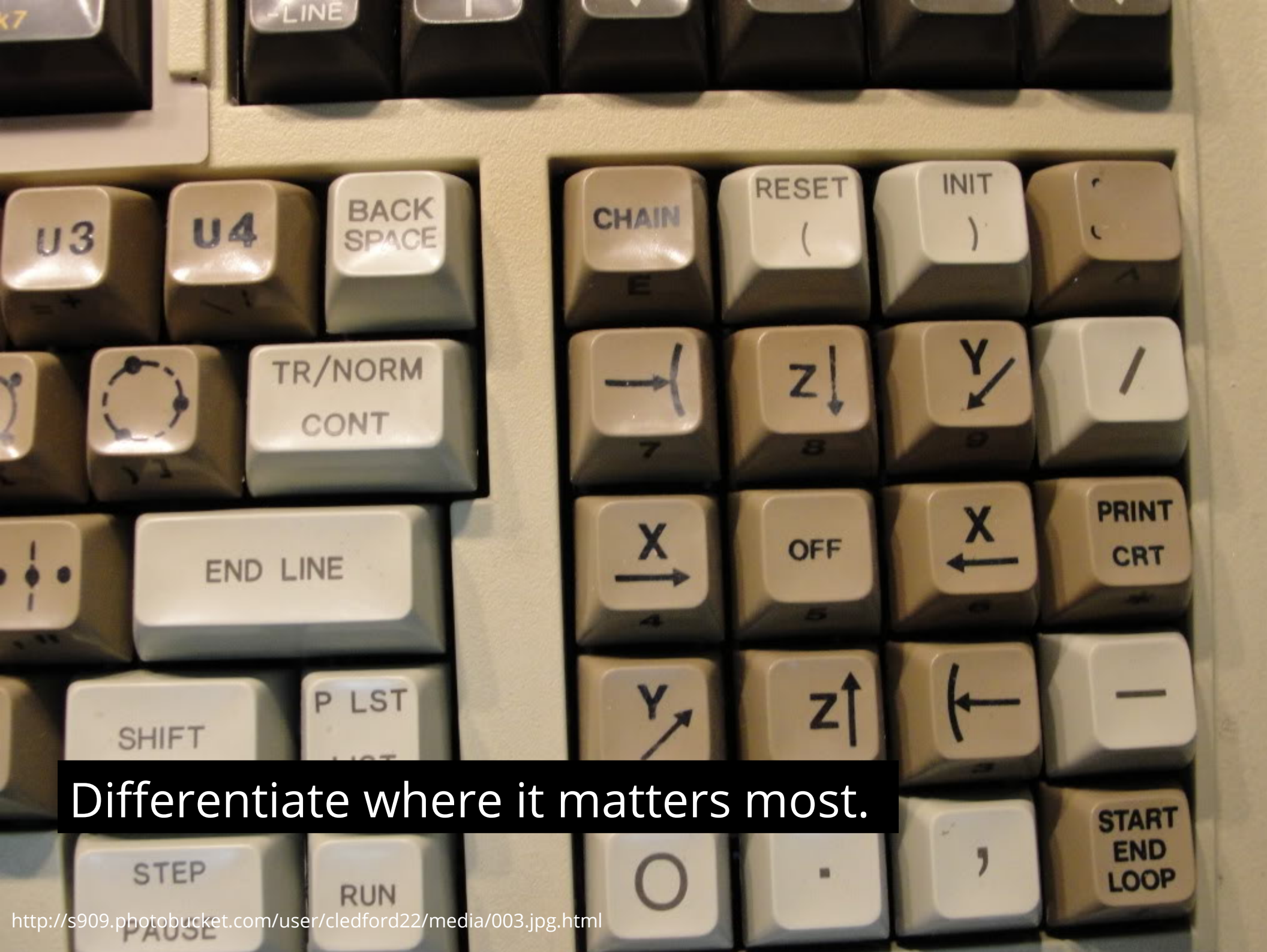
Differentiate where it matters most.

What makes your project better?

- The build system?

- The way you include dependencies?

- Your choice of programming language?

- Your coding style?

- Your FIFO implementation?

Finding common ground

- Use an intelligent HDL build system

    – fusesoc, hdlmake

- Speak a common language

- Work across tools

    – Polyfills for unsupported language features?

- Go for a common coding style

We need more collaboration!

Kevin Dooley via Flickr, CC BY 2.0
https://www.flickr.com/photos/pagedooley/8435953365

Say Hello to LibreCores!

LibreCores
We drive Free and Open Digital Hardware.

What IP is missing in your project?   🔍 Find IP Project

engage

You're not alone. Meet other developers, engage and tackle your challenges together. Join the mailing list, hop on to IRC, or read blogs.

» Community Hub

code

Express your ideas in code. Create new digital hardware designs and improve existing ones. Open them up for the world to see.

» Add a project

share

LibreCores is sharing

Share IP cores. Share knowledge. Share the fun of free and open digital hardware design.

reuse

Implement your ideas faster: build on a great stack of existing work! The LibreCores Project Repository lists IP cores and related projects.

» Browse projects

learn

Get started with VHDL, Verilog and other languages. Find out what tools you can use. Learn which licenses are best for your project.

» Read the docs

Looking Ahead

- ## Quality Metrics
  - machine-generated
    - commit statistics
    - build/test status
  - user-generated
    - reviews, likes, "I've taped this out", etc.
  - more ideas welcome!

Quality Score 5.0

No change with last release details

Community Rating 3.9

Based on 8 questions answered details

Tell us about this module

How helpful are the docs? not at all helpful

How easy to use? hard easy

Does what it promises? Y N

Works without changes? Y N

Used in production? Y N

Screenshot from Puppet Forge

Looking Ahead (II)

- Integration with package managers (like fusesoc)

  - prior art: npm, packagist, cargo, puppet forge, Linux package managers, and many more

  To use this module, add this declaration to your Puppetfile:

  ```
  mod 'puppetlabs-ciscopuppet'
  ```

  Learn more about managing modules with a Puppetfile   **download latest tar.gz**

  To manually install this module with puppet module tool:

  ```
  puppet module install puppetlabs-ciscopuppet
  ```

- document FOSSi best practices

  - license choices

  - project hosting

  - testing/verification

- LibreCores Continuous Integration (CI)
  Currently in prototyping, see https://librecores.org/static/librecores-ci for details.

Contributing

- Add your project to LibreCores

- Add your blog to Planet LibreCores
  → just open a GH issue, or write me a mail

- Write, edit and comment on documentation

- Write code

http://librecores-web.readthedocs.io/en/latest/contributing.html has a list of contribution opportunities, good first bugs, and contact information.

Isn't that a nice place?

Rochdale Canal at Hebden Bridge. Poliphilo via Wikimedia Commons, CC0

Save the date for **the** open source digital design conference

ORConf 2017

Hebden Bridge, UK

September 8 – 10, 2017
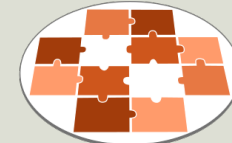
www.orconf.org

let's talk!

me

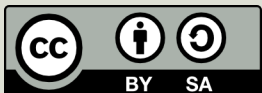Philipp Wagner

mail@philipp-wagner.com
www.philipp-wagner.com

FOSSi Foundation

discussion@lists.librecores.org

www.fossi-foundation.org
#librecores on freenode