



Sustaining Student Software

Kathryn D. Huff

University of Illinois at Urbana-Champaign, Department of Nuclear, Plasma, and Radiological Engineering, Urbana, IL 61801



Challenge

Although computational research groups build capability by leveraging and sustaining software, not all of it can (or should) be sustained. That is because :

- Student researchers are transient.
- Untrained students generate unsustainable software.
- Trained students sometimes also generate unsustainable software.
- An ever-expanding maintenance burden is unsustainable.

Introduction

As an early career faculty member, I'm learning many lessons about this challenge. Figure 1 shows a graduate student educational timeline in the context of the software development workflow, tool stack, and community strategies we use in the Advanced Reactors and Fuel Cycles group. However, **I still struggle to:**

- prioritize the various roles of the PI,
- motivate best practices,
- and responsibly manage contributor transitions.

Humans Don't Scale

Finally, I have found that known strategies for generating more sustainable software are time consuming and difficult to motivate in a research context:

- Pair programming
- Peer mentorship
- Code review by peers
- Code review by PI
- Training
- Test-driven development

Suggestions welcome!

Contact Information

- Web: <http://nppe.illinois.edu>
- Email: kdhuff@illinois.edu

Current Strategy

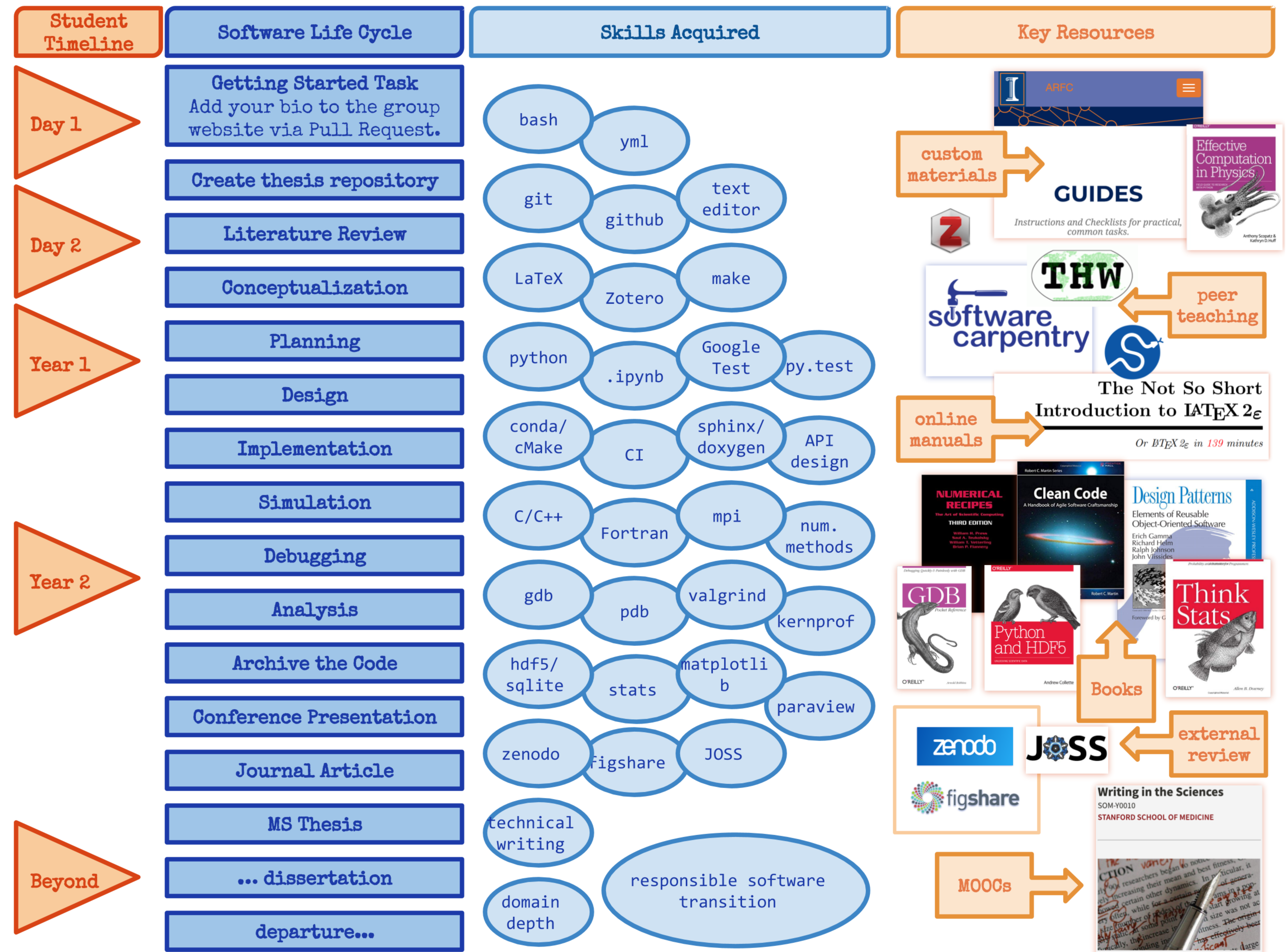


Figure 1: My current approach to software sustainability starts on day 1 in the hopes that student software transitions will be smooth.

Acknowledgements

This material is based upon work supported by the Department of Energy National Nuclear Security Administration under Award Number(s) DE-NA0002576 through the Consortium for Nonproliferation Enabling Capabilities.