

How to make software outlive the research project

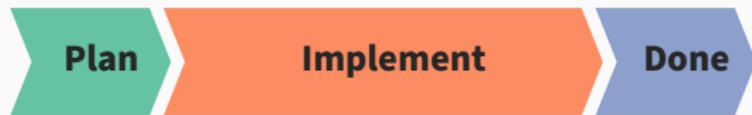
Fabian Klötzl

Future Opportunities for Software in Research 2022

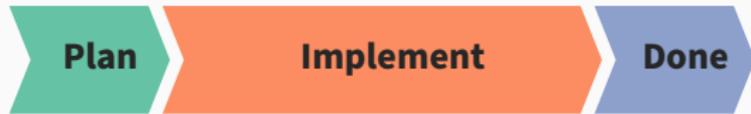
Illumina, Ltd.

Slides: kloetzl.info

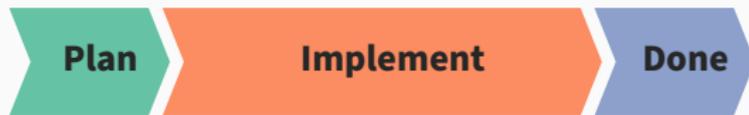
Software life cycle



Software life cycle



Software life cycle



It takes a minimum of two to three years for a piece of scientific software to become mature enough to publicize.

- Titus Brown



- bug fixes
Users will find edge cases.
- new features
Users will come up with ideas for new features.
- merge contributions
Users will fix bugs or implement new features themselves.

Requirements

- Solve a problem

The easiest way to be at the top of your field is to choose a very small field.

– Simone Giertz

Requirements

- Solve a problem
- Freely available

I have discovered a truly marvelous proof of this, which this margin is too narrow to contain.

– Pierre de Fermat

Requirements

- Solve a problem
 - Freely available
 - Allows modification
- Talk by Tobias Schlauch on Licenses

Requirements

- Solve a problem
- Freely available
- Allows modification
Talk by Tobias Schlauch on Licenses
- Release

Release early, release often.

– Eric Raymond

Requirements

- Solve a problem
- Freely available
- Allows modification
Talk by Tobias Schlauch on Licenses
- Release
- Installation, Packaging

- Be welcoming
Talk by Yo Yehudi on open source communities

All creatures are welcome / Be excellent to each other
– Chaos Computer Club

Contribute

- Be welcoming
Talk by Yo Yehudi on open source communities
- Documentation, examples

There are only two hard things in Computer Science: cache invalidation and naming things.

– Phil Karlton

- Be welcoming
Talk by Yo Yehudi on open source communities
- Documentation, examples
- Tests, CI/CD automation
Talk by Christine Muehleib on GitLab

Computers are good at following instructions, but not at reading your mind.

– Donald Knuth

Back and Forth



- Easy maintenance requires rigorous implementation.
- A sloppy implementation will lead to later problems.

I think support isn't a critical thing, in a strange way. Rather, it's the lack of need for support that's important.

– Richard Durbin

Summary

- Adhere to good software engineering principles.

Summary

- Adhere to good software engineering principles.

Call to Action

- Lead by example
- Lower entry threshold
- Educate policy makers to appreciate software
- Evangelize

It takes a lot of time to implement a prototype. Then it actually takes a lot of time to really make it better.

– Heng Li