Working software

Core tenets

- Developing software is really hard and mostly not enjoyable.
- Sometimes it is enjoyable, however it is still very hard.
- Without a lot of effort invested software will behave badly, look weird or both

Software development can become enjoyable if developers really start caring about the product. This is harder to achieve if the product is complex. Smart people used to avoid too much complexity. We now have to learn how to embrace the right kind of complexity.

Developer happiness mostly happens when you:

- are young and naive (or just optimistic and that's awesome ;)
- start developing something of your own
- start influencing real change in the world together with a team that cares

Not all software is created equal

We have as many purposes for software as we have purposes for anything, really. If we say vehicle is a vehicle or that any software is designed to work under similar conditions or for similar people etc. then this is like saying that a space rocket is similar to a submarine because both of them are vehicles that are supposed to get you from A to B (and back!) safely.

What types of software are out there?

Many many many. It's really a mess! There is a criticial sections of green zones that work and are maintaned reliably. This is called **working software**.

Working software can have any combination of these properties:

Category	**A**	vs **B**
Age	Old	Modern
Processing	Localized in one place	Distributed
Usability errors / crashes	Strive towards none	Don't care much
Openness	Closed source	Open source
Price for code	Free	Some business model
Support	Free	Some business model
Modularity	Modular	Monolithic
Dependence	Very dependent on other technologies or systems	As independent / resilient as possible (but **still integrated with other synergetic software and systems**).

But there is a tendency in the world so that BLUE choices are preferred and make for a **higher chance of having a long-term working system for many decades**.