Full Life Cycle Use Case

- How will users use the product?

- How will users discover a need or opportunity to do something different?
- How will users find out about the product?
- How will users analyze the product?
- How will users acquire the product?
- How will users install/set up/integrate the product?
- How will users use the product?
- How will users determine the value gained from the product?
- How will users pay for the product?
- How will users receive support for the product?
- How will users spread awareness of the product?

Full life cycle use case

What are all the things the user has to do? Obstacles: awareness, effort and risk.

Don’t presume customer enthusiasm.

Different use cases for different markets/customers.

Details!

Product == feature.

How will users discover a need or opportunity?

Users won’t go looking for a solution for a problem they don’t know they have.

Users won’t be receptive to ads for things they don’t need.

Users who’ve rolled their own solution have recognized the need, but may feel it has already been adequately addressed.

How will users find out about the project?

Once users feel a need, where will they look?

What Google search keywords will they use?

What people will they ask?

Who sets best practices in the field?

How will users analyze the product?

Even if it’s free in both speech and beer, integrating software with their existing software and workflow takes effort and introduces risk.

How will users recognize the value of your product?

Do users think the product actually works? Can you be relied upon to support it?

Different markets have different thresholds.

Default is to do nothing.

How will users acquire the product?

Software distribution is easier than physical distribution, but it can still cause friction, just when the user has decided to try the product.

GitHub, package managers, your own project website?

Flash drives or discs given away by influencers or at conferences?

Software as a service/”cloud”? How do users sign up and how is their usage metered?

How will users install/set up/integrate the product?

Pre-built binaries? Package managers? Dependencies?

Do you have getting-started documentation?

Is there a pre-built VM with everything already configured?

If you’re offering software as a service, how do users import their data?

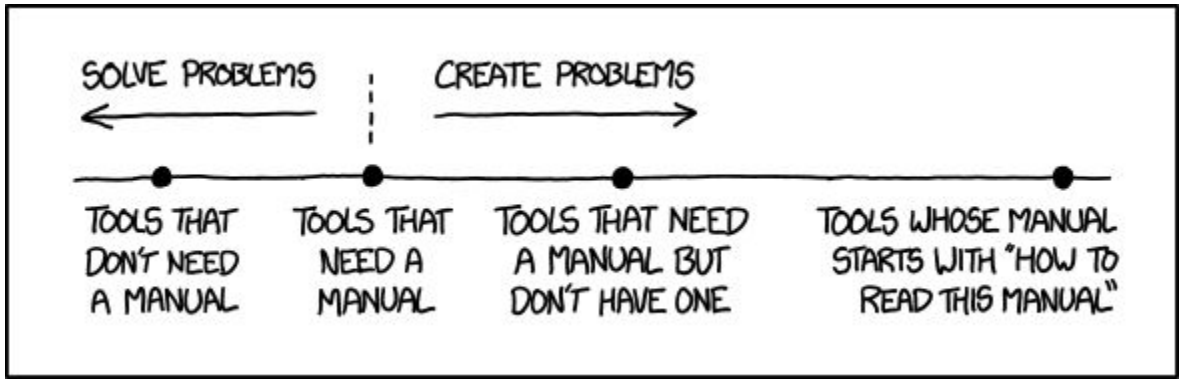
Simple demos are useful to demonstrate the product works, even if more configuration is necessary for real uses.

Users aren’t invested, so they will give up quickly.

How will users use the product?

Think like a user. Don’t presume enthusiasm.

How will users determine value gained?



Does your product just replace one problem with another?

Would it have been easier for your user to roll their own?

Cheerleaders or naysayers?

How will users pay for the product?

The software may not cost money, but using it is not truly free.

How do you minimize ongoing costs?

Will users contribute to the product?

Will users share alike?

How will users receive support for the product?

Self-help documentation? Developer mailing list? User mailing list? Web forum? Stack Overflow?

Where do users file bugs? Are bugs actually fixed?

User vs. contributor documentation.

How will users spread awareness of the product?

Will satisfied users talk about the project?

Who are the influencers and will they promote the product?

Can you establish *de facto* standards, such as file formats?

High-Level Specification

High-level specification

A user-level overview of the product, suitable for feedback.

Not a coding specification!

Helps your team (including mentor) agree on what you're actually building.

Iterate with your potential customers. If you build it, will they actually use it?

What are the specific benefits of your product?

You're not selling the product yet.

What’s in a spec?

Storyboards/mockups showing the user workflow. Can the user identify what the product will do? Are they surprised?

Example code and APIs. No implementation yet, but the user can try coding against the API to find what’s missing (and what can be cut).

A brochure or other promotional material forces you to be brief, yet specific.