Do the simplest thing that could possibly work

The most important rule in our development is always to do the simplest thing that could possibly work. Not the most stupid thing, not something that clearly can't work. But simplicity is the most important contributor to the ability to make rapid progress.

We find that we have to remind ourselves of this rule continually. Developers like to develop, and most of us have years of experience in creating a "general solution" to whatever we're asked for. Real progress against the real problem is maximized if we just work on what the problem really is.

On the contrary, if we know we're going to need something we should build it in while we're building the object the first time. It'll save time.

- How can it possibly save time to do more rather than less? The best you can hope for is to break even. A little bad luck, and you'll come out behind.
- When you're thinking about "we're going to need this someday", you're not thinking about "we need this today". You just distracted yourself from your goal. Don't compound the error by chasing tomorrow.
- Software follows an 80/20 rule: 80 percent of the benefit comes from 20 percent of the work. Find that simplest 20 and do it.
- Studies show that developers are not really that good at predicting what will be needed. It's better to wait for a real need, and provide for it then.

Smalltalk code is extremely easy to modify. We do not have to design or build for the future. Progress is fastest if we just do what we need to do now: leave the future to the future.

On the contrary, when I'm immersed in a new object, I may see how to do something that will later be more difficult because I won't be up to speed.

- If the object is so complex that it will be hard to modify later, it is just too complex. Simplify it so that adding new capabilities will be easy, but don't make it even more complex by adding them now.
- Add commentary (a class comment or method comment) describing the key idea. Your mission is to make the fastest progress against what the problem really is, not against what the problem might be or might become.

[Home] [XP] [XP Practices Frame]

ronjeffries@acm.org
http://www.armaties.com