

Kent Beck, Cynthia Andres (2004) *extreme programming explained - embrace change*: 2nd edition. p 66f

Code and Tests

Maintain only the code and the tests as permanent artifacts. Generate other documents from the code and tests. Rely on social mechanisms to keep alive important history of the project.

Customers pay for the what the system does today and what the team can make the system do tomorrow. Any artifacts contributing to these two sources of value are themselves valuable. Everything else is waste.

Code and Tests is a practice that is easy to approach a little at a time. A complicated five-stage document-driven process can be lightened up a little at a time as the team acquires more skill. The better the team is at incremental design, the fewer design decisions it has to make up front. The clearer the quarterly cycle becomes at expressing the business priorities, the slimmer the requirement document needs to be.

The trend in software development has been just the opposite for decades. Ceremony interferes with the flow of value. The valuable decisions in software development are: What are we going to do? What aren't we going to do? and How are we going to do what we do? Bringing those decisions together so they can feed each other smooths the flow of value. Eliminating now-obsolete artifacts enables that improvement.