

Handling Requests for Unnecessary Artifacts

by Mike Cohn • 22 Comments

“Working software over comprehensive documentation.” You’ve certainly seen that statement on the Agile Manifesto. It is perhaps the most important of the Manifesto’s four value statements—working software is, after all, the reason a team has undertaken a software development effort.

[Huge, Messy Stack of Papers](#)

It is also one of the most misused parts of the Manifesto. This is the quote people cite when trying to get out of *all* documentation, which is not what the Manifesto says we should value.

Some documentation on a project can be great. But most non-agile teams write too much and talk too little. Some agile teams go to the opposite extreme, but many seem to find a good balance.

Occasionally, though, a team and product owner may disagree on the necessity of a document—usually with the product owner wanting a document and the team saying it’s not necessary. I’ve found two guidelines helpful in determining how to handle requests for various artifacts, especially documentation, on an agile project.

Guideline No. 1: If a team would produce an artifact while in the process of creating working software, that artifact is just naturally produced.

This guideline covers essentially everything a team would want to produce while on the way to building a system or product. It includes, for example, source code. It also includes any design documents, user guides and other items that the team wants to produce for the benefit of the current team, future teams maintaining the software or end users.

Guideline No. 2: If an artifact would not naturally be produced in the process of creating working software, the artifact is added to the product backlog.

The second guideline is there to cover cases when the product owner (or any other outside

stakeholder) wants an artifact produced (usually a document) that the team would not normally produce.

For example, suppose the product owner asks the team to write a document describing every table and field in the database. I've certainly seen projects where such a document has been extremely helpful. (In fact, I've both requested and written such a document before.) But, I've always seen projects where this would have been unnecessary.

If the team thought this database description document were helpful, they would produce it in the process of creating the working software. And Guideline No. 1 would apply. But if they don't think this document is necessary, they won't produce it. Unless, that is, the product owner insists, which is where Guideline No. 2 comes in.

If the product owner wants this document, the product owner creates a new product backlog item saying so. The team can then estimate the time it will take to develop this document, just like they'd estimate any other product backlog item.

Putting an estimate on creating the document makes its cost explicit. This forces a product owner to think about the opportunity cost of developing that document. The product owner will be able to ponder: *This five-point document or five points worth of new features?*

I don't know which the product owner will choose, but this approach makes the cost of that artifact explicit, allowing it to be compared with the value of additional features instead.

I'd love to know your thoughts on this. How does your team handle product owner requests for artifacts the team wouldn't naturally produce? What artifacts does your team find helpful? Please share your thoughts in the discussion below.

Posted: September 30, 2014

Tagged: product backlog, product owner, agile manifesto, documentation, artifacts, working software
