

## MINIMAL AGILE DOCUMENTATION

### *What is everyone else doing?*

A complaint I recently heard from some Agile teams I've been working with was that they didn't know what was going on with the other teams. They were all focused on their sprint work and they didn't know what other teams were doing or how they were doing it. Team members were concerned that there might not be consistency across the teams which could create headaches later on.

Agile has historically pushed back against documentation. Working software over comprehensive documentation. The writers of the Agile Manifesto were rebelling against the enormous software specifications that teams had to write before they could write a single line of code. They claimed that they were usually write-only documents that quickly grew out of date. The code was the real documentation so software specifications were not required. If someone wanted to know about what your team was doing, they could ask or take a look at your code repository and figure it out. That's true but there should be an easier way.

The Agile Modeling folks have stated that “effective documentation is a balancing act; the goal being to have just enough documentation at just the right time for just the right audience” (<http://www.agilemodeling.com/essays/agileDocumentation.htm#IssuesWithDocumentation>). Minimal Agile documentation should include answers to three questions:

- What are you doing?
- How are you doing it?
- Why are you doing it that way?

For the Agile teams I've been working with, we created a document that did just that. We call it the Implementation Review Document (IRD). It is intended to be a living artifact for what the sprint team is working on which can be revised when needed. The IRD is designed to be very lightweight and should take no more than an hour, maybe two, to write. It's a good way to check the design of what you plan to implement and get feedback from people outside of your sprint team. It can also help maintain some level of consistency across teams.

It is suggested that, when appropriate, sprint teams add a “Write and present an IRD” story (or perhaps “Revise and present IRD”) into their sprints. Ideally, they should present it before the associated work is started. The presentations have usually taken 15 to 20 minutes. Some have taken longer because there was a lot of interest and discussion. They should include it in their repository or knowledge store (e.g. Confluence).



When is an IRD appropriate? It wouldn't make much sense to write an IRD if you are doing something simple such as adding a field or button to an existing screen. An IRD is more appropriate when you are using a new approach or tool. For example, let's say that you are implementing an electronic fax feature and introducing a message queue to queue up the request. The IRD could look like this:

- **What are we doing?** Sending requests to an electronic fax gateway.
- **How are we doing it?** Send requests to a RabbitMQ message queue that are then read by a separate process that sends them to the electronic fax gateway.
- **Why are we doing it that way?** By using a message queue, we separate the initial request from the actual transmission to the electronic fax gateway, creating a better user experience and building a more scalable solution.

The documentation is succinct and readers get a quick overview of the implementation. Notice that a specific technology, in this case RabbitMQ, is included. This is important. This is how we can capture when new or different products are being introduced into the environment.

We have found that IRDs help sprint teams better understand what other teams are doing and how they are doing it. IRDs also help provide improved consistency and adherence to standards as well as monitoring what new technologies are being introduced.



This TIP was written by Lorenzo De Leon, who specializes in Financial Services and Healthcare & Life Sciences. Lorenzo welcomes comments and discussion on this topic and can be reached at [Lorenzo.deleon@trexin.com](mailto:Lorenzo.deleon@trexin.com).