

HOW TO SUCCEED AS AN AGILE TECHNICAL WRITER: 5 TIPS



Creating documentation within an Agile team is both challenging and rewarding. Agile moves fast, and emphasizes team collaboration.

Many Agile evangelists think that being Agile means no documentation. Actually, the same amount of documentation is required in Agile. The difference is when and how that documentation is created.

Consider the following tips.

1. JOIN THE DAILY STAND UPS



A stand up is a short meeting where the team reports on what they've done, what they plan to do, and anything preventing them from doing their work. If you're not attending, you'll quickly fall out of synch.

When attending stand ups:

- Focus on team activities and priorities.
- Even if there are no changes to user documentation, listen and learn.

If you are not considered part of the team, or are not invited to the stand ups:

- The project may be extremely difficult.
- Inform the Scrum Master and Product Owner.
- Explain your role, and how your absence negatively affects documentation.

2. WRITE DOCUMENTATION USER STORIES



User stories are the format used for defining software requirements in Agile. Stories take the form of:

"As a (role) I want to (do something) so I can (get something)."

User stories:

- Should not cover every detail.
- Should not include implementation details.
- Should be used to start team conversations, where details are discussed and explored.

Documentation user stories highlight your work and how that work fits into the rest of the project. Software delivery is incomplete without documentation. Good Agile teams will appreciate your contribution.

Developers often loathe working on documentation. By politely insisting that documentation user stories are an integral part of the definition of done, you'll gain respect and cooperation.

To write documentation user stories, define your audience, and consider WHY they need to read the documentation. Is the audience:

- End users who are confused?
- Developers who need to integrate and extend functionality?
- Administrators who need reference information?

What is the desired result of reading the documentation? What problem does the documentation solve?

"As an end user, I want to read user help to understand a procedure."

"As a developer, I want to read API documentation to avoid wasting time when integrating the system."

"As a buyer, I want to see the documentation to make sure the system is complete."

"As a technical support, I want documentation to troubleshoot customer issues."

3. GET SYNCHRONIZED



Many Agile teams delay documentation until after delivery. This is a mistake. Any user interaction flaws the technical writer detects are delayed, and may never get addressed. You end up scrambling to document what you cannot understand because you weren't in stand ups.

If you are not a full team member, working on the same features at the same time as the rest of the team, you'll always be behind.

You'll be the first "smoke tester" for new features. You can make sure that user stories are consistent and don't stray into "how to" territory.

A good team appreciates your non-technical perspective. They'll want to know as early as possible if something is confusing or inconsistent.

4. WORK ON DOCUMENTATION DEBT



A technical writer's work is never done. No product is ever 100% documented. Similarly, the development team always has a certain amount of backlog.

Keep track of the documentation debt. When the development team is resolving backlog, you can fill documentation gaps. Write documentation user stories, or write that missing help for advanced features.

5. USE INTELLIGENT GUESSING

Because Agile moves so fast, you may not have time to wait for the team to finish the UI before writing.



What can you do?

User stories are a great help to writers. With a good grasp of the UI, you can guess how the new features will work in advance. Just make up the steps and descriptions and be reasonable.

Then share your work with the team. Tell them in the daily stand up that:

- You've guessed how the new features will work.
- They can use this to guide their own decisions, or
- They can tell you what you've got wrong, so you can fix it.

At first, this can be daunting. "How can I know beforehand?"

Take advantage of inflated engineer egos. They love to point out errors. Encourage them to correct your guesses. Everyone saves time. They can focus on clarifying what is wrong.

This takes a big dose of both confidence and humility, but you'll quickly earn the team's respect. Yes, you'll get things wrong. But you were going to make mistakes anyway. At first the team will just focus on your errors. Eventually they'll use your guesses to guide them in developing features.

CONCLUSION

Agile is here to stay. The Agile Manifesto puts working products before extensive up front documentation, but it never says that Agile is documentation-free.

In fact, the same amount of documentation must be done. If there's no technical writer, the engineers must write it themselves (which they usually hate), or it doesn't get done until a customer complains. At that point it becomes a real challenge to figure out after the fact why the feature ended up as it was delivered.

Joining an Agile team as a technical writer isn't an option. It's a necessity. Using these five tips should help.



Fred Williams is a highly regarded speaker, writer, and teacher. Fred recently emceed the Agile Prague conference. He wrote the free Technical Writing Course at Wikiversity, certifies technical writers world-wide, and is a specialist in creating integrated Waterfall/Agile requirements specifications. Able to speak several languages, he now lives in Prague, Czech Republic, where he founded the Open Learning Center and Williams Technical. His latest projects, which include an innovative process for integrating remote technical writers into Agile teams, have been well received by the Agile community, software startups, and government institutions.

Fred is a featured speaker in an upcoming Content Wrangler webinar. Find out more here: <http://saiffsolutions.com/home/technical-writing-outsourcing-in-agile>

Fred is available to train your team!

You can learn more about Fred, and watch some free videos, at <http://www.williamstechnical.cz/>

Contact Saiff Solutions, Inc. for all of your technical writing and training needs:
<http://www.saiffsolutions.com/>