magazine for international information management
tcworld
December 2016

By Keith Schengili-Roberts

Image: © portishead1/istockphoto.com

Ten reasons why DITA and Agile are made for each other

DITA's topic-based approach to enable content reuse complements the principles of Agile methodologies. And that's by far not the only reason why the two principles make a perfect match.

Both the Darwin Information Typing Architecture (DITA) and Agile were born out of necessity for software development teams. Technical writers within the software division at IBM established DITA in order to efficiently create effective and collaboratively written topic-based documentation. Similarly, the *Agile Manifesto* came from a group of software developers seeking a more lightweight way to create their deliverables. In just over ten years of existence, DITA has seen the highest adoption rates within software development – the same industry in which Agile has had the largest uptake.

Agile software development makes specific demands on documentation teams, whose technical writers are required to be nimble, describe features in a piecemeal fashion, and report on their progress. The structure of DITA is ideally suited to these needs. Though originally created for different reasons, DITA and Agile share common roots in the software development world, and DITA has proved that Agile for documentation teams is possible.

The need for collaborative technical writing

Arguably the most significant evolutionary steps in modern technical writing developed in tandem with the growth of the software industry. Desktop publication programs made it possible for authors to create technical documents from beginning to end. This in turn opened up possibilities for collaboration.

As the volume of content increased – particularly for complex software projects – collaborative documentation practices became a necessity. It was no longer one writer producing a single document, but instead several writers producing multiple documents. As documentation teams grew, there was a real need to coordinate technical writing projects. JoAnn Hackos' *Information Development* (2006) was not only the first book devoted solely to the subject of documentation management, but it also referred to Agile-based documentation processes: "It is most important for information project managers to recognize that traditional project management methods must be adapted to Agile projects." Not coincidentally, DITA had just been released the year before.

Keith Schengili-Roberts works for IXIASOFT as a DITA Information Architect. He also runs the popular DITAWriter.com website, and teaches Information Architecture at the University of Toronto. He has been working with DITA for over ten years.

keith.roberts[at]ixiasoft.com
Twitter: @KeithIXIASOFT
www.ixiasoft.com

Enter Agile

Back in the winter of 2001, 17 software developers got together in a remote ski resort to talk about the common issues that they faced with traditional development practices. They published the *Agile Manifesto*, which focused on "uncovering better ways of developing software by doing it and helping others do it". All of the various "flavors" of Agile – Lean, Scrum, Kanban, Extreme Programming and others – are all ultimately derived from the common principles laid out in the original manifesto, each providing their own take on how to create software more efficiently than by using the waterfall method.

Agile thrives in environments where short release cycles are possible, and the interviews I have conducted with members of technical documentation teams who are working within Agile confirm this. Most people I spoke with worked either at a software firm or within the software division of a company in a different industry sector. For example, I learned of cases where Agile and DITA were used together in the Medical Devices and Heavy Machinery sectors, but in both cases the push for Agile came from the software divisions within these firms. It appears that Agile is rarely used in highly regulated environments or in those with long development times, such as Heavy Machinery.

Using data from VersionOne's *10th Annual State of Agile Report* (2016) and comparing it with my own findings from DITA-using firms (published on ditawriter.com), I have found that there is some overlapping between firms that use DITA and Agile: At least 25 percent of all firms using DITA and Agile separately are software firms.

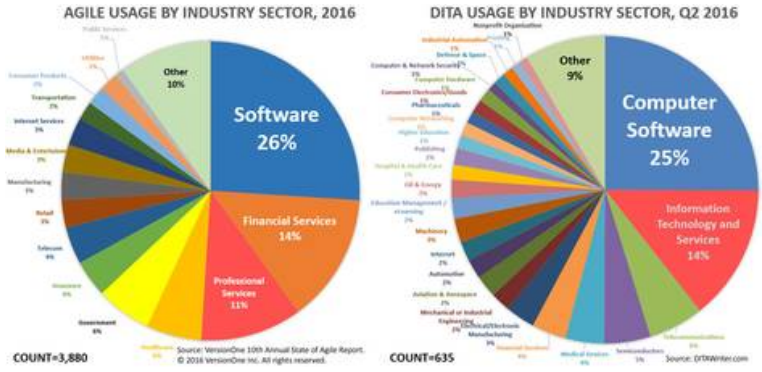

Figure 1: Agile and DITA adoption by sector
Source: DITAWriter.com

There are some significant differences elsewhere, but the strong mutual link to software and related technology sectors is apparent.

My research based on LinkedIn technical writer profile descriptions has shown that just over half of all technical writers who claim experience with DITA also have experience with Agile – a significantly higher percentage than those claiming experience with other XML standards, and even more than for Adobe FrameMaker. Obviously many people who are using DITA are doing so in an Agile environment, or have come from a place where Agile processes were used.

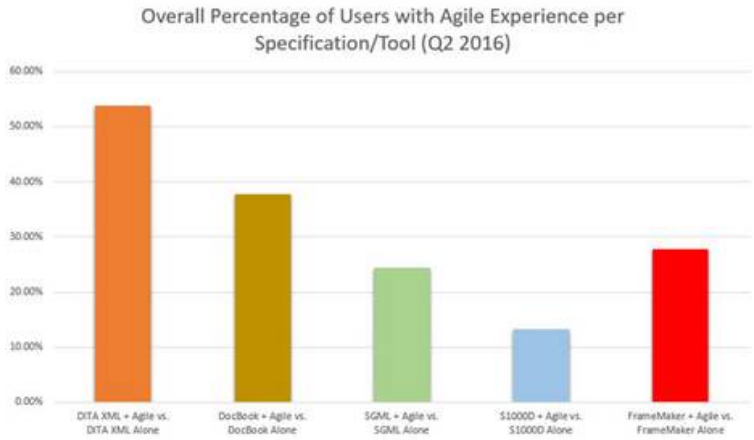

Figure 2: Percentage of technical writer profiles on LinkedIn claiming experience with Agile and specific XML standards/tools
Source: DITAWriter.com

In environments where business factors are pushing for rapid change in product development, Agile methodologies are more likely to be introduced. And where Agile exists in a business environment, it is also clear that DITA is more likely to be used to support documentation efforts.

DITA + Agile = a great partnership

There are many key factors that make DITA-based technical documentation complementary to Agile-based product development. Among others, here are ten reasons why DITA and Agile make a great partnership:

1. Topic-based approach in DITA assists with incremental development

One of the tenets of DITA is content reuse, encouraging technical writers to "write once, use many". This also means that there is no need to rewrite what already exists – a writer can simply reuse entire topics, paragraphs, or phrases used elsewhere thanks to reuse mechanisms within DITA. This enables writers to easily keep up with the rapid pace of development changes.

2. Agile user stories map well to the task topic type in DITA

Scrum-based Agile often calls upon user stories to help craft development efforts. These often take the form of various procedures that users will want to accomplish. This format fits nicely with the DITA task topic type. One common practice I found when interviewing Agile-based DITA technical writers was for them to embed the concept as the context for a task instead of writing separate concept and task topics. Additionally, Agile "epics" are collections of related user stories that comprise the complete workflow for a type of user. From a DITA standpoint, epics can be used to help refine audience-based conditional processing of content or maps (chapters) within a bookmap when an epic story hierarchy exists.

DITA 1.3 also adds troubleshooting as a new topic type, designed to provide specific solutions to scenarios that are likely to arise, and how to solve them. This new DITA topic type is perfect for writers looking for a troubleshooting option for user stories where a task may not be an appropriate solution.

3. DITA best practices advocate that content is focused squarely on the user

Technical writers are able to provide early feedback on products through their active use of the product. In this way, technical writers often become an advocate for users; this in turn helps define realistic user stories. The constant change and iterations of content over multiple releases forces a change in the typical writer's mindset from "document everything" to "document only what the user needs". Again, the granular, topic-based nature of DITA helps to make this possible.

4. Individual topics can be counted, allowing for documentation project measurement

In a typical Scrum-based Agile environment, everyone involved in a project gathers in regular meetings to discuss progress. Using a traditional DTP-based approach, all that can typically be reported is the word count or the number of chapters completed since the last sprint. With DITA, it is possible to match development features to individual topics, making it easier to report in a more realistic manner on documentation progress. If a CCMS is used, workflow status – draft, in review, done, etc. – can also be measured and reported at the Scrum meeting.

5. The DITA best practice of minimalism reduces "waste"

One of the key concepts of Lean Management – an Agile methodology – is to reduce waste wherever possible. This is encapsulated in the Japanese term "muda". In the case of documentation, this refers to content that is unnecessary in order for the customer to use the product. Technical writers can use it as a check against writing "filler" – typically background or marketing-related content that a user does not need in order to accomplish a particular task or action. One of the philosophical underpinnings of DITA is minimalism, which similarly tells writers to trim content to its essentials.

6. Topic reuse improves content consistency

The various content reuse mechanisms in DITA – topic, conrefs, and keys – contribute to greater consistency in documentation output generated within an Agile environment. Due to short deadlines and time constraints within Agile environments, it is more convenient for technical writers to reuse content where it exists, and DITA provides easy mechanisms for accomplishing this. In many business environments, if you have a topic and it has been reviewed and approved, and you want to reuse it elsewhere, it does not have to be sent out for review again, as it has already been approved. (The exceptions are regulated environments where all content must be reviewed in context, but even here content reuse speeds up the approval processes).

7. The separation of content from formatting saves time

Thanks to the separation of content formatting built into DITA, technical writers can focus on creating content rather than formatting it. This can save a considerable amount of time. An informal survey I did several years ago with a team of technical writers using a popular DTP software package to produce their documentation showed that roughly half of their time was spent formatting content. Now this time can be used for writing more Agile content in a DITA-based environment. This also eliminates

the wasted time where SMEs comment on formatting instead of on the content they are supposed to be reviewing.

8. Agile encourages continuous feedback; topic-based review is easier

Due to shorter review times, it is easier for developers to review a topic rather than a chapter produced by a DTP tool. In this way, documentation can also support broader communication between teams, customers, audit processes, etc. Work cycles are faster, and documentation feedback becomes more critical. One of the technical writers I interviewed for this article told me that Agile developers "left no room for procrastination, so this was an easy way for them to check this off their own task list."

9. DITA's short descriptions direct users to content

Writing short descriptions for DITA topics is already considered a best practice. It is arguably more so for Agile-based content, as it provides a means of progressive disclosure with regard to content relevancy for users when they search for information. A short description can often be similar to the intent of an Agile user story, where: "user x can do y based on z".

10. DITA makes publishing on demand to multiple formats straightforward

The DITA Open Toolkit is designed to produce documentation outputs in multiple formats, including HTML, WebHelp and PDF. It is also possible to flexibly produce documentation at the chapter or individual topic level on demand. Waiting for documentation deliverables is rarely a bottleneck in a DITA-based process.

Some parting thoughts

DITA is not a cure-all when it comes to working with Agile – during my interviews I encountered teams for whom Agile adoption within a DITA-based documentation team had failed. This usually happened due to poor communication, training, and lack of buy-in for documentation team members within the development environment. In cases where there was poor change management, some technical writing teams would revert to waterfall modes of engaging with development staff.

But in general, I discovered that the consensus in my research and interviews was that DITA made the job of working in an Agile environment possible. Though created for different purposes, the fact that DITA and Agile evolved from similar roots in the software sector means that there are common needs and approaches that complement each other. In particular, the topic-based structure of DITA, its emphasis on reuse, and the separation of formatting from content enables Agile documentation teams to keep up with the rapid pace that development teams set. In short, DITA helps make Agile-based documentation possible. One of my favorite quotes from my interviews highlights this. Nathalie Laroche, Lead Technical Writer at IXIASOFT, said: "Agile development goes hand in hand with topic writing, and I think this is why it's a perfect match for DITA. I love working in Agile! It makes my life as a writer much, much easier."