Towards an Agile authoring methodology: learning from the Lean methodology

Executive summary

- Agile development is a way of managing IT development teams and projects that creates new challenges for those involved in providing User Assistance for those products.
- As the Agile methodology is based upon the principles of Lean methodology, this white paper outlines ways in which Lean techniques could be applied to the technical writing process improving the value of technical documentation and reducing any "waste" during the technical writing process.
- We illustrate how these Lean techniques can be adopted by using some of the features within the Adobe Technical Communication Suite 4 and RoboHelp Server 9 products.

The emergence of Agile

Agile development is a way of managing IT development teams and projects that was introduced around 2001. It is a set of principles that aims to result in more successful projects by breaking a project down into a series of incremental and iterative stages. As Agile has become increasingly popular, technical communicators have needed to learn how they can work effectively within these projects.

Features of Agile

Technical communicators have traditionally worked in projects where the specification of what will be delivered has been defined and documented at the start of the project.

Key stages in a traditional project

In Agile projects, the goal is to develop robust products quickly with minimal investment in detailed, upfront design. The project and its deliverables are typically broken down into two-week development cycles, and, based on the feedback and lessons learnt from these cycles, the project adapts and changes over time. The team delivers early, frequent and continuous releases of products.
The objective is for teams to discover any problems quickly, and either correct a problem immediately or eliminate the affected feature altogether without wasting design work that would have occurred if the team had adopted a traditional project methodology. Project management relies far more on face-to-face communication and self-directing teams than on formal documentation, planning and control.

**Agile’s effect on writing**

The Agile approach can affect the work of the technical communicator, and today many technical communicators are looking at how to work effectively in this environment. Technical communicators are faced with working within a methodology that does not directly address how user documentation can be created in an effective and efficient way.

Here are some of the challenges technical communicators can face:

- The iterative release of products can lead to changing documentation requirements and the need to modify or delete documentation that has been completed. If the technical communicator were to wait until the product was close to final and complete release, they could find they had less time than before to write the documentation.
- The elimination or minimization of the written specifications for the product by the development team can mean that the technical communicator has little or no written information on how the product should work.
- The lack of specification, prototype or stated business requirement also means it can be very hard for the technical communicator to provide accurate estimates for their documentation tasks.1
- Technical communicators often have to produce documentation based on word-of-mouth descriptions or evolving sketches, prototypes and business cases. This means, when the product has been finally completed, the documentation may not be completely accurate and might need to be revised on certain places.

**Towards an Agile authoring methodology**

If Agile methodology does not directly address how to create user documentation, then perhaps technical communicators must develop an equivalent and complementary methodology an “Agile authoring” methodology.

There have already been teams and bodies that have outlined how to manage documentation projects in an Agile environment.2 However, it may be a mistake to take Agile programming as the starting point for developing it. The developers of Agile drew upon the principles of Lean manufacturing, and perhaps technical communicators should do the same.

---


2 For example, ISO/IEC/IEEE 26515:2011 Systems and software engineering developing user documentation in an Agile environment
Learning from Lean

The Lean methodology is a process for making things. Originally developed for manufacturing, today it is used in healthcare, programming and other business areas.

According to Capgemini:

At its most basic level Lean Management is a simple proposition to understand it is about "value". Every good Lean initiative starts by understanding what value means in the eyes of the customer. Lean then sets out to pinpoint where and how value is added and then focuses on ensuring that resources are targeted to deliver that value as effectively and efficiently as possible, eliminating 'waste' in the process. The result is greater emphasis on providing the value customers are willing to pay for while in many cases being able to simultaneously reduce cost.

Removing waste in the technical writing process

Lean is an adaptation of the Toyota Production System, which states the three types of waste are: "non-value adding work", "overburden" and "unevenness" (such as waiting). In Lean, waste has been redefined as eight common types.

If we look at the technical writing process, these eight Lean waste categories correspond to common documentation issues:

<table>
<thead>
<tr>
<th>Waste</th>
<th>Equivalent in technical publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waiting</td>
<td>Waiting for information from Subject Matter Experts or test versions of the product. Delays in approving and publishing content. Lost time due to late receipt of source material.</td>
</tr>
<tr>
<td>Over processing</td>
<td>Repeated content. Creating content that’s not needed. Duplication of collecting source material. Collecting of non-essential source material. Too frequent reporting of performance than necessary. Not meeting customer’s requirements.</td>
</tr>
<tr>
<td>Defects</td>
<td>Changes arising from uncaught errors. Having to make post-publishing amendments to the output generated from an authoring tool.</td>
</tr>
<tr>
<td>rework and correcting</td>
<td></td>
</tr>
<tr>
<td>Moving things</td>
<td>Finding information. Hand-offs for reviewing and translating. Reviewing data more often than necessary.</td>
</tr>
<tr>
<td>Overproduction</td>
<td>Editing/multiple drafts. Too much work. Not enough time.</td>
</tr>
<tr>
<td>Inventory</td>
<td>List of requirements and topics. New plans are produced while stakeholders are still processing previous plans.</td>
</tr>
<tr>
<td>Talent misused</td>
<td>Underutilization of people.</td>
</tr>
<tr>
<td>Motion (by people)</td>
<td>Moving from one authoring application to another. Meeting Subject Matter Experts. Attending development meetings. Meeting with inadequate attendees. Checking with new business units/product groups for their initial plans.</td>
</tr>
</tbody>
</table>

Lean helps us consider and measure wastes that may not have been considered by taking an Agile approach. We may find that leaving documenting to a late stage in the development process (in order to avoid the wastes caused by creating content that’s not needed or requires reworking) may be more wasteful than starting the project earlier (in order to minimize other wastes).

Lean can help us understand the value of documentation to the user

Ultimately, the purpose of User Assistance is also to make the user more productive and reduce the "wasteful" situations where they become stuck. A well-written document that is accessible to users can significantly improve the productivity of the end user and therefore contribute to lower operating costs, increased productivity and lower levels of rework.

From our experience, technical communicators can contribute during the development process itself, by identifying the points where the user might experience waste. We are able to provide developers with valuable feedback on the product early on in the development cycle. Also, if the root cause of a problem cannot be fixed, User Assistance can also provide a proportionate intermediate solution.

Removing waste from user documentation

Lean focuses on stripping waste; if something does not add value to the customer, it is eliminated. Using the three original types of waste, Lean can also help us identify potential areas of waste in the documentation from the user's perspective:

<table>
<thead>
<tr>
<th>Waste</th>
<th>How it can affect the user</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-value adding</td>
<td>Content that's not needed or doesn't meet their needs</td>
</tr>
<tr>
<td>Overburden</td>
<td>Content is too difficult or detailed</td>
</tr>
<tr>
<td>Unevenness</td>
<td>There are delays in finding information</td>
</tr>
</tbody>
</table>

Under Lean, you should only document what customers tell you is absolutely necessary. The Lean approach can help us identify which parts to document and which to leave out, and how we can best assist the user at the 'point of need'.

Writing in an Agile environment

Let's look at how we could apply Lean principles to writing technical documentation in an Agile environment, and illustrate how Adobe Technical Communication Suite 4 and RoboHelp Server 9 could assist us in doing this.

Synchronize to customer demands

Under Lean, the goal is to deliver only what adds value to the customer. In the context of user documentation, this implies you should only document what customers tell you is absolutely necessary. Only the information that adds value to the user, or increases their productivity, is included.

An iterative publication of content

If you have different types of users, you could publish content iteratively. This might mean, in a startup company, you write information for early adopters only at initial release (early adopters are often happy to work out some things themselves), and update the documentation for less able users when the product is adopted by a wider audience.

Rather than a single major release, information could be published incrementally. That leads to a new challenge for technical communicators: knowing which parts to document and which to leave out.

This requires an authoring system where it is easy to publish content on-demand to effortlessly create different outputs of the documentation and publish the content quickly and painlessly.

You may decide to control the publishing of the more complex variations of the content by using an enterprise Content Management System (CMS). In this case, it is important you have a robust authoring and publishing tool to integrate with the CMS. Better integration between the content authoring and management engines ensures better version control, more streamlined workflows and increased ease-of-component reusability. Adobe FrameMaker 11 offers out-of-the-box integration with popular CMSs such as EMC Documentum and Microsoft SharePoint, allowing users of these CMS solutions to manage FrameMaker binary and XML files.
Understand variations in demand

We need to be able to respond quickly to changes in customers’ needs, and have the ability to quickly reconfigure content to respond rapidly to any late changes to the delivered product’s functionality.

Variations in customer demand

Most technical communicators are familiar with topic-oriented writing: authoring concise, self-contained units of information about a specific topic. These topics can be structured and published in multiple ways to suit a variety of readers and provide the right documentation at the right time.

Adobe Technical Communications Suite 4 and RoboHelp Server 9 can help us meet these variations in customers’ demands:

- FrameMaker has comprehensive DITA support, which means you can create customized information models with automatic generation of DITA 1.2 elements (such as titles, tables of content and indexes), page rounding and numbering. You can easily create different DITA maps to suit different audiences.

  According to Anne Gentle⁴:

  DITA maps offer just-in-time assembly and output generation of documentation, which provides a huge advantage when creating Agile documentation. In an Agile environment, changes that make the software better, even at the end of an iteration, are welcomed and encouraged. But if a new function is eliminated from an iteration after you have already documented it, you just remove those topics from the DITA map for that iteration’s build.

- RoboHelp’s conditional build tags feature enables you to create filtered content, or content created with a particular audience in mind.

- Adobe RoboHelp Server allows you to host multiple Help files and Help file versions through a combination of contexts and areas.

Variations in project plans

As Agile projects change over time, so do the requirements for the documentation. Some functions may be added or deleted, and the technical communicator needs to be able to adapt to these changes.

By adopting a topic-based approach to writing, we can build a repository of topics that can be used in published documents. Any unused topics may be kept for the next time that feature is worked into a user story, and we could simply link those topics in a new DITA map and generate our document.

We can also use user-defined variables to store static global information that can be used repeatedly in projects, making information portable and simple to update. When you modify a variable or value, every occurrence of that variable or value is updated across the project.

Load leveling

Where Lean can probably have the greatest impact on the technical writing process within an Agile environment is on document scheduling. As we mentioned earlier, Lean’s focus on where waste is created may show that leaving documenting to a late stage in the development process can be more wasteful than starting the project earlier.

Level loading is one of the key foundations of Lean and its precursor, the Toyota Production System. The Lean methodology contains many techniques for smoothing out a work schedule.

Leveling allows a consistent workflow, which makes it possible to set standards and identify abnormalities. One of the key ways we can level the technical writing workload is simply to start the project earlier. Although we may need to go back and amend some of the content written at the start of the project, we will reduce other waste by being able to have a smoother workflow.

We can also monitor fluctuations in demand for technical writing resources and adjust the manning levels to match requirements. This may require you to be able to call upon an external technical writing partner to provide that resource. This can be more easily achieved if you have authoring tools that are commonly used, such as Adobe Technical Communication Suite. Alternatively, you can have team members who have the ability to contribute to the writing process—in this case, it helps to have a multi-user, collaborative authoring tool that is easy to use.

⁴ http://justwriteclick.com/2007/07/02/writingenduserdocumentationinanagiledevelopmentenvironment
Continuous (or one-piece) flow

Technical communicators typically batch their work – they often wait for a minimum number of source materials to build-up before working on them. They also send work to reviewers in batches (such as a first draft of a document). This is because batching eliminates the need for several setups - which can occur if people have to multitask, participate in more than one project, or have other calls on their time that takes them away from their main (value-creating) work.

Although it may seem counterintuitive, implementing a continuous flow system instead of working in batches could have a significant impact on reducing throughput time, minimizing waste and improving value adding activity. A continuous flow system works like people passing a bucket of water from one to the next in order to put out a fire. People only work on one unit at a time and only one unit of work moves from process to process.

In the context of technical writing, this could mean passing on content for review at a much earlier stage - perhaps even issuing content for review on a daily basis (thereby making it an integral and persistent part of the development process). It could even extend to publishing sign-off content on a topic-by-topic basis.

The benefits of continuous, one-piece, flow typically include:
- The quick detection of defects, which prevents a large batch of defects.
- Shorter leadtimes to publication.
- Reduction in bottlenecks.
- Reduction in waiting.

Against this must be balanced the potential cost of rework. If this is a greater cost, you may need to adopt the “document late” approach and delay writing topics until they do not require updating.

With Adobe Technical Communication Suite 4, you can create a workflow for reviewing content. Within RoboHelp you can publish commenting-enabled PDFs that anyone can review using the free Acrobat Reader software. You can send all or just a few of the topics in the project for review.
Authors can import the comments back into RoboHelp and choose to accept or reject comments as they see fit.

Seiton: The efficient placement and arrangement of equipment and material

In manufacturing, long changeovers from performing one type of work to another often cause interruptions to the production flow, so Lean includes techniques for ensuring processes and equipment are placed in a sequence for continuous flow.

In technical communication, we need an authoring and publishing system that is efficient, tidy, standardized and integrated.

Adobe Technical Communication Suite 4 offers cross-product integration and workflow between:

- FrameMaker and RoboHelp, Illustrator, Photoshop and Acrobat; and
- RoboHelp and Adobe Captivate.

You can integrate content from other suite components natively in FrameMaker and speed up time-to-market.

For those authoring in XML, FrameMaker offers a range of productivity features such as Intelliprompt of elements and attributes, template-based structure suggestions, automatic insertion of end tags, dynamic validation and pretty printing. The features mean you can work more efficiently and faster with large XML/DITA files and DITA maps.
Standardized processes and work

Lean has techniques for ensuring that projects are carried out in a consistent and precise manner. Through standardization, staff workers can work efficiently and without hesitation. Standardizing involves establishing a consistent approach for carrying out tasks and procedures.

Adobe Technical Communications Suite 4 contains many features for standardizing the technical publications process. For example:

- FrameMaker and RoboHelp have always been strong template-driven applications. Authors apply styles as they write. These styles are key to publishing your content, legacy and new, to your choice of 17 different file formats, from either Adobe FrameMaker or RoboHelp.
- RoboHelp’s Master Page feature enables users to create standardized HTML templates for a consistent look and feel. You can separate the actual content from the layout of the output and may do it from a single place. You define the placement of headers, footers, and placeholders for the body, breadcrumbs, and topic TOC. Master pages can include snippets and user-defined variables. You can create a topic using a master page or associate an existing topic with a master page. When you create a topic using a master page, the body content is placed in the resulting topic.
- FrameMaker’s comprehensive DITA support means you can also use DITA standards, such as Task, Reference and Concept topics. You can also use FrameMaker “silent publishing” to publish DITA and other content to online formats through FrameMaker.
- Through the integration capabilities of Adobe Technical Communication Suite 4, authors can publish technical content to a wide range of formats without leaving their preferred authoring environment by invoking RoboHelp from within FrameMaker.
- When an image is imported into an Adobe Captivate project, a link is automatically established between the version of the image within Captivate and its source file. If the source file is ever edited, in most cases every instance of the image in your project will be updated automatically.
- In Adobe Technical Communication Suite 4, FrameMaker allows you to publish directly to multiple formats in a turnkey fashion; you don’t need to switch applications when publishing.

Alerting and fixing any problems early

Within Lean, when a problem or issue is identified, everyone is empowered to stop the process, and swarm around the problem to solve it immediately.

In the context of technical writing, this could mean that the technical communicator has the right to flag when they are overburdened with work and expect assistance from their colleagues. For this to happen, the team needs to have an authoring system that enables non-professional technical communicators (such as developers) to be able to contribute content.

Within Adobe Technical Communications Suite 4, there are a number of capabilities for enabling this:

- Create content directly in RoboHelp or FrameMaker and use their abilities to manage tracked changes amongst multiple content creators. RoboHelp now has a powerful track changes facility, very similar to that introduced in FrameMaker a few releases ago. RoboHelp also has a feature that allows the reviewer or author to identify themselves, so their name will be associated with any insertions or deletions that the specific team member makes. This method requires someone to take overall responsibility, and it needs careful setting up and outlining of people’s responsibilities.
- Create content in a word processor or HTML editor, and import the content into RoboHelp. RoboHelp’s Automap styles feature automatically maps styles created in FrameMaker or Microsoft Word documents to the corresponding RoboHelp document. The styles include paragraphs, characters, tables, and headings.
- Link to content located external to your documentation. Adobe Air Help and RoboHelp’s External Content Search pod enables you to specify URLs for content available outside the Help system (for example, in blogs) and map them to terms that users are likely to search for. By curating relevant content in this way, you can optimize the search experience for users and provide easy access to user-generated content.
Minimize the need for rework

Rework (the work done to correct defects) is something we want to avoid having to do, particularly at the publishing stage. Publishing DITA content, for example, can be challenging, especially for PDF output. Classic XML publishing solutions such as the DITA Open Toolkit and XSLFO, can be complex, brittle, and require programming skills to make even simple modifications.

With Adobe Technical Communication Suite 4, you can combine DITA and non-DITA content in your workflows and create an engaging experience for your customers using the rich media capabilities. You have the power to publish high-quality outputs for multiple channels and devices.

Making “the value stream” visible

Lean places great emphasis on measuring the value of the work created and the production process itself.

Measuring and identifying “the value stream” highlights steps in the process that do not add any value, waste by overburden and waiting, waste caused by rework, waste caused by excessive checking, and so on. Many development teams are surprised by how much time and money is wasted once they’ve been made aware of it.

You could also use Adobe RoboHelp Server 9, a server-based Help solution that provides real-time end-user feedback on your Help and knowledge bases. RoboHelp Server gathers and logs data about what questions users ask while searching content and how users navigate through topics. Results are displayed in an easy-to-view graphical format for quick interpretation. Your Help system resides on a server, and you can make instant updates to your Help system content.

A/B testing

In order to understand we are meeting the customer’s demands, it is important that we have means by which we can measure the outputs and value of what we create.

This means using analytics, carrying out usability testing, and conducting A/B testing of the User Assistance itself.

An A/B test, or split test, is an experiment in which different versions of an item are offered to customers at the same time. The goal of an A/B test is to observe changes in behaviour between the two groups and to measure the impact of each version on an actionable metric.

With RoboHelp, you can insert the Analytics code in your master page, which is then applied to your entire Help project when you generate your output.5

User rating of topics

Another approach is to rate the usefulness of pages. Adobe AIR Help gives you the ability to enable users to provide ratings for an individual Help topic.

5 http://forums.adobe.com/message/2586505#2586505
Summary

In this document, we have suggested some ideas and approaches towards an Agile authoring methodology for you to consider. Reframing technical documentation in the context of the Lean methodology, could help technical communicators demonstrate its value, and identify all the areas where time and effort is wasted, to teams using Agile methodologies.

Technical communicators need to be able to respond to the requirements of users and to any changes to the project. In addition to having efficient working practices, this also means you need authoring tools that are sufficiently flexible and capable.

Adobe Technical Communication Suite 4 provides:

- Efficient technical communication workflows that span: print, PDF, and online delivery on multiple devices, in 17 output formats.
- Support for standards-based and structured authoring workflows.
- Linked document source files across applications to automatically propagate changes.
- Workgroup efficiency by sharing publishing configuration settings among multiple authors.
- A streamlined review and approval process, with the ability to import PDF comments into FrameMaker or RoboHelp.
- The power of Adobe AIR for enhanced information delivery and customer feedback.

Biography

Ellis is Director and Help Strategist at Cherryleaf Technical Authors, a technical writing services and training company based near London, in the United Kingdom. He has over fifteen years’ experience working in the field of documentation, has a BA in Business Studies, and is an Associate of the Institution of Engineering and Technology. Ellis is also an author and editor of two books: ‘How to Write Instructions’ and ‘Trends in Technical Communication’. Ranked the most the influential blogger on technical communication in Europe, Ellis is a specialist in the field of creating clear and simple information users will love.