IXIASOFT

Agile, Structured Content and Content Collaboration

Keith Schengili-Roberts

DITA Information Architect, IXIASOFT

Agenda

- Introduction
- Waterfall vs. Agile
- Popularity of DITA as a form of Structured Content
- How DITA Helps Technical Writers Working in an Agile Environment
  - Specific examples
  - Case studies
- Q/A

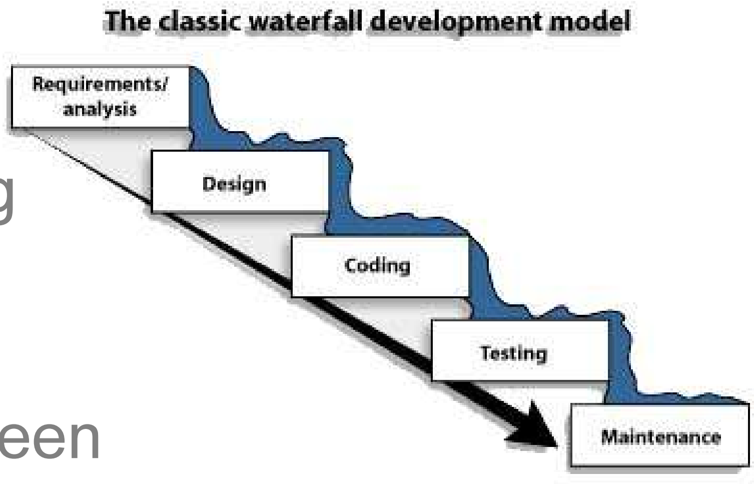
Who's This Guy?

IXIASOFT

- Keith Schengili-Roberts
- DITA Information Architect with IXIASOFT
  - Supports DITA CMS
  - Member on OASIS DITA Adoption and Technical Committees
- Also the guy behind the DITAWriter.com website
- Formerly Tech Docs Manager at AMD
- Have been using DITA for 10+ years

Waterfall Management and Technical Writing

- The waterfall model also began in the software development realm, first described in detail in early 1970s
- Sequential design process, starting with analysis and ending with maintenance (updates)
- Technical writing typically fell between Coding and Testing phases, well after Requirements and Design phases
  - "Just document what's there (or will be there."



Problems with Waterfall Management

- Waterfall-based software projects were prone to failure
  - In 1995 DoD found that of $35.7 billion spent by the organization on software, only 2 percent of the software was usable as delivered, and that 75% was either never used or was cancelled prior to delivery.

- Waterfall does not deal with changing/adapting to customer needs gracefully

Development of Agile

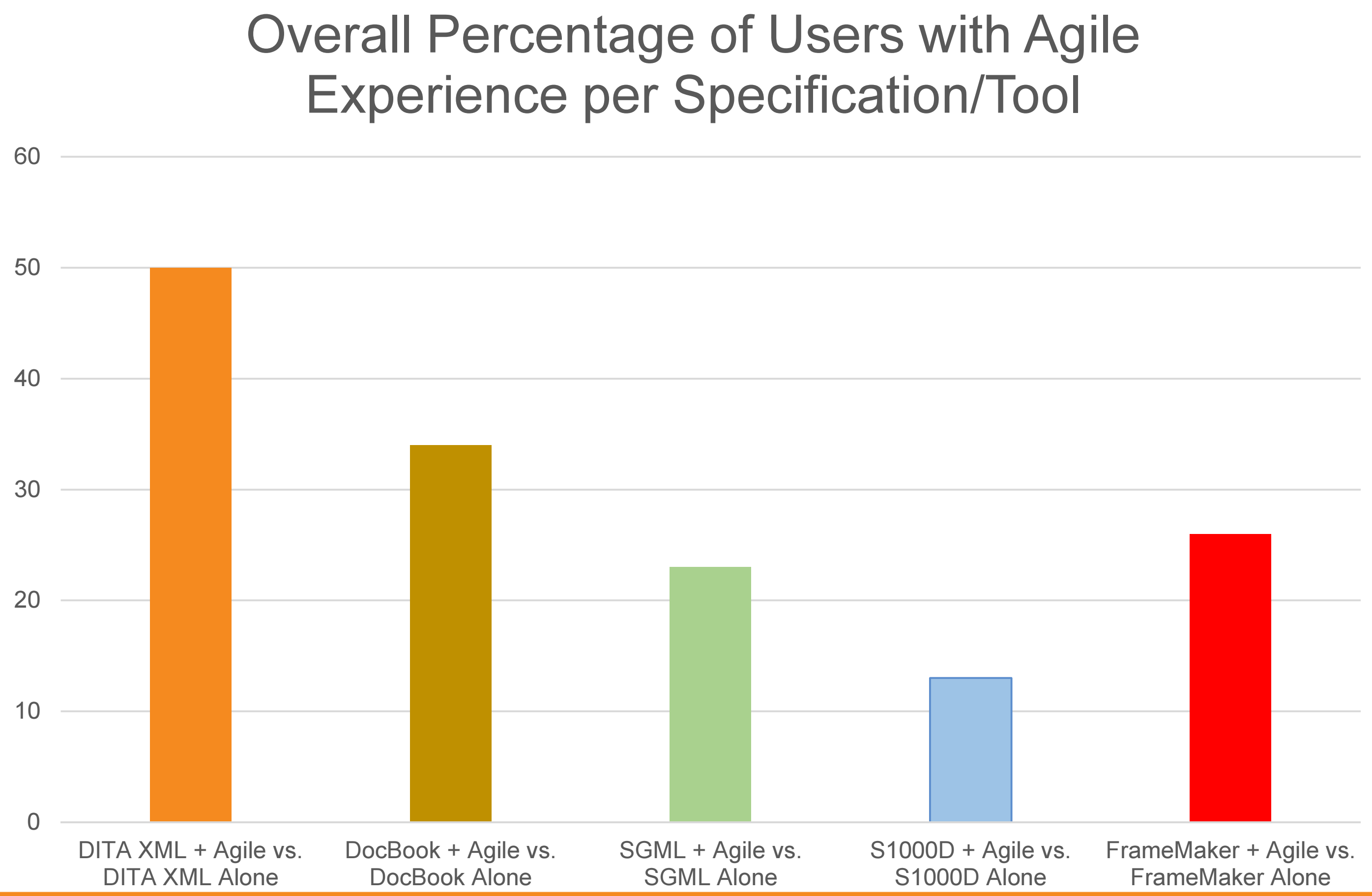
- Agile Manifesto written in 2001, advocated:
  - Releasing early and often
  - Build code daily, iterate on changes quickly
  - Embed skilled teams
- Many “flavors” of Agile have been developed since then, but the basic tenants still hold true
- For technical writers this meant:
  - Work more closely with developers
  - Provide early feedback on product
  - Constant change/iterations of content

What This Means in Practice for Tech Doc Teams

- Work more closely with developers
  - Tech writer assigned to work with one or more development teams; does regular reports on progress (Scrum with Stand-up meetings, Kanban priorities, etc.)
- Provide early feedback on product
  - Through active use of product tech writers often become advocate for users; helps define realistic user stories
- Constant change/iterations of content
  - Incremental releases, and a change of focus from “document everything” to “document only what the user needs”

DITA is Clearly Popular Among Agile Teams

- DITA is the most popular form of structured content used by Agile teams (data from LinkedIn):



IXIASOFT

Who Uses DITA?

- Used by hundreds of companies worldwide in over 60 major corporate sectors

Agile and Content Reuse in DITA

- Content reuse: “write once, use many”
  - No need to re-write what already exists
  - Content consistency
  - Single-sourcing is built in

“[DITA] handles the reuse of small information chunks brilliantly. My engineers reused functions and objects constantly as they developed new features. I found it invaluable to be able to conref (reuse by reference) previously written tables, sections, paragraphs, procedure steps, etc.. During that last long night at the end of sprint I was never too proud to reuse available writing.”

– Stan Doherty

A DITA Advantage: Separation of Form and Content

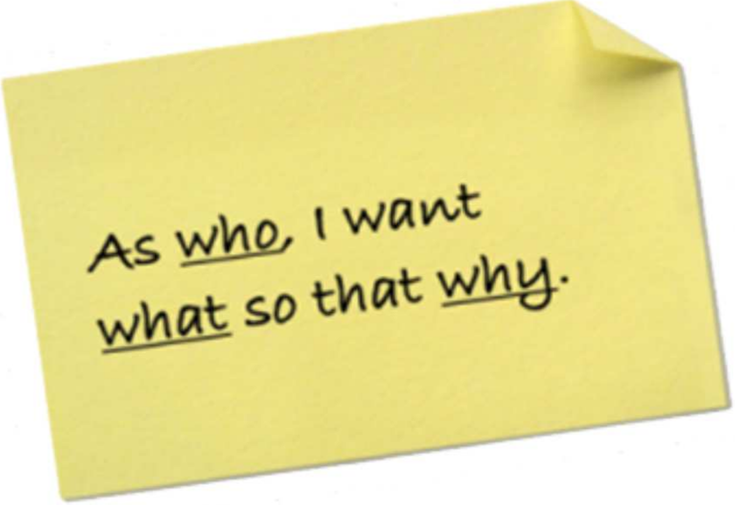
- Time is spent writing rather than formatting
  - Separating content from formatting saves considerable time

In an informal survey done on a team of technical writers using popular DTP software, roughly half of their time was spent formatting content. That time can now instead be put towards writing more Agile content in a structured XML environment

IXIASOFT

User Stories and DITA Tasks

- Scrum-based Agile often calls upon User Stories to craft development
- Often take form of various procedures that users will want to accomplish; this fits DITA's topic types nicely

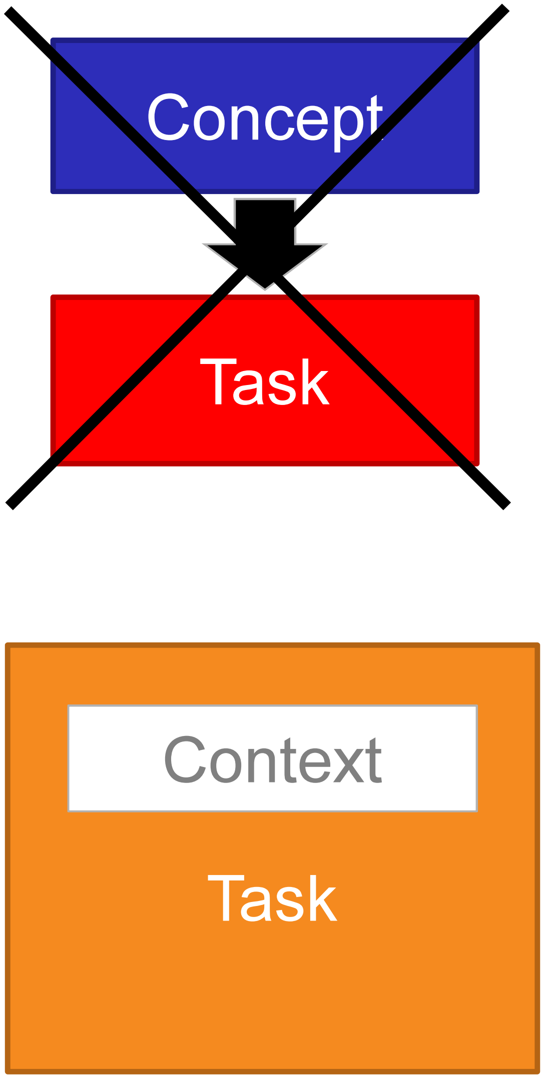


As who, I want what so that why.

"DITA allows correlating user stories to specific procedures much easier than other less granular formats. This can be utilized in some pretty creative ways to apply principals of continuous integration, and testing to documentation."
- Casey Jordan

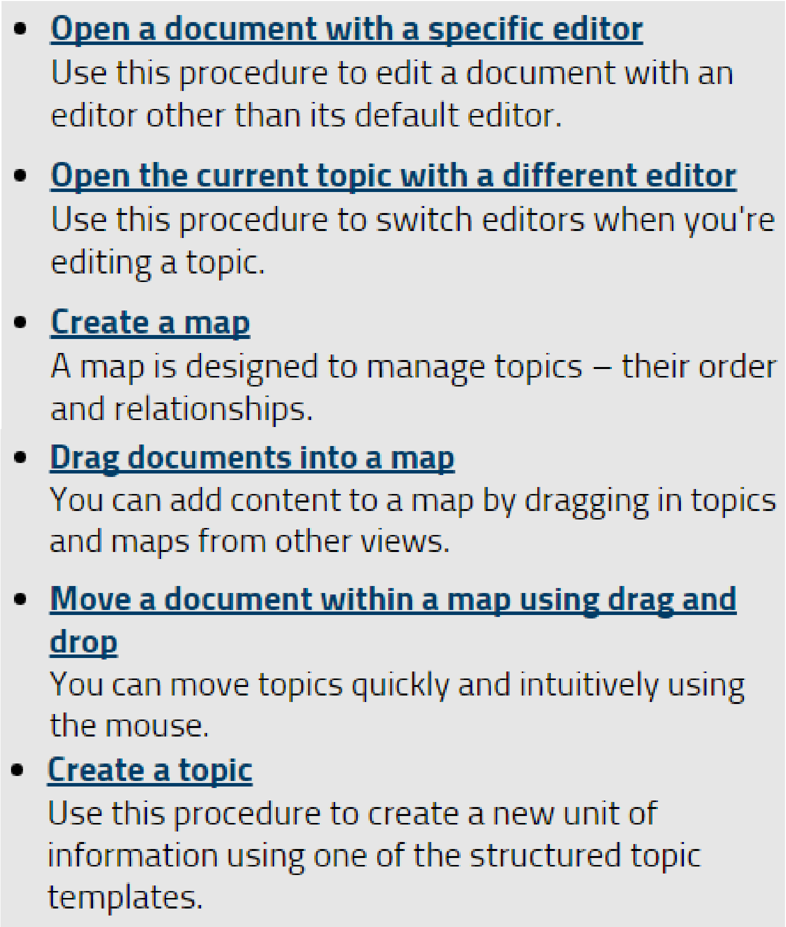
User Stories and DITA Tasks (cont.)

- Agile Best Practice for writing tasks:
  - Instead of writing a concept to be followed by a task, encapsulate that concept as the context for a task instead
  - Depending on scenario, describe expected outcomes for individual steps/conclusion
  - Use concept topics to link *between* tasks



Short Descriptions Helps Direct Users to Content

- Writing short description for DITA topics is already considered a best practice
  - Arguably more so for Agile-based content, as it provides a means of progressive disclosure as to the relevancy of content to users
  - Can be similar in intent to a user story: “User x can do y based on z”



- **Open a document with a specific editor**
  Use this procedure to edit a document with an editor other than its default editor.
- **Open the current topic with a different editor**
  Use this procedure to switch editors when you're editing a topic.
- **Create a map**
  A map is designed to manage topics – their order and relationships.
- **Drag documents into a map**
  You can add content to a map by dragging in topics and maps from other views.
- **Move a document within a map using drag and drop**
  You can move topics quickly and intuitively using the mouse.
- **Create a topic**
  Use this procedure to create a new unit of information using one of the structured topic templates.

DITA 1.3 Troubleshooting and Agile

- DITA 1.3 adds troubleshooting as a new topic type
- Designed to provide specific solutions to scenarios that are likely to arise, and how to solve them
- Will be welcomed by Agile writers who are looking for a trouble-shooting option for user stories and where a task may not be an appropriate solution

"The troubleshooting pattern of condition > cause > remedy is essentially a scenario."

- Bob Thomas

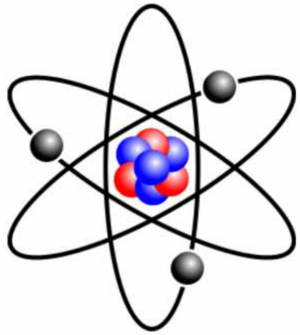
DITA Topic Granularity and Measurability

- DITA’s topic-based approach also makes it easy to measure content
  - Within a CMS it is also possible to track how “done” topics within a map are
- DTP-based docs much harder to track due to lack of this level of granularity

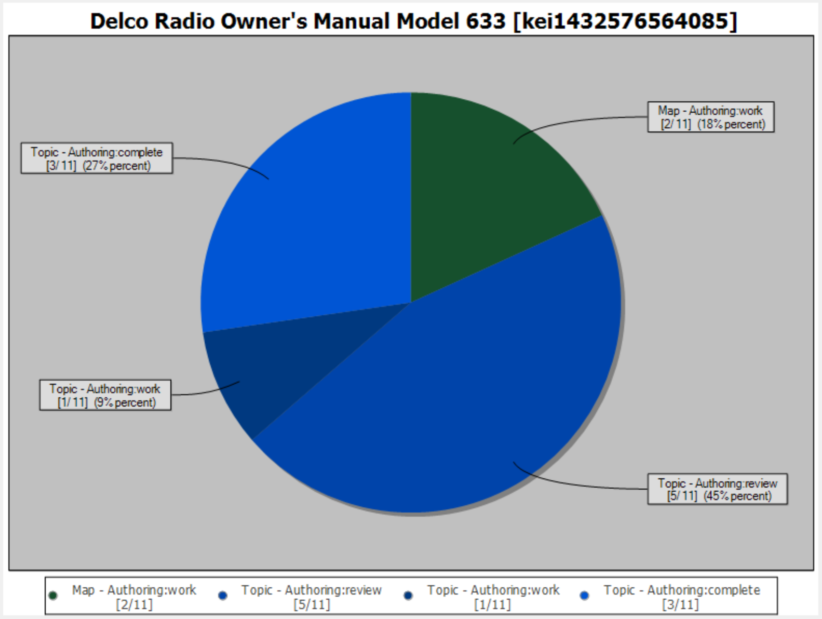
“Our project managers could track progress of documentation deliverables within our DITA-based CMS on a daily basis.”

- Jason Owen

Topic =



Production Metrics
DITA + CMS = Ability to Measure Outputs



IXIASOFT

Feedback is Part of Agile

- Documentation feedback is a developer requirement under Agile
- Using DITA, turnaround of topic-based review with SMEs much reduced
  - SMEs can provide feedback in a more timely manner

Easier to get feedback from a SME on a topic than a book(s)

"Developers would review topics on the spot in the Agile team room. Agile also left no room for procrastination, so this was an easy way for them to check this off their own task list."
- Jason Owen

Common Problems Encountered by Tech Doc Teams

Problem #1: Agile will not solve understaffed tech doc teams

Symptoms:

- Writers cannot attend stand-up meetings due to scheduling conflicts
- Writers perennially falling behind on assigned topics to write

A good Agile process manager will recognize the problem and either throttle back work or bolster effort for new hires

Common Problems Encountered by Tech Doc Teams

Problem #2: Need to make documentation “glue” for publications

- Applies to cases where a full manual is expected
- High-level introductory or conceptual material not typically accounted for in a sprint
- There’s still a need to answer the “why would you use this?” type of question

Solution is to recognize this need up front, and allow for it in the overall documentation plan

An Example of How DITA Can *Enable* Agile

Lean methodology employed at AMD; early on localization was a focus:

- Under old toolchain could only localize software (with 1 month cadence) once every *6 months*
- Using DTP-based processes, it was costly, slow and process did not allow for feedback

DITA + CMS made localizing on a monthly cadence possible

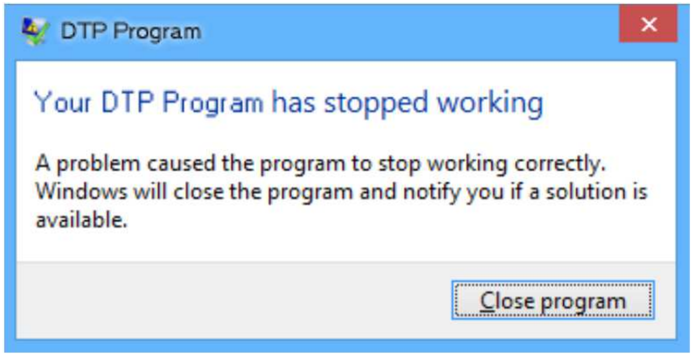
- Demonstrated considerable costs savings
- Localization staff could focus on quality and provide developers with feedback

Localization Process Pre-Lean:

Localization Process After-Lean + DITA + CMS:

DITA + Agile Case Study #1

- Client in semiconductor industry
  - Prior to move to DITA, used traditional Waterfall method
  - A “doc build” with their DTP program could take as long as a day; longer if it crashed
  - DITA opened up possibility of moving tech docs to a more Agile approach
- Results:
  - Considerable time saved by no longer dealing with formatting issues
  - Topic-based review process has greater uptake with SMEs than “here’s a whole finished manual for you to read”
  - Can now do daily “publication builds” of their content; early access given to tier 1 clients

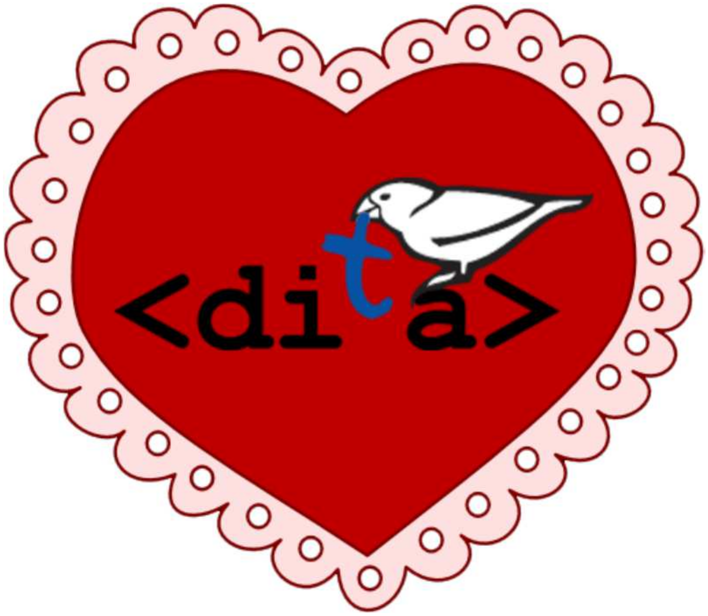

DTP Program

Your DTP Program has stopped working

A problem caused the program to stop working correctly. Windows will close the program and notify you if a solution is available.

DITA + Agile Case Study #2

- Client in Software sector
  - Writers were already embedded in software development teams, but existing DTP tools meant they were always trying to catch up
  - Lack of granularity meant that DTP-produced documents were hard to track
- Results:
  - DITA + CMS means that writers now have the time to both create content and to participate fully in the Agile process
  - Per topic progress reports now possible; now a regular part of scrum meetings, and can even be done on-the-fly on request



IXIASOFT

Some Parting Thoughts

"DITA did not directly enable or guarantee effective documentation in an Agile/SCRUM environment, but it sure saved my bacon in supporting multiple scrum teams with variant definitions of done."

- Stan Doherty

"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."

- Nathalie Laroche

Q/A

IXIASOFT

IXIASOFT

www.ixiasoft.com