



October 29, 2019

DITA Worst Practices: The Sequel



Keith Schengili-Roberts
Senior DITA Content Strategist
at IXIASOFT

Agenda

1. Introduction
2. Dud Decisions
3. Poor Processes
4. Who is Actually “Practicing” DITA?
5. Q/A



About Keith



Senior DITA
Content Strategist
at IXIASOFT



Member of OASIS DITA
Technical Committee
and LwDITA Sub-
committee



Chair of OASIS DITA
Adoption Committee



14+ years of
experience with
DITA XML



Lecturer on Information
Architecture at the
University of Toronto

How This Presentation Came About

- A few years ago I went to the “Best Practices” conference, and I got to thinking...
- How often do best practices emerge after first doing what ended up being a “worst practice?”
- I also think we learn things more readily when they are framed as a story of “things gone wrong”



This Presentation Is Crowd-sourced

- While I have some good stories of my own of DITA mishaps and adventures, I decided to reach out to my colleagues within IXIASOFT and to peers in the industry to get their take on the subject.
- In some cases the names have been changed to protect the innocent. And the “innocent.”
- In all cases, no company names are mentioned.



A Pattern Emerged...

Two main types of issues appeared:

1. DITA-specific issues (aka, “poor information architecture decisions were made”)
2. Problems with processes (aka, “how not to work effectively with people”)
 - Boundaries between categories can be fuzzy; ultimately it comes back to the people behind the issues

Stake Holders

“Who makes which bad decisions”
by role helps to clarify things:

- Worst executive/sponsor/stakeholder decisions
- Worst people management decisions
- Worst DITA Information Architect decisions
- Worst DITA toolsmith decisions
- Worst writer decisions



Original image by Jes <https://www.flickr.com/photos/mugley/310184558/>

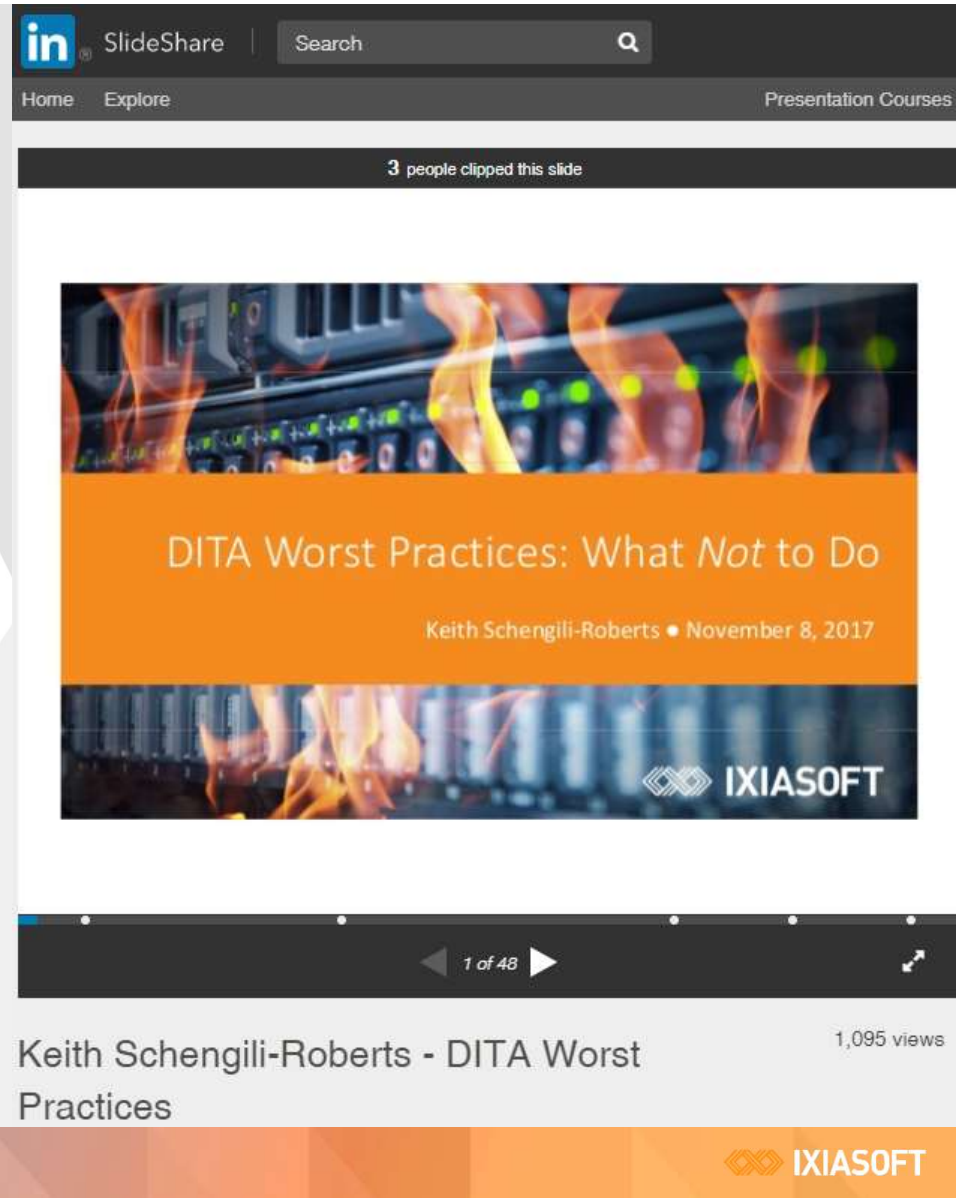
What Was Covered in “Part 1”

The following scenarios were covered in the original presentation:

1. Content Reuse is Always a Good Thing, Right? (No.)
2. Never Specialize Your Content (Wrong.)
3. If a DITA Tag Exists, We Should Use It (Nope.)
4. All DITA is Good DITA, Right? (Definitely not.)
5. A DITA Test Output Document? What’s That? (Make one!)
6. “Just Let IT Choose a CCMS for You” (Insert facepalm emoji.)
7. “Workflow Captures / Enforces All Possibilities” (So wrong.)
8. The Person with Too Many Hats (Haberdashery dashed.)
9. “They Can Learn DITA as They Use It” (Another facepalm.)
10. “We Already Have Sufficient Executive Buy-in” (Groan...)

Original is Available Online

- Original slide deck available from SlideShare at:
<https://www.slideshare.net/JackMolisani1/keith-schengiliroberts-dita-worst-practices>
- Or, to make it simpler:
<http://bit.ly/2lcRq5P>



SlideShare

Search

Home Explore Presentation Courses

3 people clipped this slide

DITA Worst Practices: What *Not* to Do

Keith Schengili-Roberts • November 8, 2017

IXIASOFT

1 of 48

Keith Schengili-Roberts - DITA Worst Practices

1,095 views

IXIASOFT

Like Any Good Sequel...

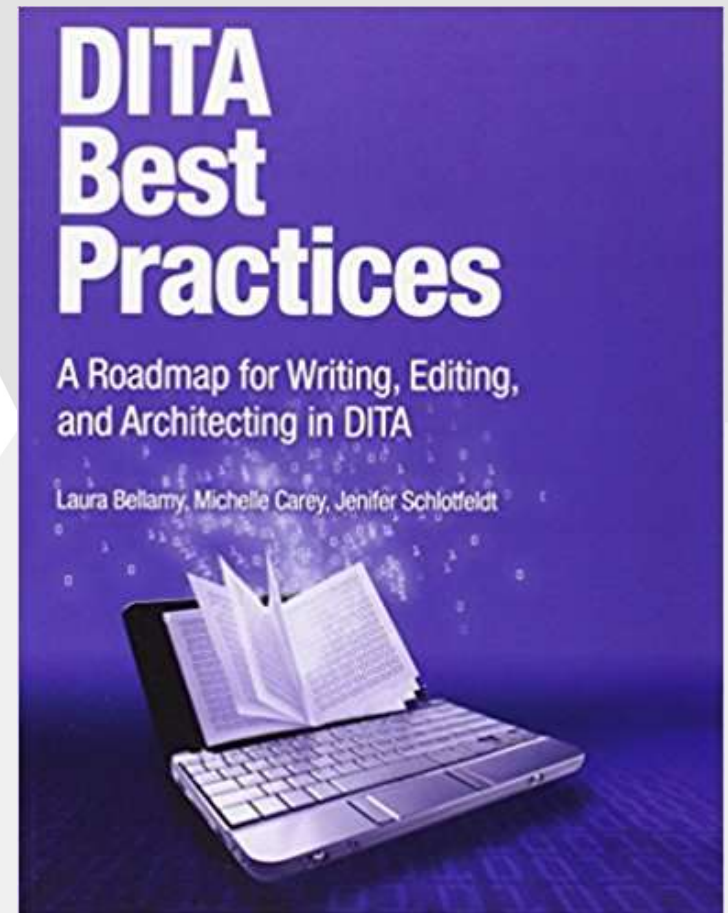
There are the return of some familiar characters and situations from last time, but now with more info:

1. Content Reuse is Always a Good Thing, Right? (No.)
2. Never Specialize Your Content (Wrong.)
3. If a DITA Tag Exists, We Should Use It (Nope.)
4. All DITA is Good DITA, Right? (Definitely not.)
5. A DITA Test Output Document? What's That? (Make one!)
6. "Just Let IT Choose a CCMS for You" (Insert facepalm emoji.)
7. "Workflow Captures / Enforces All Possibilities" (So wrong.)
8. The Person with Too Many Hats (Haberdashery dashed.)
9. "They Can Learn DITA as They Use It" (Another facepalm.)
10. "We Already Have Sufficient Executive Buy-in" (Groan...)

Ideas on DITA Best Practices are More Widely Known

- Before I get started, I just want to point out that some of the examples here are from the early days of DITA deployments
- While many of these problems keep coming up, greater dissemination of DITA best practices from people who have experience using DITA and better tools means that there is a growing “core” of people with good experience out there
- But we can still learn from “Worst Practices”

A really good book, from 2011



A middle-aged man with grey hair, wearing a dark suit, white shirt, and grey tie, is sitting at a desk. He has a distressed expression, with his right hand pressed against his forehead. In front of him is a laptop. To the left, there is a stack of books and a mouse. To the right, another laptop is visible. The background is a plain, light-colored wall.

DITA-specific Issues
(aka “Poor Information Architecture
Decisions Were Made”)

#1: Key Madness!

- At one company, an over-zealous information architect decided to make DITA keys (keydefs) to cover *every* possibility
- At the end of the process, over 1K keys were created to cover every product name, trademark, interface control, file path, etc.



This caused problems...

- The number of keys was not, in itself, the problem:
 - There was no centralized “key store” for easy access
 - Writers would create new keys if they couldn’t find the one they were looking for (duplication became rampant)
 - Some of the names grew so long that the XML editor being used could not display the full name of the key





Solution: Effective Key Management

- All keys were subsequently centralized within “key store” maps for easy access
- A concise, consistent naming scheme for key definitions was adopted
- Naming schemes enforced human-readable/-understandable key definitions
- Key definition length had to be within constraints of authoring tool

#2: Content Reuse is Always a Good Thing, Right?

Yes, but it also depends on how you do it.
Consider this example:

- A tech docs group finds that they want to use the conref mechanism for reuse purposes, such as:
 - Creating a standard way of referencing standardized phrases, such as “Click OK”
 - “Let’s use the new trademark term used in this topic and use that everywhere”
 - One writer likes a paragraph another writer has created in a topic

In all/most cases, the first version encountered is conrefed, which is in turn re-conrefed, and so on, and so on...



Welcome to “Spaghetti Conrefs”

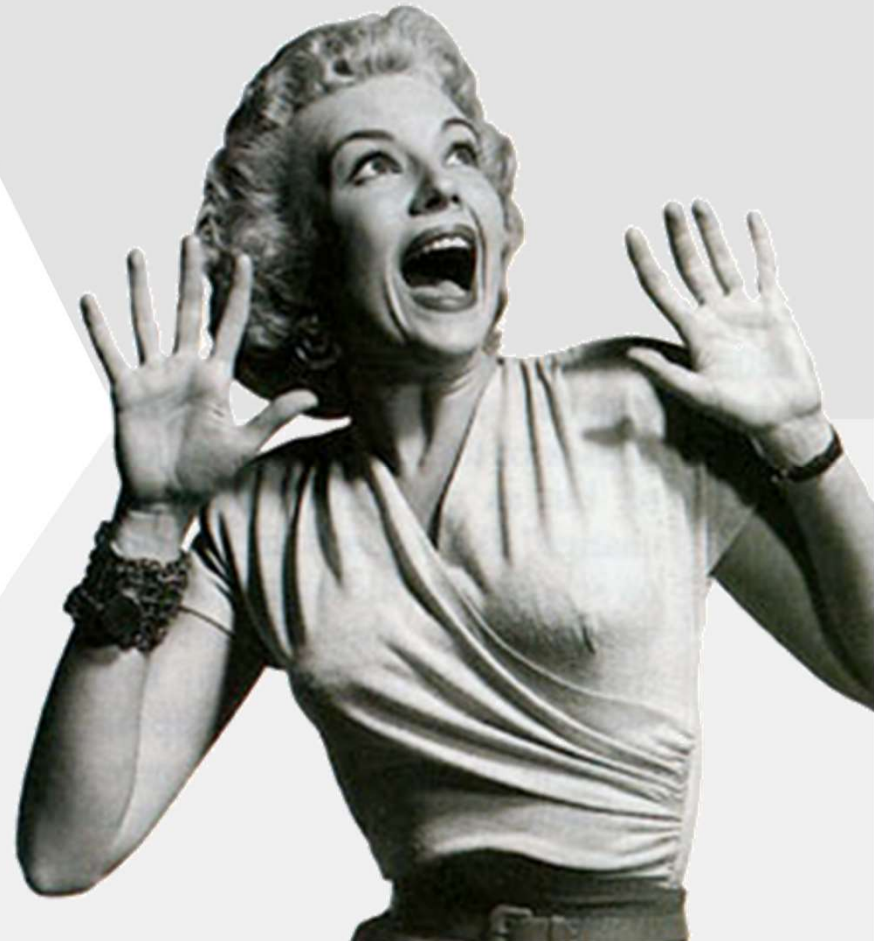
How the pain emerges:

- Someone conrefs a phrase containing a conref, which may contain another conref, and so on, and so on...
- Nobody knows the origin of the original conref and it needs to be changed...
- Somebody changes the targeted conref-ed word/phrase/block unknowingly, affecting all publications that link to it



The Horror... The Horror...

“I have seen 20K references to a single topic, in a folder containing 40K objects. I saw this when a small amount of content needed to be extracted and handed off and *10K topics having to be included for a 20-topic map, because you can never get to the bottom of the broken links to files which link to other files which link to...*”

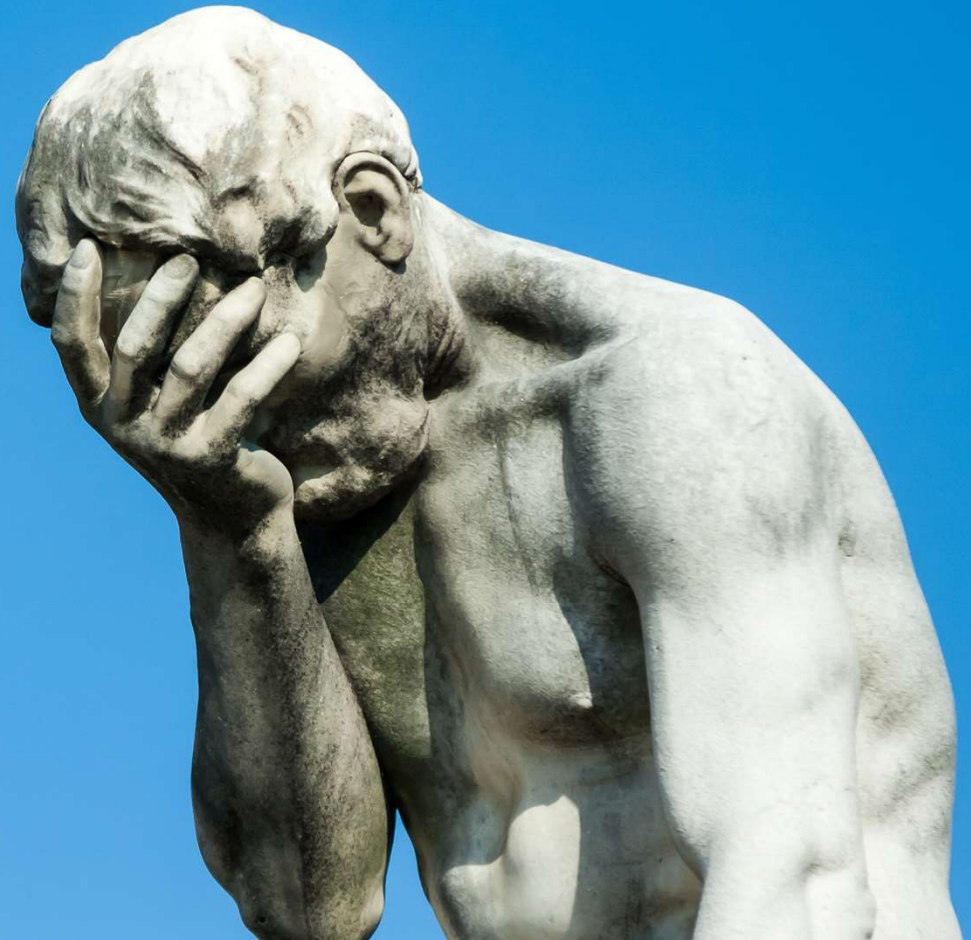


Solutions to Spaghetti Conrefs

- ~~1. Just don't do that.~~ For phrases that are commonly used, create a “conref warehouse”: this is a single file or set of files that cover off most commonly-used blocks of content
 - When people want to add new items, discuss with the group, then add accepted additions to this warehouse topic
2. For trademark terms / product / company names, consider using keys, and storing their values in a key store map

#3: Constraining Out Short Descriptions

- There was once a writing team that didn't like writing short descriptions, and didn't see the point in doing them; they got their Information Architect to constrain them away



What Could Go Wrong with That?

- Became harder for writers to re-use content, as the title of the topic was all they had to go on when searching for material to reuse; the writers found it harder to find content to reuse
- Resulting SEO is poor as users find it harder to locate content

“...of all the DITA elements, shortdesc is most like a credit card with a loyalty program that rewards you for using it.”

-Don Day

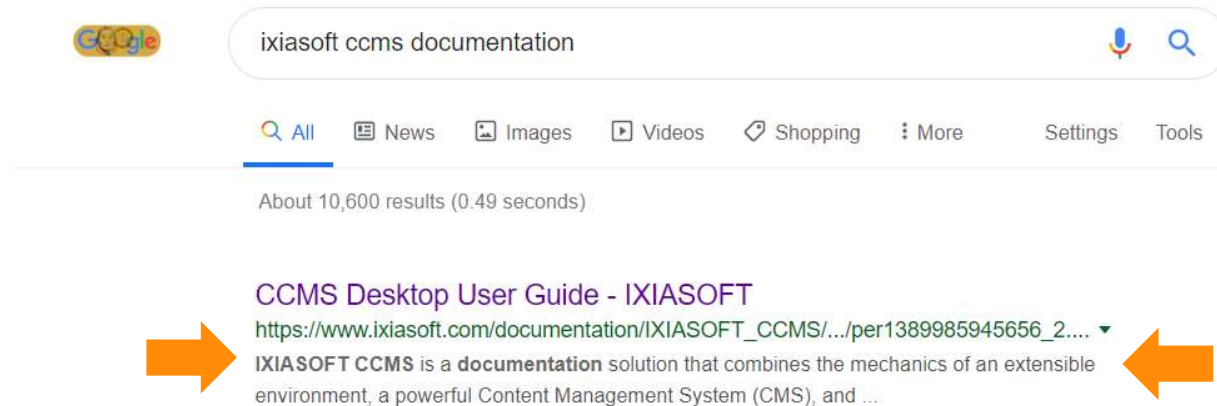


Short Descriptions and SEO

Short Descriptions help with SEO, as they are displayed immediately after the webpage title in a Google Search:

```
<title>Introduction</title>
<shortdesc><keyword keyref="CCMS_long"/> is a documentation solution that
  combines the mechanics of an extensible environment, a powerful Content Management System
  (CMS), and the elegant modularity of DITA.</shortdesc>
```

```
<title>Introduction</title>
<p class="shortdesc">
<span class="keyword">IXIASOFT CCMS</span> is a documentation solution that
  combines the mechanics of an extensible environment, a powerful Content
  Management System (CMS), and the elegant modularity of DITA.</p>
```



The screenshot shows a Google search interface. The search bar contains the text "ixiasoft ccms documentation". Below the search bar, there are navigation options: "All", "News", "Images", "Videos", "Shopping", "More", "Settings", and "Tools". The search results show "About 10,600 results (0.49 seconds)". The first result is titled "CCMS Desktop User Guide - IXIASOFT" with a URL starting with "https://www.ixiasoft.com/documentation/IXIASOFT_CCMS/...". Below the title and URL, a short description is displayed: "IXIASOFT CCMS is a **documentation** solution that combines the mechanics of an extensible environment, a powerful Content Management System (CMS), and ...". Two orange arrows point towards the short description from the left and right sides.

DITA



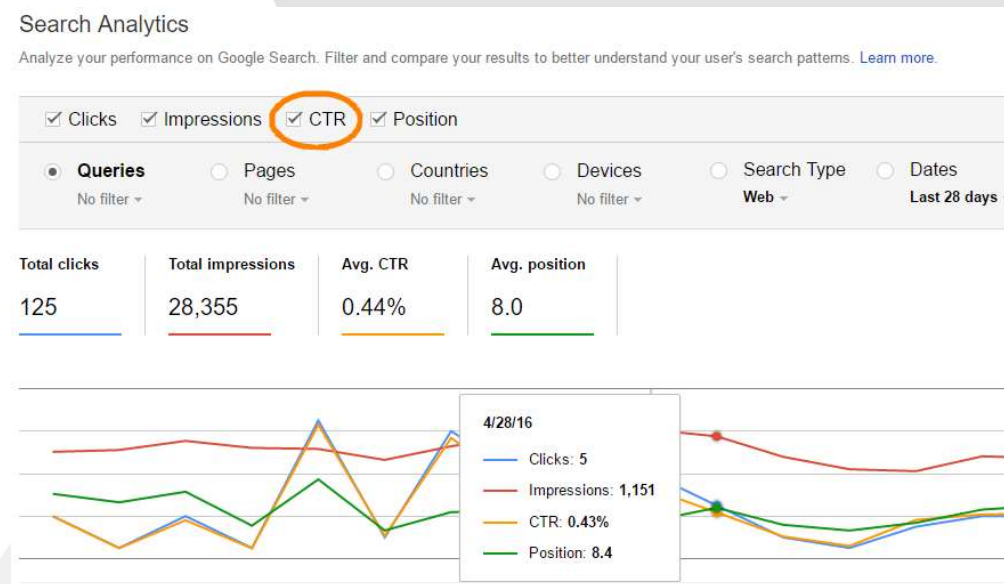
XHTML



Google
Search

Short Descriptions and Click-throughs

- While short descriptions are not factored in search engine rankings, user behaviours are
- Google measures click-through rates (CTR)
- A well-written, descriptive short description ensures more click-throughs



Writing Effective Short Descriptions for SEO

- A well-written short description tells the would-be reader why it is worth clicking on
 - Task: tell users what they can accomplish
 - Concept: tell users about what you are describing and why they should care
 - Reference: tell users what the referenced item does or what it can be used for
 - Troubleshooting: describe the symptoms of a problem a user may encounter and let them know that this topic can help
- While shortdesc best practices suggests two sentences, Google truncates search results at ~156 characters
- They also help when searching for reusable content within a CCMS

#4: A DITA Test Output Document? What's That?

- A DITA output related tale: a DITA test document is a map + topics designed specifically to test output conditions
- As new XSL transforms are added, include new content to the test document to see how they work
- Can be used to test fonts, image sizing, widow/orphan control, how errors appear; basically to test the “look and feel” of everything

78 Sample ElementFormattingDescriptions

Caution:
Remember to discharge your body's static electricity by touching the power supply case or the metal surface of the computer chassis.

4. Remove any unneeded graphics cards from the computer.

Warning:
Some graphics cards can get hot while operating. Wait approximately five minutes after turning off the computer before touching the card.

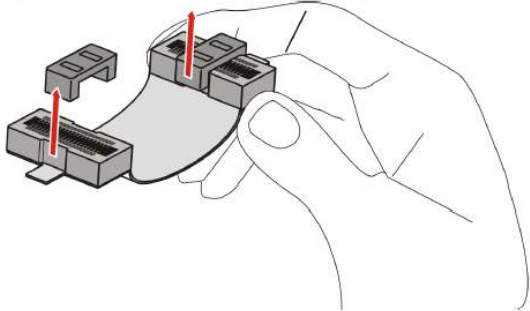
5. Locate the PCIe slots on the motherboard. If necessary, remove the corresponding back panel cover.

6. Align the new graphics card in the secondary PCIe slot next to the existing graphics card and press firmly until the card is fully seated.

7. Remove protection tags from the ends of the AMD CrossFire interconnect and connect the cards.

Caution:
Always handle the AMD CrossFire bridge interconnects by the connector only. Do not push or pull on the flexible cable.

Figure 3-32 AMD CrossFire™ Bridge Interconnect Protection Tag



EDITING PRELIMINARY

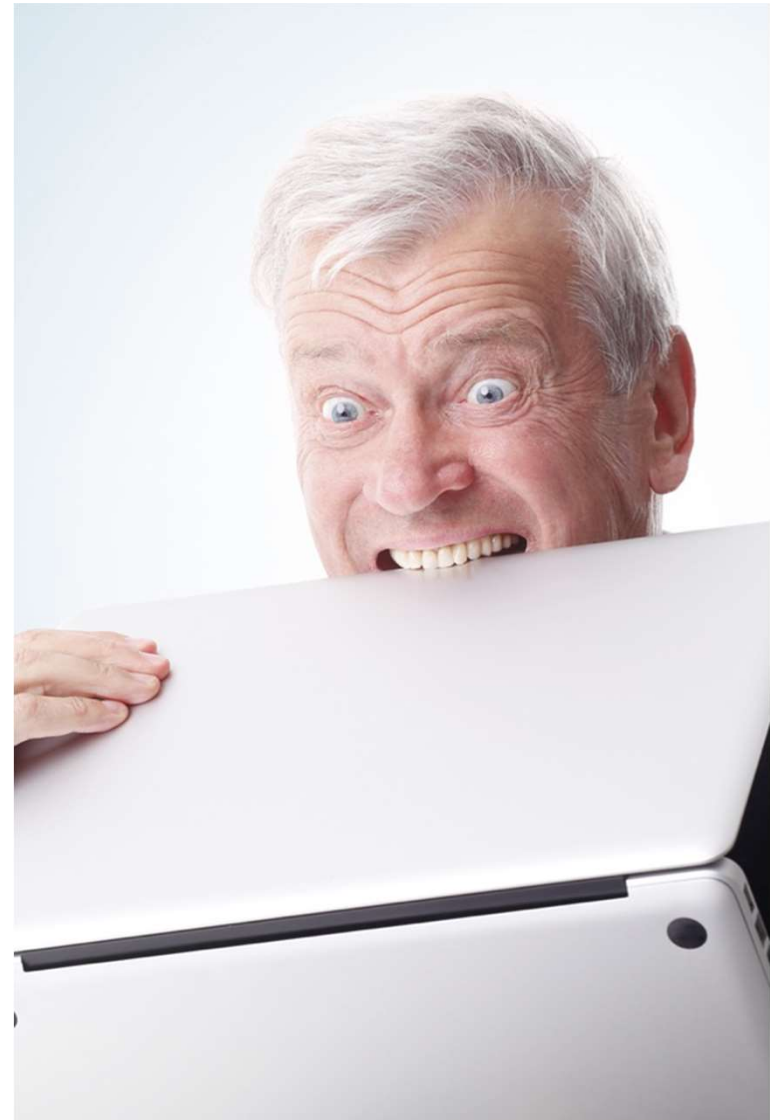
sau1147098367112.ditamap 2011/09/28 10:00 EDT

Template Test—Comprehensive © 2006 Advanced Micro Devices Inc.

“It Was Working Before”

Why having a test document can be useful:

- Example: shortly after a software upgrade, images were no longer rendered properly for a new document; blame was put on software
- Separate test installation showed that new software was working fine
- A recent change in how images were inserted into DITA code was the real culprit; in the end a single line of XSL code had to be tweaked
- Discovering the root cause without having a test document *took 8 hours...*



Solution: Create Your Own Output Test Document

- It should test all aspects of your DITA output conditions
- It can serve as a companion to your style guide, showing expected output under all circumstances
- Every time something is changed (new DITA feature, software, rendering engine, etc.), check output using test document and compare to previous version
- It should not be an existing document, as it can always be changed (as the last example shows)

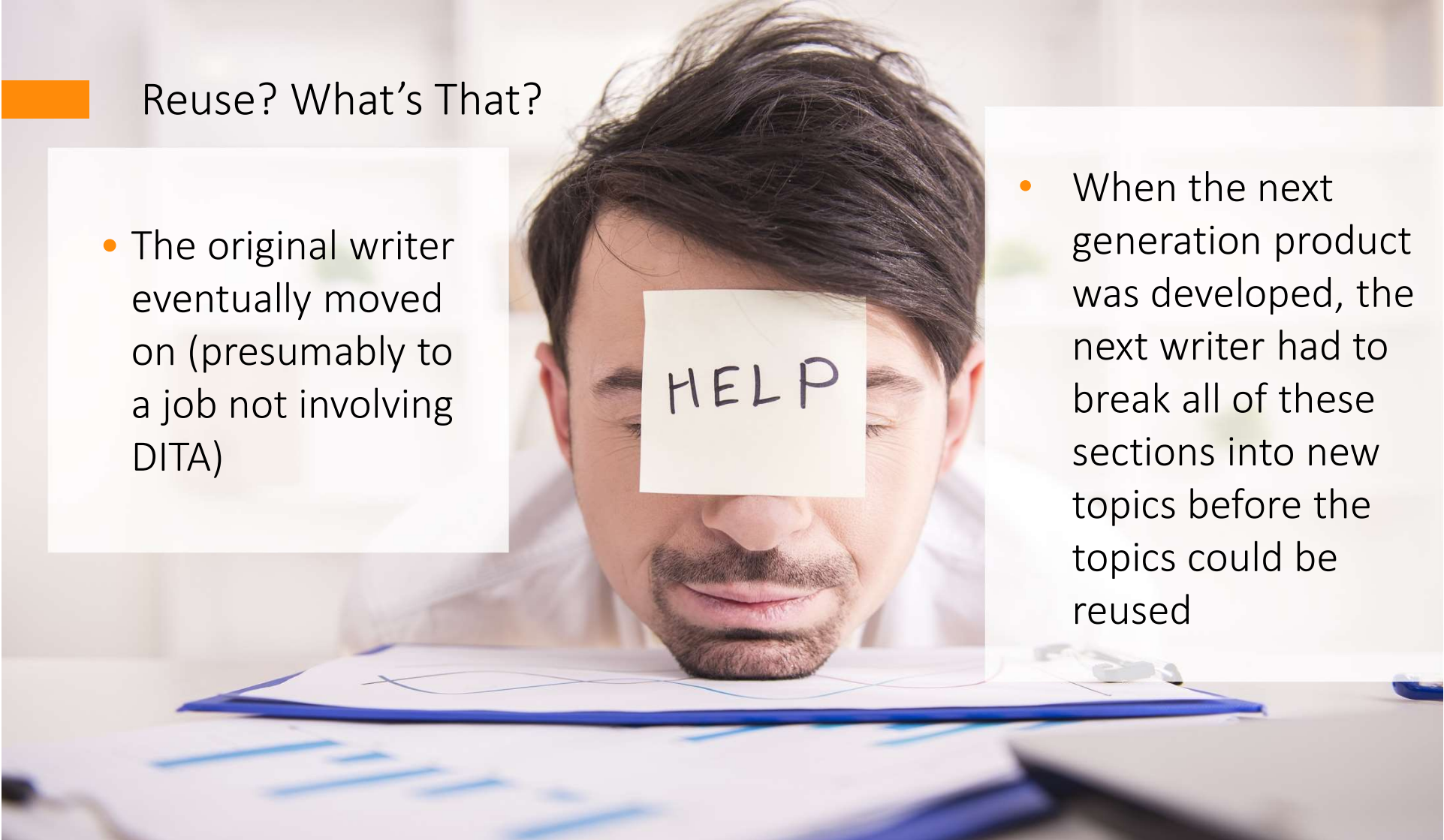
2.4 Glass-Eating Test Table

Language	"I can eat glass, it doesn't hurt me."
Sanskrit	
Sanskrit (standard transcription)	kācaḥ śaknomyattum; nopahinasti mām.
Classical Greek	ἕλαον φαγεῖν δύνωμαι· τοῦτο οὐ με βλάπτει.
Greek (monotonic)	Μπορώ να φάω σπασμένα γυαλιά χωρίς να πάθω τίποτα.
Etruscan	(NEEDED)
Latin	Vitrum edere possum; mihi non nocet.
Old French	Je puis mangier del voirre. Ne me nuit.
French	Je peux manger du verre, ça ne me fait pas de mal.
Provençal / Occitan	Pòdi manjar de veire, me nafrariá pas.
Québécois	J'peux manger d'la vitre, ça m'fa pas mal.
Walloon	Dji pou magnî do vêre, çoula m' freut nén mâ.
Champanois	(NEEDED)
Lorrain	(NEEDED)
Picard	Ch'peux mingi du verre, cha m'foé mie n'ma.
Corsican	(NEEDED)
Jèrriais	(NEEDED)
Kreyòl Ayisyen	Mwen kap manje vè, li pa blese'm.
Basque	Kristala jan dezaket, ez dit minik ematen.
Catalan / Català	Puc menjar vidre, que no em fa mal.
Spanish	Puedo comer vidrio, no me hace daño.
Aragones	Puedo minchar beire, no me'n fa mal.
Galician	Eu podo xantar cristais e non cortarme.
Portuguese	Posso comer vidro, não me faz mal.
Brazilian Portuguese (7)	Posso comer vidro, não me machuca.
Caboverdiano	M' podé cumé vidru, ca ta maguá-m'.
Papiamentu	Ami por kome glas anto e no ta hasimi dañu.
Italian	Posso mangiare il vetro e non mi fa male.
Milanese	Són bòn de magnà el véder, el me fa minga mal.
Roman	Me posso magna' er vetro, e nun me fa male.
Napoletano	M' pozz magna' o'vetr, e nun m' fa mal.

#5: A Document within a Topic

- At a firm where they recently introduced DITA, there was a technical writer on the team who had a hard time grasping topic-based writing
- Over the objections of their information architect, their manager (who wasn't that familiar with DITA either) let this writer use one topic with dozens of sections, basically dropping a whole chapter or even *an entire doc* within a single (generic) topic



A man with dark hair and a beard is looking down at a document on a desk. A yellow sticky note with the word 'HELP' written on it is stuck to his forehead. The background is a blurred office setting.

Reuse? What's That?

- The original writer eventually moved on (presumably to a job not involving DITA)

- When the next generation product was developed, the next writer had to break all of these sections into new topics before the topics could be reused

A Valuable Lesson Was Learned by the Team

- This caused a lot more work for the second writer. And the same problem was encountered by other writers who ended up having to update the original writer's content.
- Simple solution: don't squeeze a doc into a single topic
- But more broadly this underscores the need for effective training and ongoing support for your technical writers in the fundamentals of DITA
 - (That and listen to your Information Architects!)

#6: If a DITA Tag Exists, We Should Use It

- Sharon Figueira's story: "during the first migration I did, my team and I were so enchanted with DITA and its every last detail that we implemented as many tags as possible. We had the approach of: 'if it's there we should use it'.

Basic + Technical DITA 1.3 Elements

abbreviated-form	change-started	dlhead	groupchoice	msgblock	propdeschd	source	textentity
abbrevlist	change-summary	draft-comment	groupcomp	msgnum	properties	spare	tgroup
abstract	chapter	draftintro	groupseq	msgph	property	spares	thead
addressdetails	chdesc	dt	hasInstance	namedetails	prothead	sparesli	throughfare
administrativearea	chdeschd	dthd	hasKind	navref	proptype	started	title
alt	chhead	dvrKeyscopePrefix	hasNarrower	navtitle	proptypehd	state	titlealts
amendments	choice	dvrKeyscopeSuffix	hasPart	noconds	propvalue	stentry	tm
anchor	choices	dvrResourcePrefix	hasRelated	nosafety	propvaluehd	step	toc
anchorid	choicetable	dvrResourceSuffix	hazardstatement	nospare	pt	stepresult	topic
anchorkey	choption	edited	hazardsymbol	nosupeq	published	steps	topicapply
anchorref	choptionhd	edition	honorific	nosupply	publisher	stepsection	topicCell
apiname	chrow	elementdef	howtoavoid	note	publisherinformation	steps-informal	topicGroup
appendices	cite	emailaddress	i	notices	publishtype	steps-unordered	topichead
appendix	closereqs	emailaddresses	image	no-topic-nesting	q	steptroubleshooting	topicmeta
approved	cmd	entry	imagemap	numcharref	refbody	stepxmp	topicref
area	cmdname	enumerationdef	index-base	object	refbodydiv	shead	topicset
attributedef	codeblock	equation-block	indexlist	ol	reference	strow	topicsetref
audience	codeph	equation-figure	index-see	oper	refsyn	sub	topicsubject
author	coderef	equation-inline	index-see-also	option	related-links	subjectCell	topicSubjectHeader
authorinformation	colophon	equation-number	index-sort-as	organization	relatedSubjects	subjectcdf	topicSubjectRow
b	colspec	equation-term	index-term	organizationinfo	relcell	subjectHead	topicSubjectTable
backmatter	completed	example	indextermref	organizationname	relcolspec	subjectHeadMeta	trademarklist
bibliolist	component	exportanchors	info	organizationnamedetails	relheader	subjectref	tbody
body	conbody	featnum	isbn	otherinfo	relrow	subjectRel	troubleshooting
bodydiv	conbodydiv	fig	itemgroup	othermeta	reltable	subjectRelHeader	troublebody
bookabstract	concept	figgroup	keydef	overview	remedy	subjectRelTable	troubleSolution
bookchangehistory	condition	figurelist	keyword	p	resep	subjectRole	tt
bookevent	consequence	filepath	keywords	param	reqcond	subjectScheme	typeofhazard
bookeventtype	contactnumber	firstname	kwd	parameterentity	reqconds	substep	u
bookid	contactnumbers	fn	lastname	parml	reqcontp	substeps	uicontrol
booklibrary	context	foreign	li	parmname	reqpers	summary	ul
booklist	coords	fragment	lines	part	required-cleanup	sup	unknown
booklists	copyrfirst	fragref	line-through	pd	resourceid	supeqi	url
bookmap	copyrholder	frontmatter	link	permissions	responsibleParty	supequi	uris
bookmeta	copyright	generationidentifier	linkinfo	perscat	result	supequip	userinput
booknumber	copyrlast	glossAbbreviation	linklist	perskill	reviewed	supplies	ux-window
bookowner	copyryear	glossAcronym	linkpool	person	revised	supply	var
bookpartno	country	glossAlt	linktext	personinfo	revisionid	supplyli	varname
bookrestriction	created	glossAlternateFor	locality	personname	safecond	svg-container	volume
bookrights	critdates	glossarylist	localityname	personnel	safety	svgref	vm
booktitle	data	glossBody	longdescref	ph	schemeref	symlink	vmlist
booktitlealt	data-about	glossDef	longquoteref	platform	screen	synbik	wintitle
boolean	day	glossentry	lq	plentry	searchtitle	synnoteref	xmiatt
brand	dd	glossgroup	mainbooktitle	postalcode	section	synph	xmlelement
category	ddhd	glossPartOfSpeech	maintainer	postreq	sectiondiv	syntaxdiagram	xmlesname
cause	dedication	glossProperty	map	pre	sep	systemoutput	xmpli
change-completed	defaultSubject	glossref	mapref	preface	series	table	xref
change-historylist	delim	glossScopeNote	markupname	prelreqs	shape	tablelist	year
change-item	desc	glossShortForm	mathml	prereq	shortcut	task	
change-organization	dita	glossStatus	mathmlref	printlocation	shortdesc	taskbody	
change-person	ditaValmeta	glossSurfaceForm	menucascade	prodinfo	tbody	tasktroubleshooting	
change-request-id	ditaValref	glossSymbol	messagepanel	prodname	simpletable	tbody	
change-request-reference	div	glossSynonym	metadata	prognum	sl	term	
change-request-system	di	glossTerm	middleware	prolog	sli	tested	
change-revisionid	dientry	glossUsage	month	propdesc	sort-as	text	

433 Elements

The Problem, The Pain

- It was a self-made problem: “The ridiculous thing was that we weren’t using these tags before DITA, so why we thought we suddenly had a need for @fastpath etc. I have no idea.”
- The pain: “the writers soon revolted and refused to put in multiple inline tags per paragraph.”



You Are Not Obligated to Use All DITA Tags

- While you might want to add new semantically-descriptive tags/elements to your existing content when migrating, the suggestion is to “build out complexity slowly and in response to a well-understood need.”
- Lightweight DITA might also be something to think about, if not for your writers then perhaps for your SMEs:

<alt>	<desc>	<map>	<section>	<title>	Multimedia	<media-
<body>	<fig>	<note>	<shortdesc>	<topic>	<audio>	muted>
	<fn>		<simpletable>	<topicmeta>	<media-	<media-
<data>	<image>	<p>	<stentry>	<topicref>	autoplay>	source>
<dd>	<i>	<navtitle>	<sthead>	<u>	<media-	<media-
<dentry>	<keydef>	<ph>	<strow>		controls>	track>
<dt>	<linktext>	<pre>	<sub>	<xref>	<media-loop>	<video>
<dl>		<prolog>	<sup>			<video-
						poster>

- 39 “regular” topic elements, 48 when including multimedia elements

A server rack with flames and sparks, symbolizing a problem or failure.

Problems with Tools and Processes
(aka “how *not* to work effectively with people”)

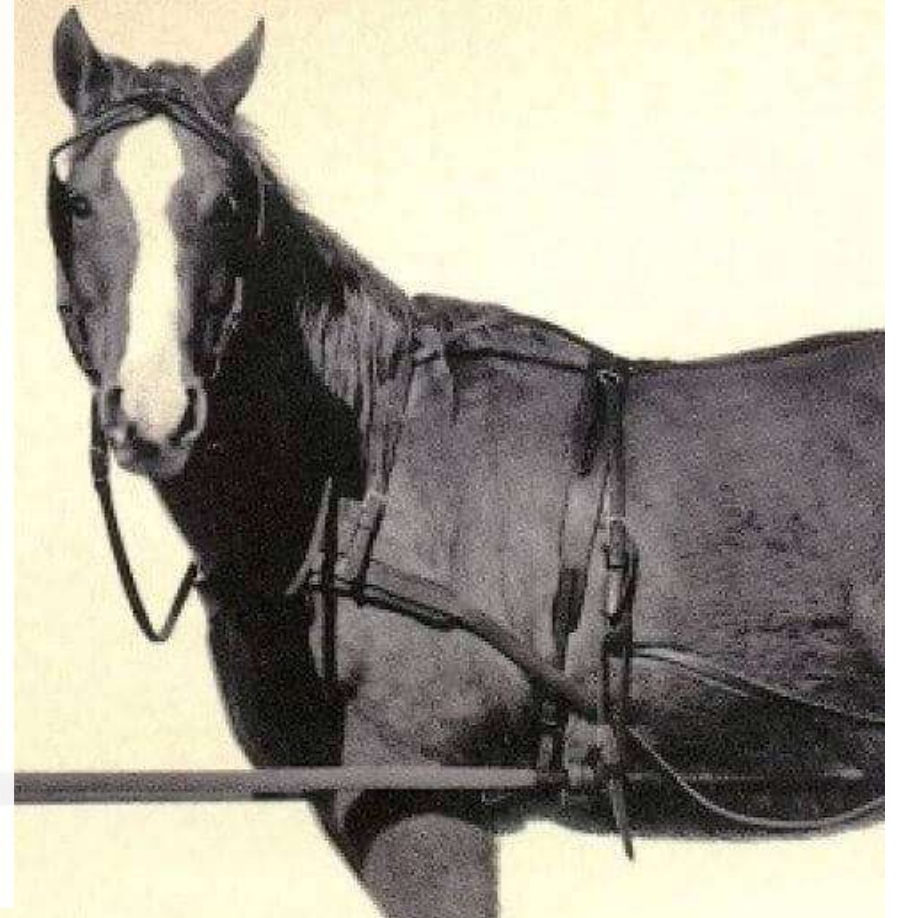
#6: Don't Choose Your CCMS First

- This is more a piece of advice from someone who has worked in the field as a consultant for a long time: Eliot Kimber
- From a post last December to the Yahoo! DITA Users Group, he talked about CCMSs and the portability of DITA content and he got to talking about tools, including CCMSs



His Thoughts on When to Buy a CCMS

“...putting a CCMS in place should be the *last* thing you do in your DITA implementation, not the first. A CCMS is a huge investment in time and money and it’s a decision that is very hard to undo once you do it. You need to proceed carefully and you can’t do that until you have enough experience with DITA in general and your requirements in particular to be able to make an informed decision”.



What to Look for When Buying a CCMS?

Again, from Eliot:

“Here are some easy tests of a potential system:

1. Give them a set of maps and topics and related resources with at least... local specializations and ask them what’s involved in getting it into their system.
2. Give them some custom OT plugins and ask what it takes to make them available.
3. Tell them you really require branched versioning and ask them how that works in their system.
4. Mention that your documents use scoped keys. What features do they have to support the storage and authoring of those documents.
5. Do [an] import and export test.”

Don't Get the Tools Until You Know They Will Do the Job

A final word on this from Eliot:

- “You cannot know whether or not a CCMS system really works until you go through your second or third revision cycle. Only then will you see where it succeeds or fails in terms of version management, revision control, support for version-aware linking, etc.”
- The original post:
<https://groups.yahoo.com/neo/groups/dita-users/conversations/messages/44354>



#7: “Just Let IT Choose a CCMS for You”

- A couple of years ago I worked on an internal project, studying the factors behind “successful” RFIs/RFQs from the vendor perspective
- Over a third of the RFIs/RFQs I reviewed had zero or minimal references to “DITA” in them
- In most of these cases it was clear that IT was choosing a CCMS based on their technical requirements, with little to no input from the technical writers



Some Not Unexpected Results When this Happens

- Technical writing team is given a CCMS that they had no part in selecting; not surprisingly, this often leads to general unhappiness
- Project may lead to outright failure; DITA / CCMS initiative may be dropped
- Or, at significant expense, a second, more appropriate CCMS is selected, this time with input from technical writing staff



Technical Writers Need to Part of Any RFI Created

- When your company is choosing a CCMS, make sure that the technical writing staff plays an active part in the selection
- Should cover not only what you need in terms of DITA requirements, but in terms of processes and expected workflows, report capabilities, metrics gathering, expected content contributions from SMEs, localization requirements, etc.!



Additional Info: Top Five CCMS Feature Requests

- Based on the information from 84 different RFIs I reviewed, these were the top five features that were requested:

FEATURES

1. Workflow
2. Need for Reviews
3. Taxonomy Support
4. Localization Support
5. Reports / SME Contribution (tie)



#8: A Unified DITA Content Strategy Ought to Be... Unified

- The following happened at a company that was undergoing a merger:
 - The “home team” was well underway with a DITA project that they had undertaken
 - Thinking they were helping, they converted a huge amount of legacy content from the TechDocs team from the newly-acquired firm

What They Didn't Do

- The DITA content basically just sat there, in a file folder
- The other team was then accused of “dragging their feet” when it came to working with DITA
- What the “home team” didn't do was provide much in the way of training, or open up lines of communication with the other team



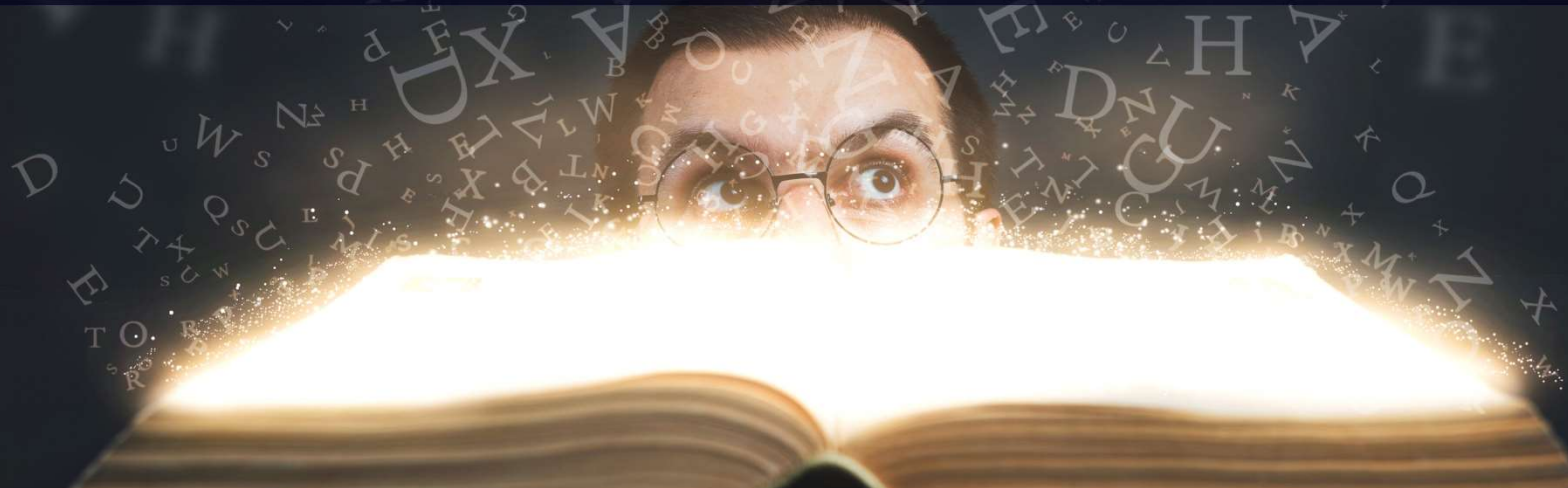


Need to Stream Content *and* Teams

- Provide training.
- Build a bridge from the tools and processes that people knew to those that they didn't know.
- Create written processes to clarify roles, especially for things that everyone used to understand as second nature.
- Foster a sense of unity.
- Communicate early and often.

Documentation Does Not Happen by Magic

- Must remember that all documentation is made by and for people
- DITA may be a driver towards creating better documentation, but it is part of a larger process that involves people and tools
- Need to think about documentation in a new way

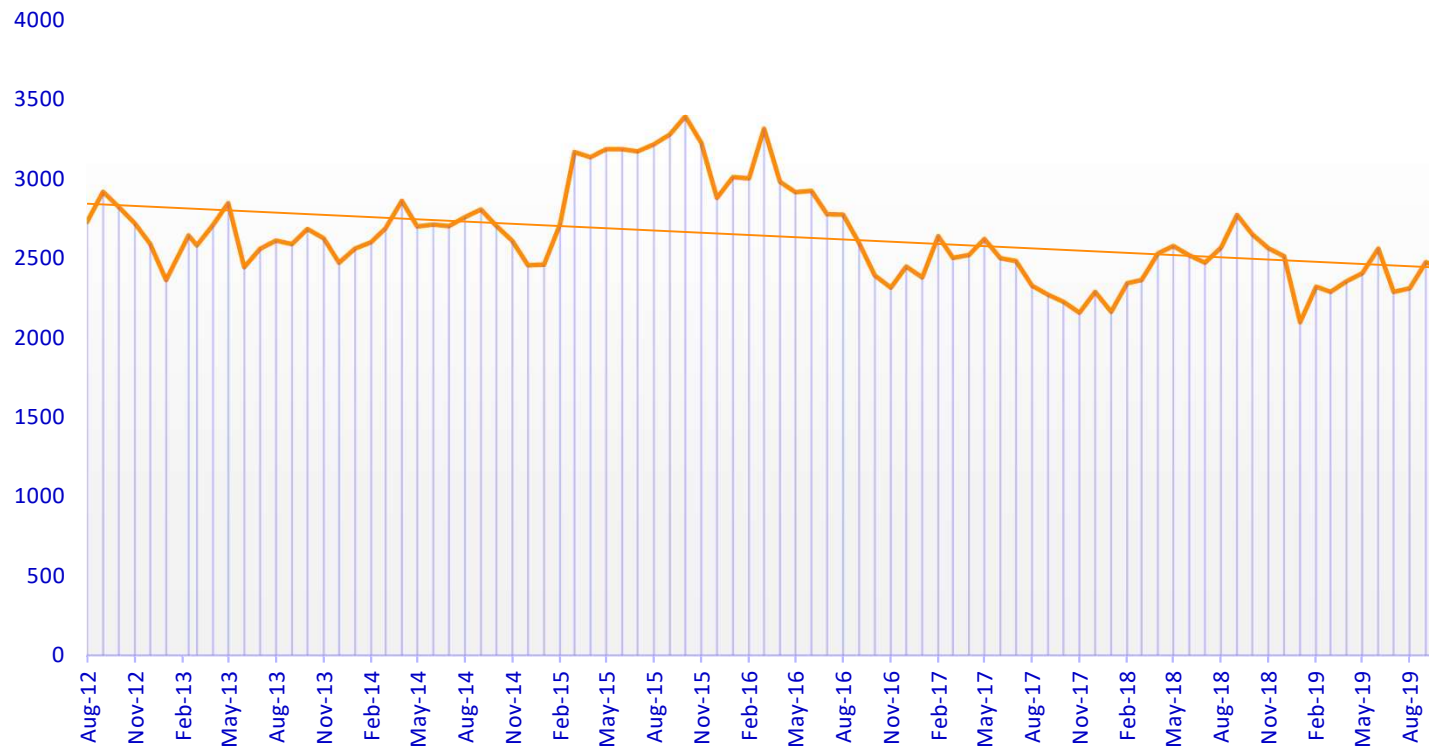


A man in a dark suit and patterned tie sits at a desk, smiling as he looks at a laptop. A woman with blonde hair, wearing a leopard-print top and dark jeans, leans over the desk from the right, also smiling and looking at the laptop. The desk is cluttered with papers, a green binder, and a red folder. In the background, there are windows with blinds and a whiteboard with some handwritten notes.

Who Is Actually “Practicing ” DITA?

■ The “Technical Writer” Job Landscape is Changing...

"Technical Writer" Job Listings on Indeed.com for Aug 2012 - Oct 2019



One Reason: SMEs are Producing More Upstream Content



- When it comes to API documentation, programmers are expected to provide much of the content. This is often framed and put into context by technical writers.
- The advent of Agile documentation processes in small software development teams means that, in some circumstances, SMEs had to write content.

The Other Reason: The Role of the “Technical Writer” is Changing

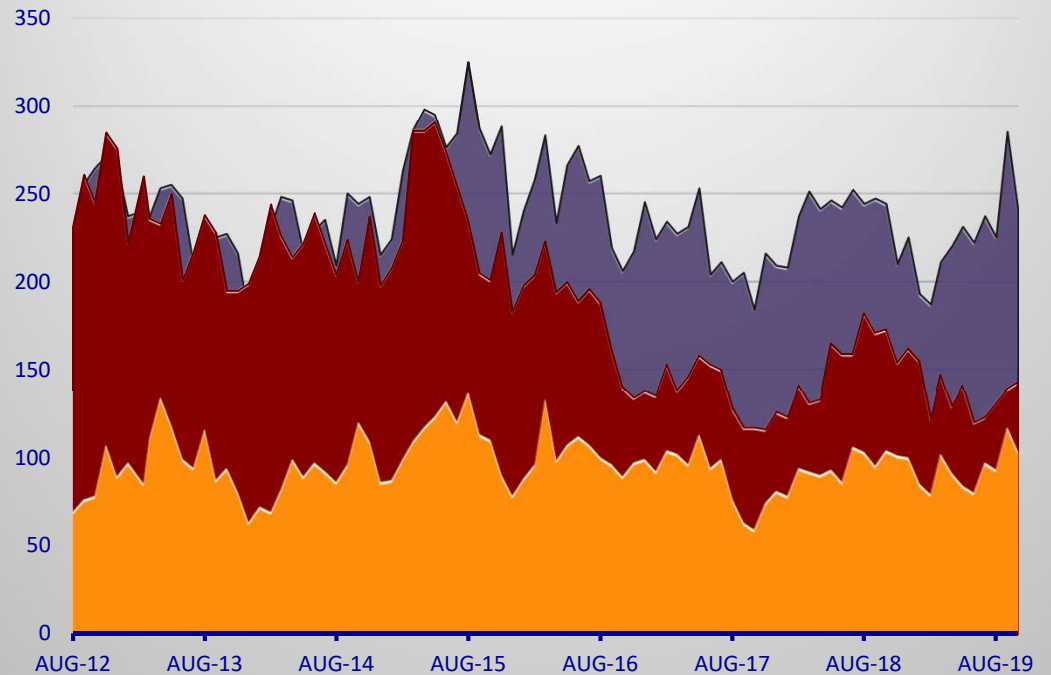
In a survey I did of 1,500 LinkedIn profiles where people claimed to be using DITA, **66%** were not employed as traditional technical writers; some selected job titles:

- Applications Engineer
- Chief Information Architect, UX Analyst
- Consulting Content Strategist
- Content Architect
- Content Developer
- Content Management Specialist
- Content Strategist
- DITA Architect
- DITA Content Strategist
- DITA Information Architect
- DITA Migration Specialist
- Information Architect
- Information Developer
- Information Experience Manager
- Knowledge Architect
- Lead Information Developer
- Localization Program Manager
- Manager, XML CMS and L10n Systems
- Principal Content Experience Developer
- Principal Information Developer
- Product Architect
- Project Manager and Documentation Engineer
- Senior Content Developer
- Senior Content Strategist
- Senior Documentation Tools Developer
- Staff Information Architect
- Team Leader Technical Documentation
- User Assistance Development Architect
- UX Designer
- XML/DITA Coordinator

■ We are in the middle of a significant industry change

“Technical writer” jobs are being replaced by positions that use structured content to add more value by focusing on the content experience for users.

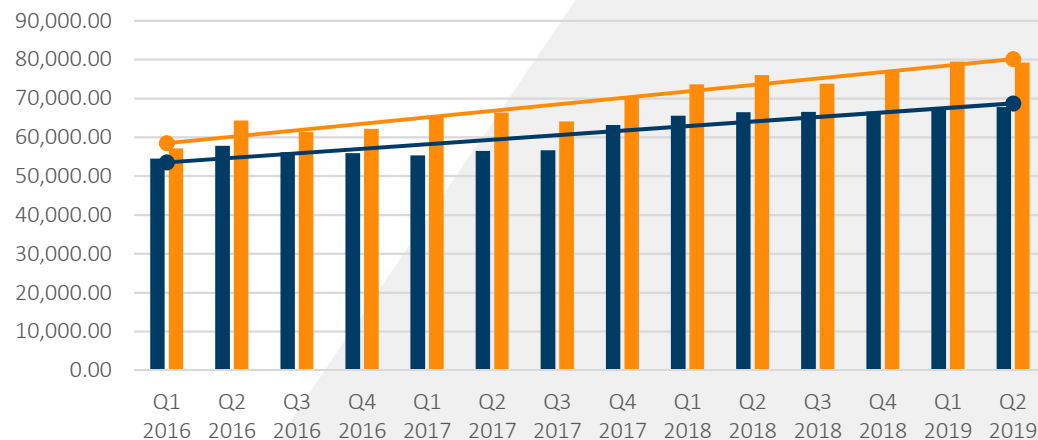
FrameMaker and XML (+ DITA) in Technical Writer Job Postings on Indeed.com from Q2 2012 to Oct 2019



A Final Thought

- “Technical Writing”, at least when it comes to DITA-based content, is no longer about writing finely-crafted prose, but about structuring content in a way that best serves the needs of users, whenever and wherever they need it
- Also, having “DITA” in your resume is definitely a good thing, as the chart below shows (starting wages in \$US for Tech Writers vs. Tech Writers + DITA)

Avg. Starting Wage for Technical Writers on Indeed.com
from Q1 2016-Q2 2019



Questions?

Keith Schengili-Roberts
Senior DITA Content Strategist

robertsk@ixiasoft.com