

TECH COMM TRENDS

PROVIDING VALUE AS A GENERALIST IN A SEA OF SPECIALISTS

By Tom Johnson / @tomjohnson
idratherbewriting.com

Slides: idratherbewriting.com/trends-generalists-specialists

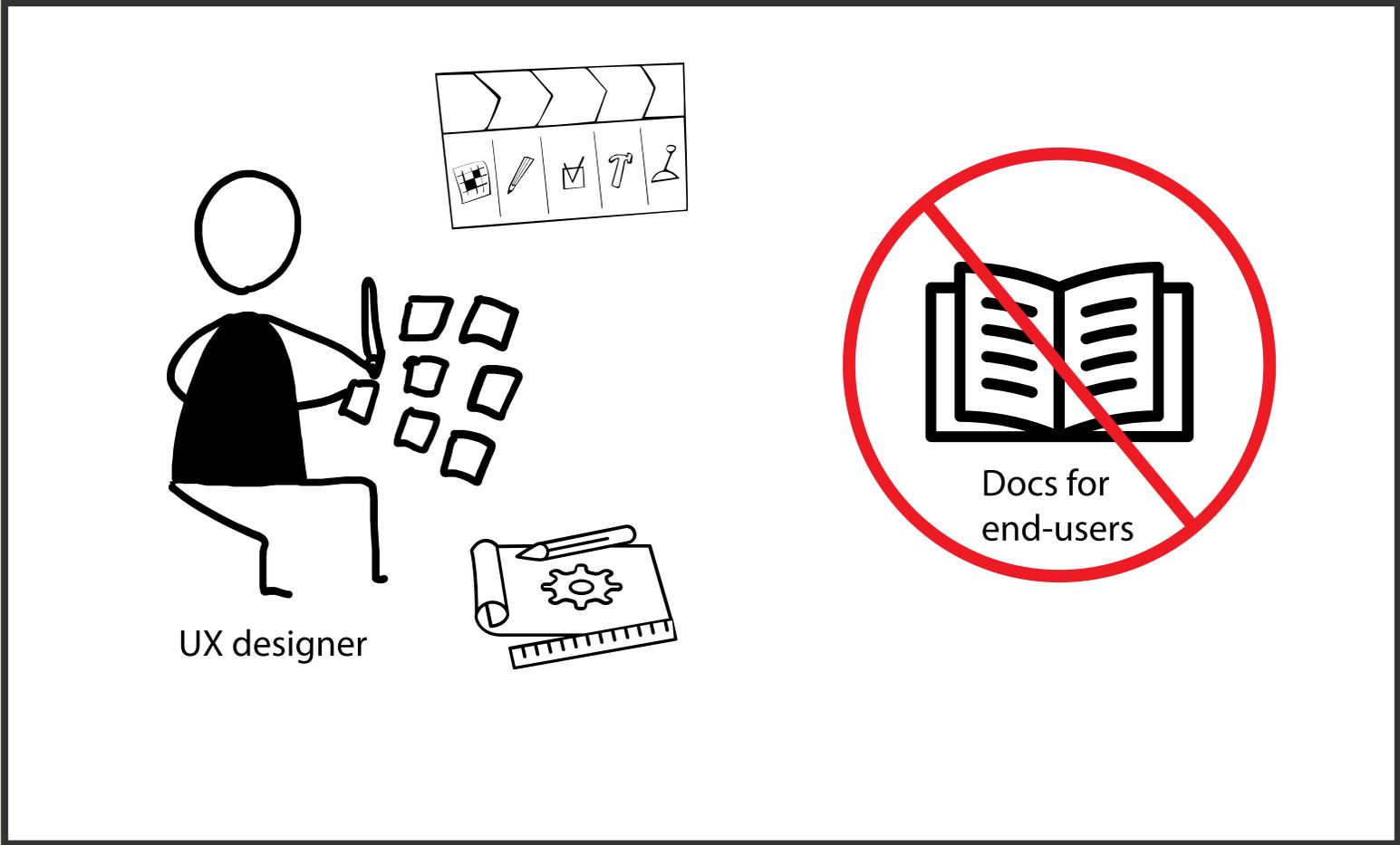
Presentation in blog form: <http://bit.ly/genandspecialisttrendspart1>

ARGUMENT OVERVIEW

- Technology is getting simpler on the front-end for end-users
- But the code underneath is becoming increasingly specialized/complex
- Tech writers are generalists, not specialists
- To provide value in specialist contexts, tech writers must exploit the gaps
- These gaps are (1) doc tools/processes, (2) understanding user feedback/experiences, and (3) information usability

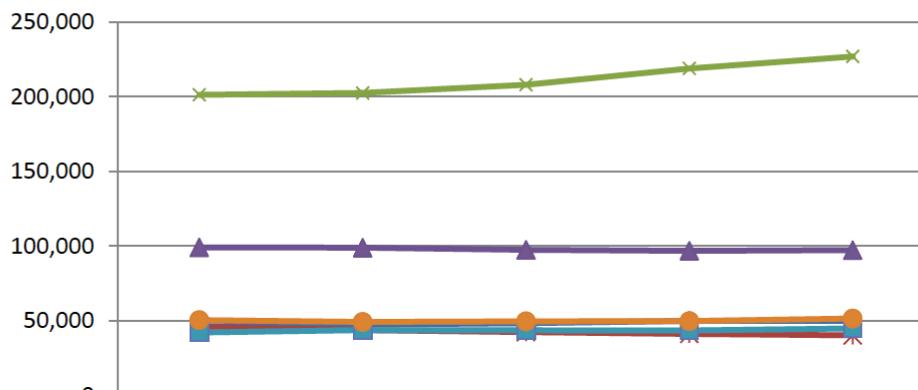
TRENDS: FROM END-USER TO DEV DOMAIN

THE IMPACT OF UX AND THE NEED FOR DOCUMENTATION



EVALUATING GREY'S ARGUMENT AND CHANGES IN THE PROFESSION

Employment Levels of Technical Writers and Other Media and Communication Occupations

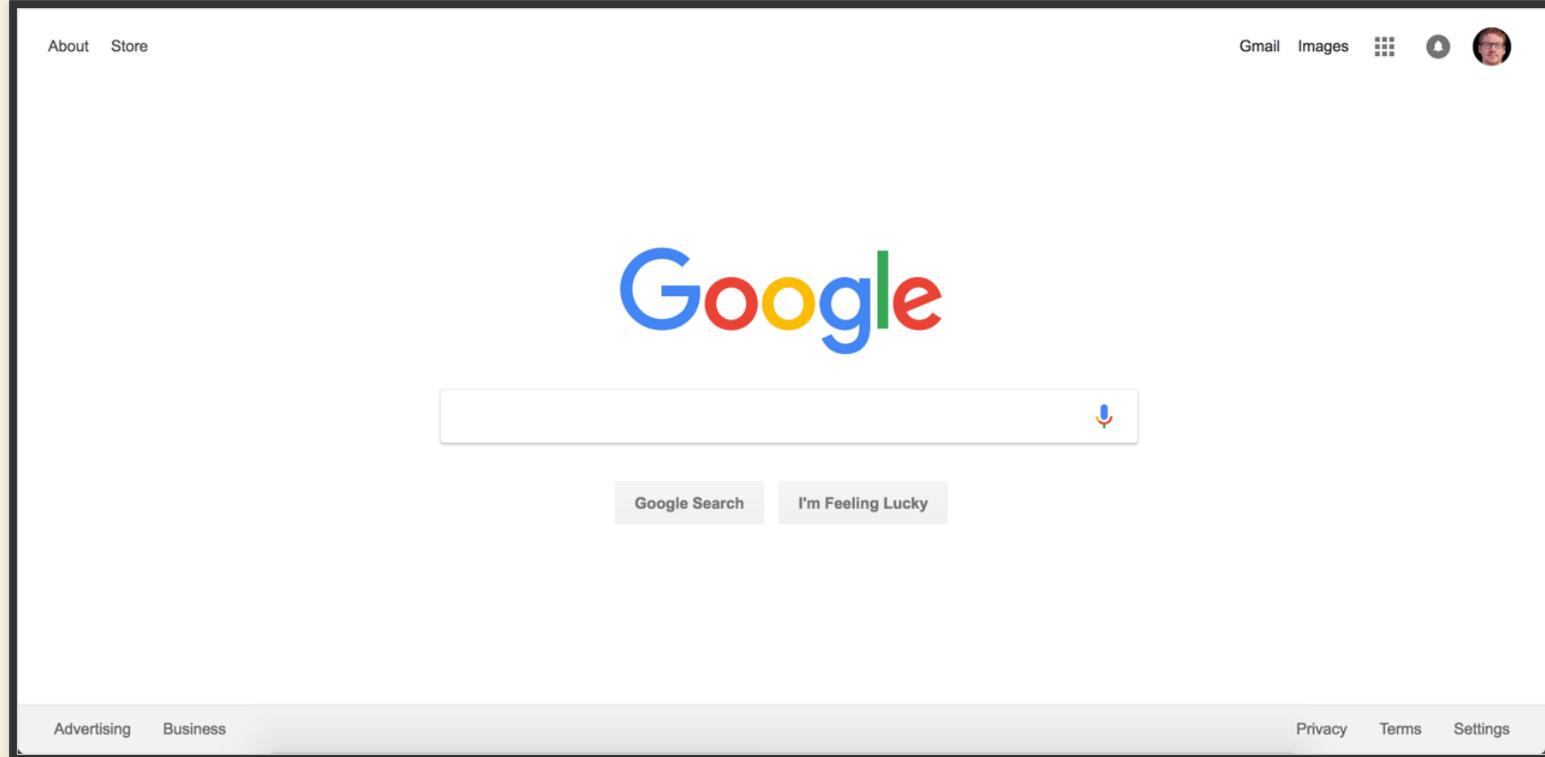


Technical Writers	46,160	47,300	48,210	49,770	49,780
Reporters and Correspondents	45,570	43,630	42,280	41,050	40,090
Public Relations Specialists	201,280	202,530	208,030	218,910	226,940
Editors	99,040	98,790	97,350	96,690	97,170
Writers and Authors	41,990	43,590	43,500	43,380	44,690
Interpreters and Translators	50,320	49,060	49,460	49,650	51,350

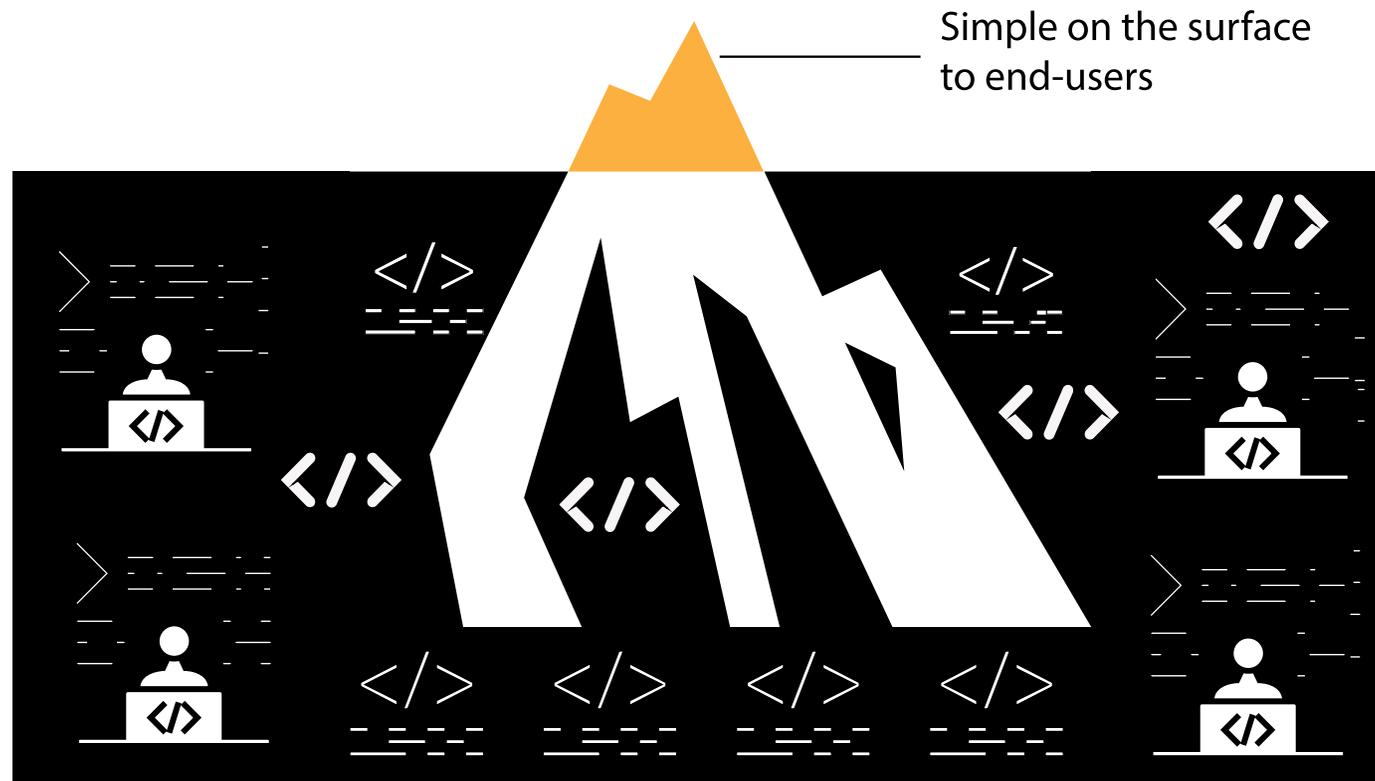


STC's 2016-2017 Salary Database
Sponsored by Adobe

UIs GET SIMPLER, CODE GETS MORE COMPLEX



NATURAL LANGUAGE INTERFACES -- LIKE AN ICEBERG UNDERNEATH



Simple on the surface
to end-users

Complex underneath with lots of development work and code

MOVING INTO HYPERSPECIALIZATION

Just as people in the early days of industrialization saw single jobs (such as a pin maker's) transformed into many jobs (Adam Smith observed 18 separate steps in a pin factory), we will now see knowledge-worker jobs — salesperson, secretary, engineer — atomize into complex networks of people all over the world performing highly specialized tasks.

— "The Big Idea: The Age of Hyperspecialization"

NO UX DESIGNERS TO VET/FILTER POOR DESIGNS IN DEV DOMAIN

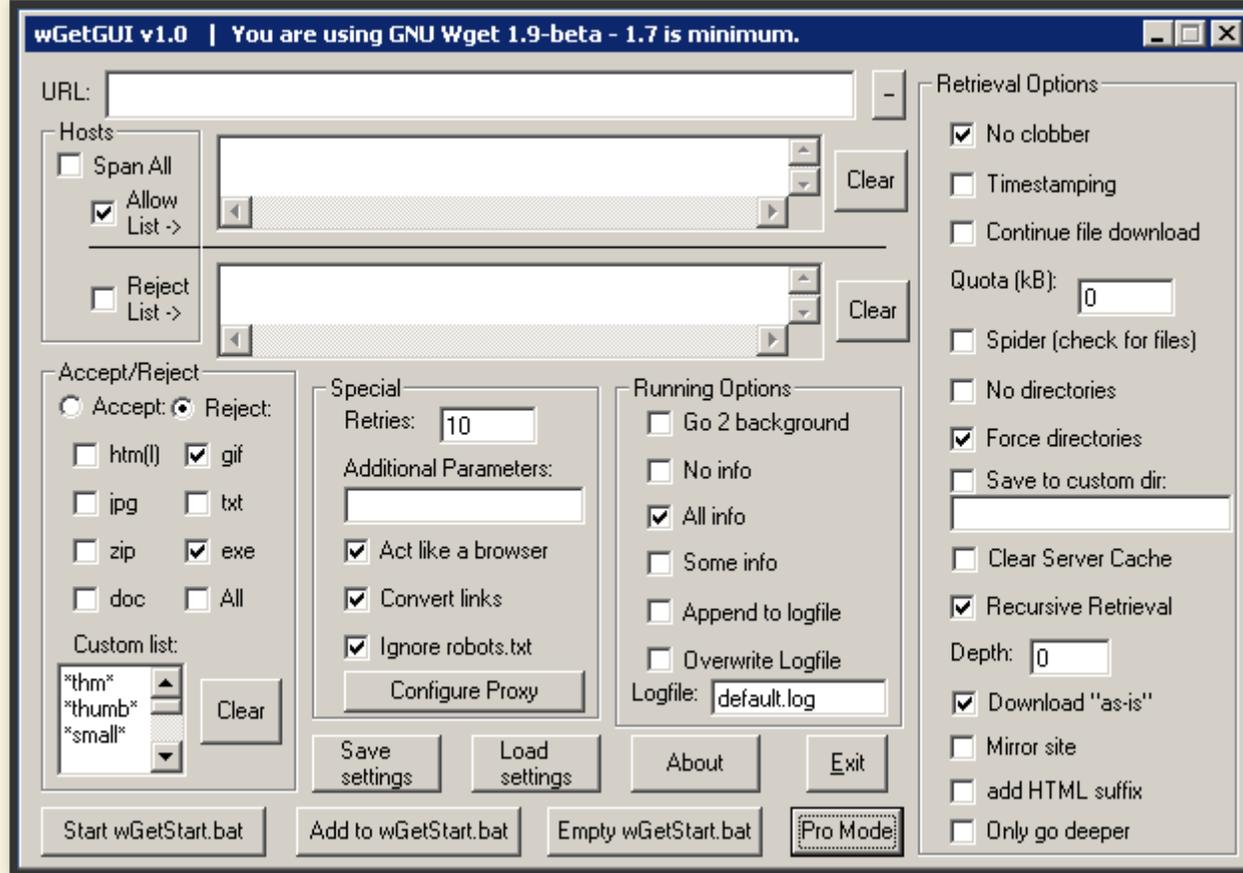
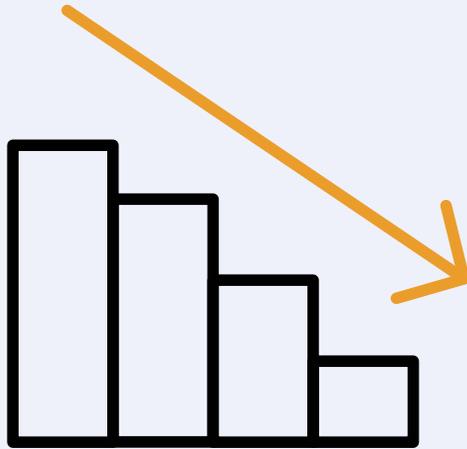


Image from Coding Horror

TC JOBS MOVING INTO DEVELOPER DOMAIN

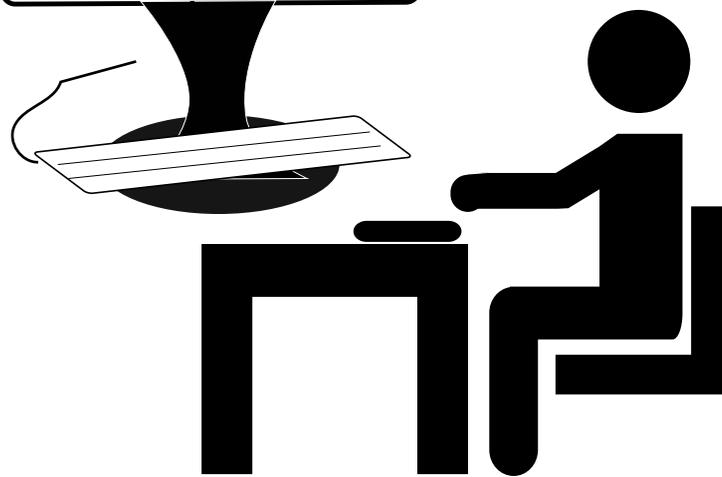
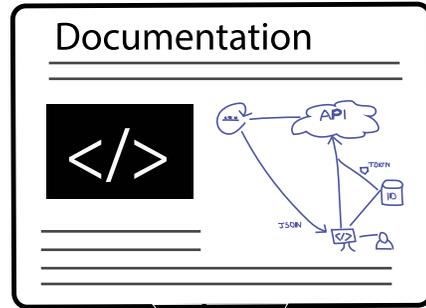


Tech Writing Jobs in
End-User Domains



Tech Writing Jobs in
Developer Domains

DEVELOPERS ARE WRITING MORE DOCS DUE TO SPECIALIZED INFO



Developer

WRITE THE DOCS AS EVIDENCE THAT DEVELOPER DOCS IS GROWING



Write the Docs is a global community of people who care about documentation. We have a Slack community, conferences on 3 continents, and local meetups!

Useful Pages

Events and Activities

- Conferences
- Meetups

Learning Resources

- Newsletter & Mailing lists
- Conference Videos

Welcome to our community!

Write the Docs is a global community of people who care about documentation. Our primary gathering places are:

- Our slack network with thousands of members
- Conferences on 3 continents
- Local meetups in over 30 cities

We consider everyone who cares about communication, documentation, and their users to be a member of our community. This can be programmers, tech writers, developer advocates, customer support, marketers, and anyone else who wants people to have great experiences with software.

Our conferences create a time and a place for the global community of Documentarians to share information, discuss ideas, and work together to improve the **art and science of documentation**.

Join the community

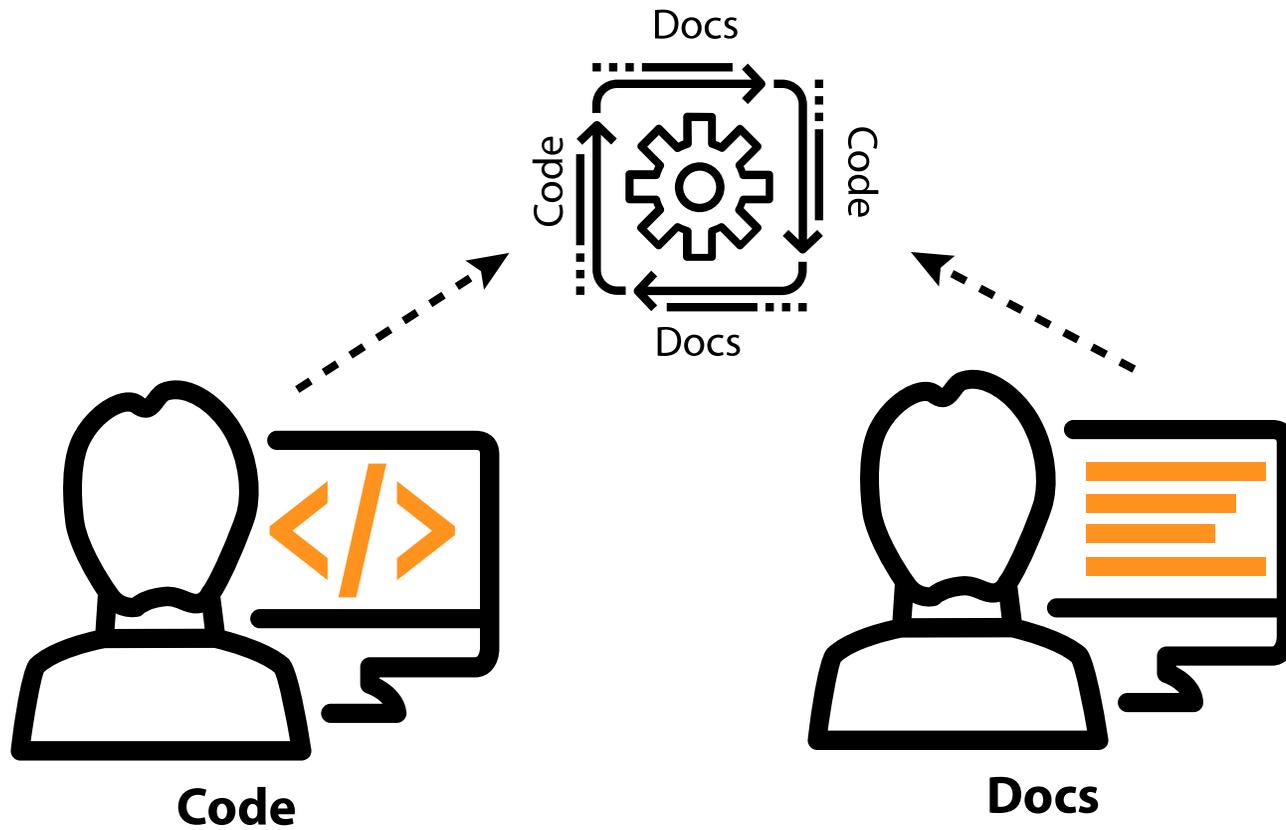
Get more information on how to meet the community, learn new things, get involved, and stand in touch. We have a few sets of resources for you to start with:

- Events and Activities
 - Conferences

Fork me on GitHub

v: latest

SHIFTS TOWARD MARKDOWN AND DOCS-AS-CODE



THE CURRENT PREDICAMENT FOR TC, AND WHERE THE GAPS ARE

Predicament

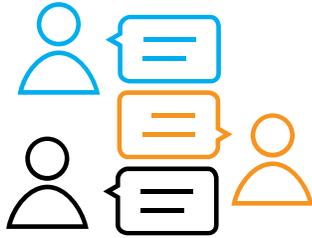
Generalists trying to prove value in a context where specialized knowledge reigns

Gaps

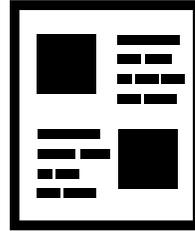
1. Authoring/publishing processes and tools
2. Knowledge/feedback about the user experience
3. Information usability

1. GAPS IN DOC TOOLING/PROCESSES

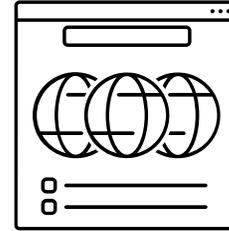
OPPORTUNITIES RELATED TO DOC TOOLS AND PROCESSES



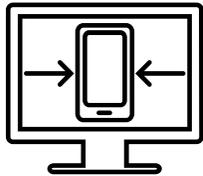
Review process



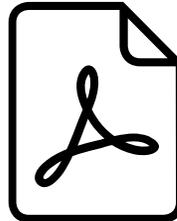
Layout & design



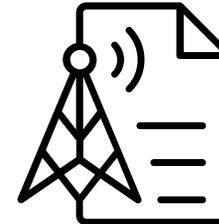
web publishing



Responsive design



PDF output

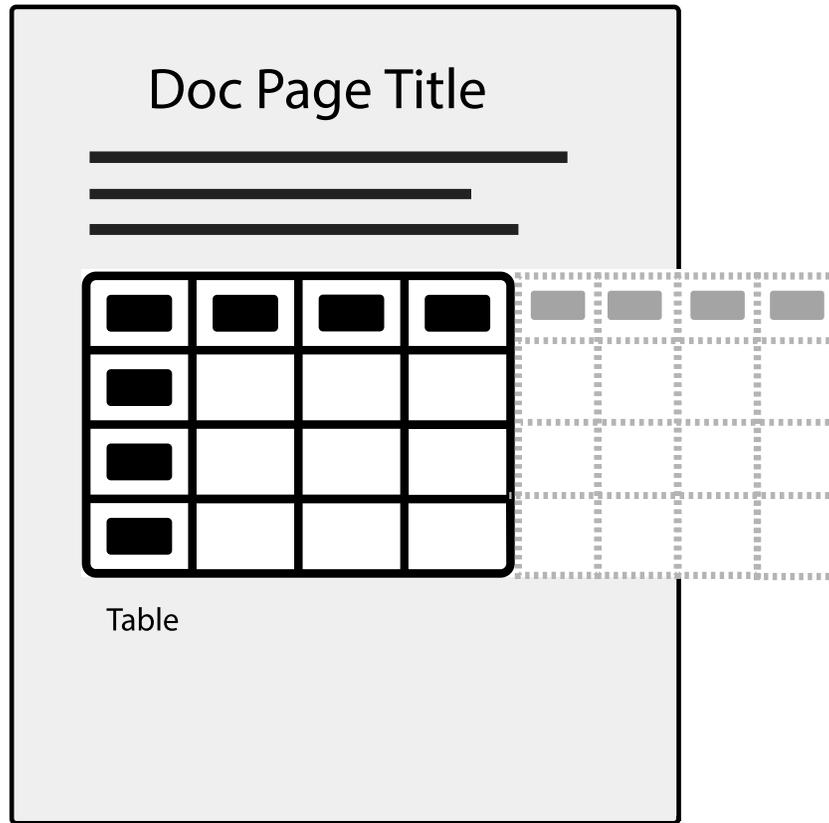


Syndication

INCORPORATING STRUCTURE INTO CONTENT

The aim [of structured writing] is not to eliminate complexity altogether – that is impossible – but to partition it so that each part of that complexity is handled by the person or process with the knowledge, skills, and resources to handle it. (xxi)
– Mark Baker, *Structured Writing*

TABLES DON'T WORK BEYOND 5-6 COLUMNS



SEPARATE THE CONTENT FROM THE DISPLAY

```
media_specifications:
  video:
    h265:
      ftvcube: Hardware accelerated up to 3840x2160p (4K) @ 60fps...
      ftvgen3: Hardware accelerated up to 3840x2160p (4K) @ 60fps...
      ftvgen2: Hardware accelerated up to 3840x2160p (4K) @ 30fps...
      ftvgen1: Not supported
      ftvstickgen2: Hardware accelerated up to 1080p @ 30fps...
      ftvstickbasicedition: Hardware accelerated up to 1080p @ 30fps...
      ftveditionelement: Hardware accelerated up to 3840x2160p (4K) @ 60fps...
      ftveditiontoshiba4k: Hardware accelerated up to 3840x2160p (4K) @ 60fps.
    h264:
      ftvcube: Hardware accelerated up to 3840x2160p @ 30fps...
      ftvgen3: Hardware accelerated up to 3840x2160p @ 30fps...
      ftvgen2: Hardware accelerated up to 1080p @ 30fps...
      ftvgen1: Hardware accelerated up to 1080p @ 30fps...
```

DISPLAY RESULT

amazon appstore Developer Console Sign In ?

Getting Started Devices Build Test Submit SDK Downloads Docs

Amazon Fire TV

[Collapse All](#) | [Expand All](#)

- Overview
 - Get Started Developing Apps and Games
 - Device Specifications for Fire TV**
 - Development Framework Comparison: Fire App Builder, WASK, and Amazon Creator
 - Fire TV Development Versus Android TV Development
 - TV App Design UX Guidelines
 - Fire OS Overview
 - What's New in Development
 - Fire TV FAQs
 - Developing for Amazon Fire TV Devices Running Fire OS 6
 - Submitting Your Fire TV App

[Submit Feedback](#)

Select the Fire TV device you want to see.

Fire TV Edition - Insignia HD (2018) ▾

- Fire TV Edition - Insignia HD (2018)**
- Fire TV Edition - Insignia 4K (2018)
- Fire TV Edition - Toshiba HD (2018)
- Fire TV Cube (2018)
- Fire TV Edition - Toshiba 4K (2018)
- Fire TV – Gen 3 (2017)
- Fire TV Edition – Element 4K (2017)

quality 2K streaming at a low cost. Fire TV Edition - Toshiba HD (2018). As Insignia HD ships with Fire OS 6 (based on Android TV) and 1GB DDR4 memory. Like other 2018 Fire TV Editions — you can use voice search through near-field voice (using your remote control) or through far-field voice. When setting up the Insignia HD, you aren't required to have any streaming apps. Several screen sizes are available: 24" (720p), 32" (720p), 43" (1080p), and 55" (4K).

IDENTIFY THE STRUCTURE

```
ElementName:  
  anchor: string  
  description: >  
    string  
  required: boolean  
  added: string  
  deprecated: string  
  parent_elements:  
    - name: string  
      deprecated: boolean  
  child_elements:  
    required:  
      - name: string  
        deprecated: boolean  
    optional:  
      - name: string
```

DISPLAY RESULT

MiniSeriesEpisode

Optional

One of the basic work types, a [MiniSeriesEpisode](#) is a single episode in a [MiniSeries](#). This content is not associated with a season and is sequenced in the context of the [MiniSeries](#).

Property	Detail
Use	Optional
Added	CDF version 1.3
Parent Elements	Catalog
Child Elements	Required: ID , Offers , Title Optional: AdultProduct , Color , ContentRatings , JP_Require18PlusAgeConfirmation , Copyright , Credits , CustomerRating , ExternalID , Genres , ImageUrl , Language , Rank , ReleaseInfo , ShortDescription , ReleaseYear , RuntimeMinutes , Source , Studios , Synopsis Note: Either MiniSeriesID or MiniSeriesTitle ,
Child Elements Specific to This Element Only	Required: EpisodeInSeries , MiniSeriesID , MiniSeriesTitle Optional: OriginalAirDate
Attributes	None

Example:

```
<MiniSeries>
  <ID>MS-123456789</ID>
  ...
</MiniSeries>
<MiniSeriesEpisode>
  <ID>MSE-2220880</ID>
```

USING SPECIFICATIONS WITH APIS

The image shows a screenshot of the Swagger IDE interface. On the left, a code editor displays a Swagger 2.0 specification in JSON format. On the right, the rendered UI for the 'Swagger Petstore (Simple)' API is shown, including a description, version, contact information, terms of service, license, and a list of paths.

```
1 swagger: '2.0'
2 info:
3   version: '1.0.0'
4   title: Swagger Petstore (Simple)
5   description: A sample API that uses a petstore
6     as an example to demonstrate features in the
7     swagger-2.0 specification
8   termsOfService: http://helloverb.com/terms/
9   contact:
10    name: Swagger API team
11    email: foo@example.com
12    url: http://swagger.io
13  license:
14    name: MIT
15    url: http://opensource.org/licenses/MIT
16  host: petstore.swagger.io
17  basePath: /api
18  schemes:
19    - http
20  consumes:
21    - application/json
22  produces:
23    - application/json
24  paths:
25    /pets:
26      get:
27        description: Returns all pets from the
28          system that the user has access to
29        operationId: findPets
30        produces:
31          - application/json
32          - application/xml
```

Swagger Petstore (Simple)

A sample API that uses a petstore as an example to demonstrate features in the swagger-2.0 specification

Version 1.0.0

Contact information
Swagger API team

foo@example.com

<http://swagger.io>

Terms of service
<http://helloverb.com/terms/>

License
MIT

Paths

[/pets](#)

GET /pets
Description

SWAGGER UI DISPLAY FROM THE OPENAPI SPECIFICATION

The screenshot shows the Swagger UI interface for the Petstore API. At the top, there is a green header with the Swagger logo, a search bar containing the URL `https://petstore.swagger.io/v2/swagger.json`, and an **Explore** button. Below the header, the main content area displays the API title **Swagger Petstore** with a version badge **1.0.0**. Underneath, it provides the base URL `petstore.swagger.io/v2` and the Swagger JSON file location `https://petstore.swagger.io/v2/swagger.json`. A paragraph of introductory text explains that this is a sample server and provides links to Swagger documentation and a special key for testing. There are also links for [Terms of service](#), [Contact the developer](#), [Apache 2.0](#), and [Find out more about Swagger](#).

Below the introductory text, there is a **Schemes** dropdown menu set to **HTTPS** and an **Authorize** button with a lock icon. The main API section is titled **pet** with the description "Everything about your Pets" and a link to <http://swagger.io>. The API endpoints are listed in a table-like format:

Method	Endpoint	Description	Lock Icon
POST	<code>/pet</code>	Add a new pet to the store	🔒
PUT	<code>/pet</code>	Update an existing pet	🔒
GET	<code>/pet/findByStatus</code>	Finds Pets by status	🔒
GET	<code>/pet/findByTags</code>	Finds Pets by tags	🔒
GET	<code>/pet/{petId}</code>	Find pet by ID	🔒

2. GAPS IN USER FEEDBACK/EXPERIENCE

IDENTIFYING WHERE THE UX PROBLEMS ARE

TABLE 2

API documentation problems reported in the exploratory survey.

Category	Problem	Description	E*	D*
Content	Incompleteness	The description of an API element or topic wasn't where it was expected to be.	20	20
	Ambiguity	The description of an API element was mostly complete but unclear.	16	15
	Unexplained examples	A code example was insufficiently explained.	10	8
	Obsolescence	The documentation on a topic referred to a previous version of the API.	6	6
	Inconsistency	The documentation of elements meant to be combined didn't agree.	5	4
	Incorrectness	Some information was incorrect.	4	4
	Total			61
Presentation	Bloat	The description of an API element or topic was verbose or excessively extensive.	12	11
	Fragmentation	The information related to an element or topic was fragmented or scattered over too many pages or sections.	5	5
	Excess structural information	The description of an element contained redundant information about the element's syntax or structure, which could be easily obtained through modern IDEs.	4	3
	Tangled information	The description of an API element or topic was tangled with information the respondent didn't need.	4	3
	Total			25

* E is the number of examples that mentioned a problem; D is the number of developers who reported a problem.

SURFACE PROBLEMS TO THE RIGHT TEAMS

So, how can we improve API documentation if the only people who can accomplish this task are too busy to do it...? One potential way [is to] reduce as much of the administrative overhead of documentation writing as possible, letting experts focus exclusively on the value-producing part of the task. ... a main challenge for evolving API documentation is identifying where a document needs to be updated. – Uddin and Robillard

DOC FEEDBACK BUTTONS

The screenshot shows the Amazon Appstore developer documentation page for Amazon Fire TV. The page has a dark blue header with the Amazon Appstore logo and navigation links: Getting Started, Devices, Build, Test, Submit, SDK Downloads, and Docs. The main content area is white and features a sidebar on the left with a tree view of topics. The main content area displays the title 'Getting Started Developing Games' and a 'Submit Feedback' button with a speech bubble icon. An arrow points to this button. Below the button is a paragraph of text and a 'Table of Contents' section with a link to 'Options for Building Fire TV Apps'.

amazon appstore

Getting Started Devices Build Test Submit SDK Downloads Docs

Amazon Fire TV

[Collapse All](#) | [Expand All](#)

- Overview
 - Get Started Developing Apps and Games**
 - Device Specifications for Fire TV
 - Development Framework Comparison: Fire App Builder, WASK, and Amazon Creator
 - Fire TV Development Versus Android TV Development
 - TV App Design UX Guidelines
 - Fire OS Overview
 - What's New in Development

Getting Started Developing Games



When developing apps for Fire TV, you can choose from a variety of your skillset (Java developer, web developer, or content developer) HTML5 web app), the features you want (advertising, authentication) will help you get started in building an app.

Table of Contents

- [Options for Building Fire TV Apps](#)

DOC FEEDBACK FORM

Documentation feedback

This feedback applies to the following page:

<https://developer.amazon.com/docs/fire-tv/getting-started-developing-apps-and-games.html>

How would you rate the documentation on the page referenced above?

	Agree	Disagree
Content is accurate	<input type="radio"/>	<input type="radio"/>
Content is complete	<input type="radio"/>	<input type="radio"/>
Content is easy to follow	<input type="radio"/>	<input type="radio"/>

How can we improve the documentation?

If you need help, see the [Appstore Developer Forums](#).

SURVEYS AT SELECT MILESTONE EVENTS

Please tell us about your recent experience in creating and publishing an app. Was anything frustrating or noteworthy?

Did you refer to Amazon's documentation while building and submitting your app?

Very Frequently

Frequently

Occasionally

Rarely

Never

SUMMARIES OF WEEKLY ISSUES RESOLVED

Fire TV

Update Fire TV device specs for new device launch

<https://issues.amazon.com/issues/DEX-FIRE-TV-88>

Gather up specification information for new device and publish along with the other specifications for Fire TV. (Points: 1)

Pages updated:

<https://developer.amazon.com/docs/fire-tv/device-specifications.html?v=ftveditioninsigniahd>

Catalog

Separate out catalog ingestion from device implementation in docs (first draft)

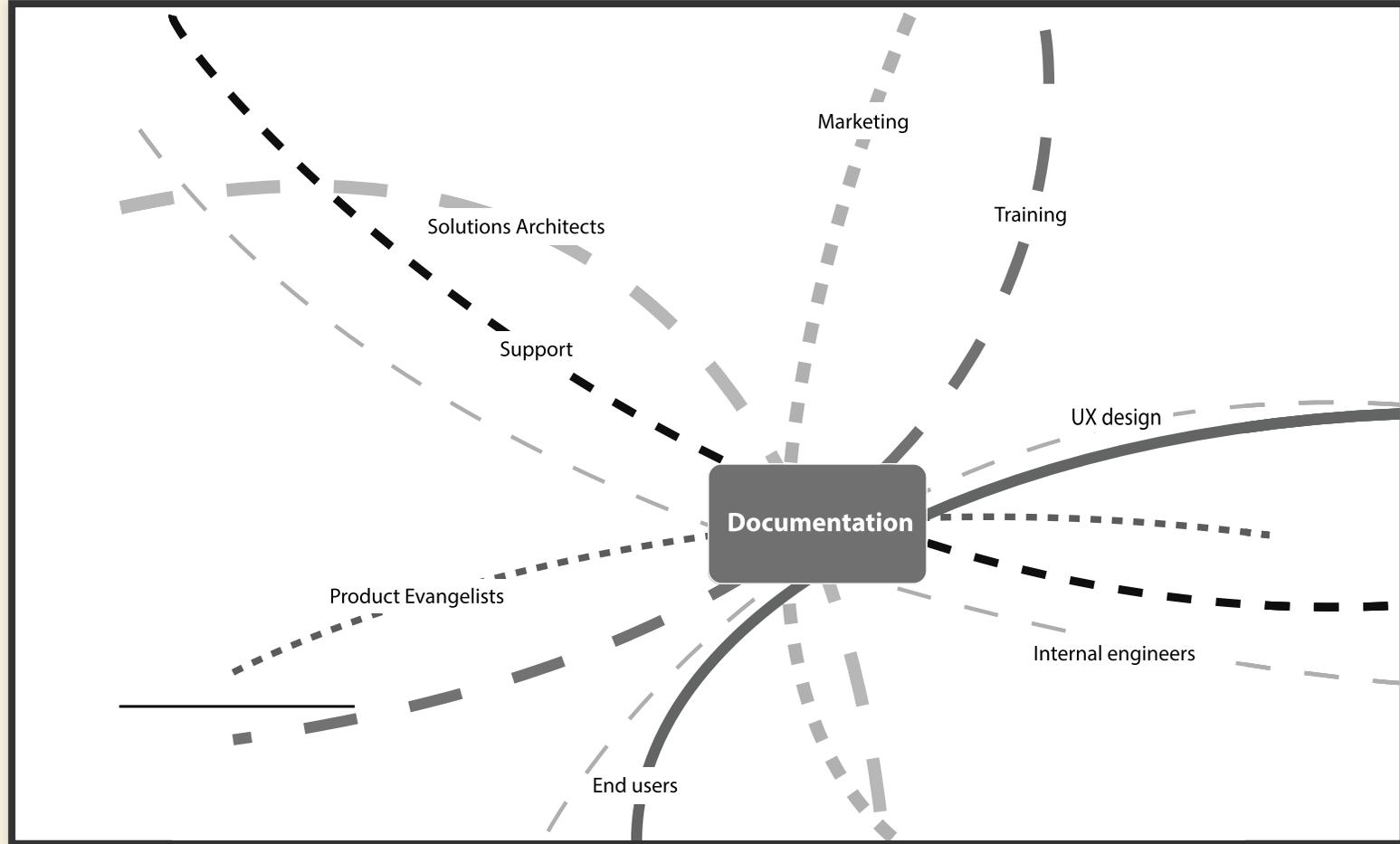
<https://issues.amazon.com/issues/DEX-CATALOG-7>

The previous catalog docs were deeply intertwined with the Fire TV implementation information, since at that time, catalog ingestion only made sense in the context of Fire TV. Now that additional devices beyond Fire TV can interact with the catalog, we needed to make catalog ingestion independent of a specific device endpoint.

Here are some notes about what we changed:

- Divided Catalog docs into two main sections: “Catalog Ingestion” and “Universal Search and Browse on Fire TV”
- Called the whole process of implementing catalog on Fire TV, integrating with launcher, etc., as implementing “Universal

LEVERAGING INFORMATION FLOW THROUGH DOCS

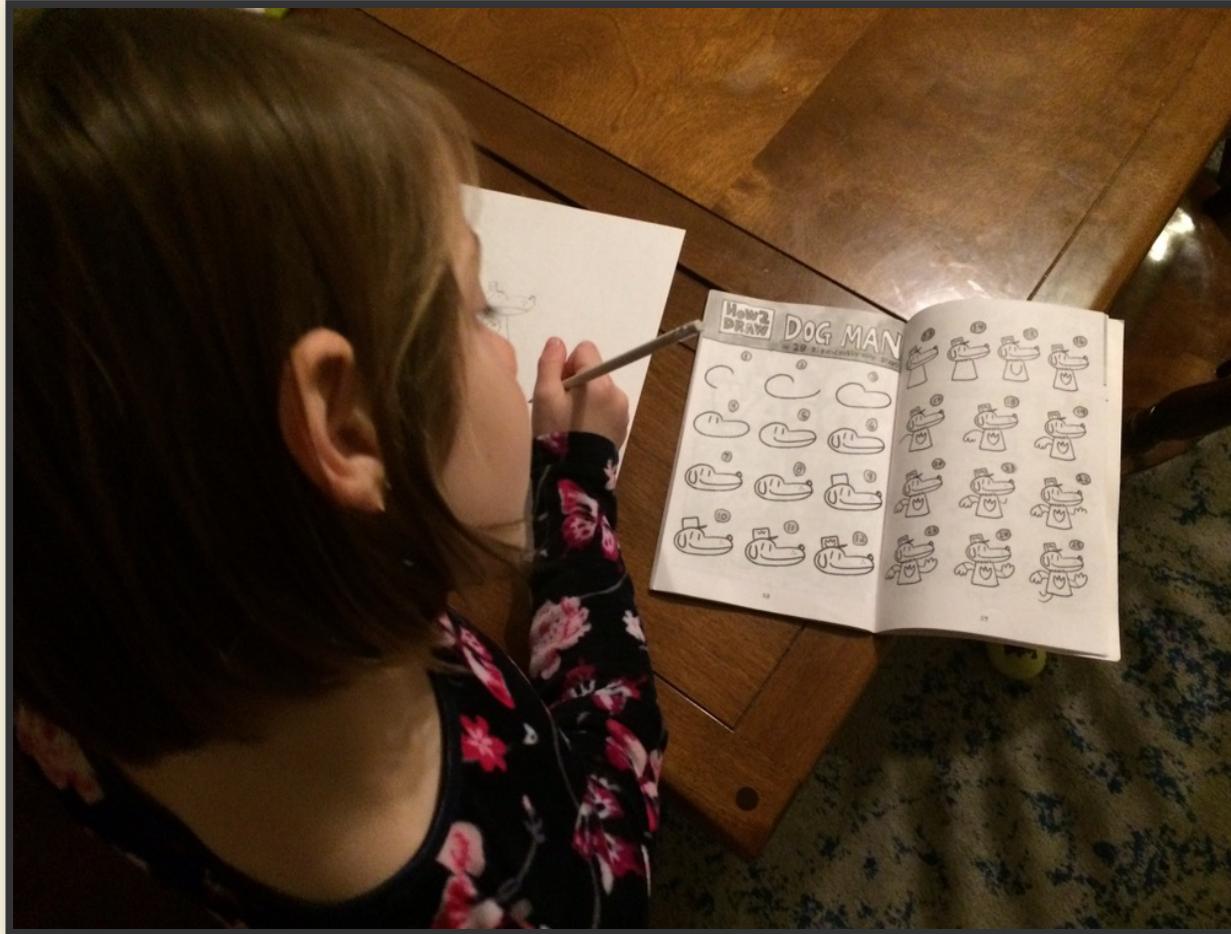


3. GAPS IN INFORMATION USABILITY

The screenshot shows the website for 'Simplifying Complexity'. The header includes the logo, tagline 'how to make complex systems easy to understand', and navigation links for 'Blog' and 'Contact'. A search bar is located on the left side. Below the search bar, there are links to 'Collapse All' and 'Expand All'. A sidebar menu is visible, listing several articles under the heading 'Simplifying Complexity', including 'Introduction' and six numbered principles. The main content area displays a breadcrumb trail 'I'd Rather Be Writing / Simplifying Complexity' and the title 'Simplifying complexity'. The text of the article begins with 'Ask a technical writer what they do, and the most common response is that tech write...' and continues with 'In systems that are complex (supporting many tasks, options, functions, and informati... goals and create clear instructions on how users can achieve those goals. The technic... in his or her ability to simplify information in these complex systems.' The next paragraph starts with 'But for all the talk about simplifying complexity, there's not a lot written on how to do i... complex processes and concepts easy to follow? How do you help users achieve thei... sophisticated, confusing applications and code? Explaining concepts in plain languag... steps is a great start, but it's not the entire solution. We need to increase our skill at th... writers need to center on what's complex for users, and use all the tools available to tr... factor in ways that add real value to organizations.' The final paragraph begins with 'I'm fascinated by the idea that my core value as a technical writer involves helping use... complex systems. We should center our efforts around the user's greatest pain, becau... deliver the greatest value — and that is also where the space is the most interesting. V... documenting simple, obvious instructions, or by providing instructions for interfaces a

P1: GIVE USERS A MAP

FIRST WE SPLIT COMPLEX PROCESS INTO CHUNKS



MAPS BRING THE PIECES TOGETHER

Appstore Submission

[Collapse All](#) | [Expand All](#)

- Get Started with App Submission
 - Getting Started with App Submission
 - Understanding Amazon Appstore Submission
 - Appstore Publishing FAQ
 - Setting User Permissions for the Amazon Appstore
 - Index of Documentation FAQs
- Prepare Your App
- Publish Your App
 - Submitting Apps to the Amazon Appstore

Step 1: Log in and Add an App



Before you can submit an app, you must register for a developer account (it's free) and add an app to the Developer Console. When you create an app, you will complete general information about your app.

P2: MAKE INFO DISCOVERABLE AS NEEDED

COMPLEX SYSTEMS CONTAIN FEEDBACK LOOPS

Each new piece of data the user uncovers affects the path taken and the eventual outcome. ... it does not lend itself to being performed with a defined set of tasks nor can those tasks be performed in a fixed order.

– Michael Albers, *Content and Complexity*

SINTERING AS AN ANALOGY



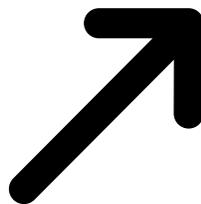
P3: ENSURE HARMONY ACROSS ALL DOCS

SYSTEMS DEVELOPED INDEPENDENTLY BUT INTERACT TOGETHER

Specialization is required in order to understand more and more about the intricate systems around us.... But at the same time, the systems we are building ... are not only intricate and complicated, but also stitch together field after field.... The design of driverless cars is a good example, requiring collaboration among ... software, lasers, automotive engineering, digital mapping, and more. — Samuel Arbseon

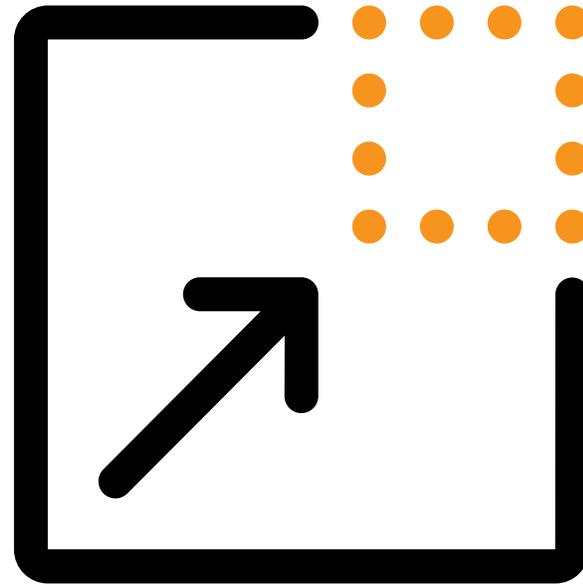
LOOKING FOR FIT ACROSS THE DOCS, BLOGS, FORUMS

addition of
new information
into a larger whole



P4: REDUCE AND DISTILL TO ITS ESSENCE

COMPRESS INTO SMALLER, MORE CONSUMABLE UNITS



distillation
of information
from a larger
whole

ARTICLE SUMMARIES

NN/g Nielsen Norman Group

Evidence-Based User Experience Research, Training, and Consulting

Search

[HOME](#) [TRAINING & EVENTS](#) [CONSULTING](#) [REPORTS](#) [ARTICLES](#) [ABOUT NN/G](#)

Topics

[E-commerce](#)
[Intranets](#)
[Mobile & Tablet](#)
[User Testing](#)
[Web Usability](#)
[Writing for the Web](#)
[▶ See all topics](#)

Recent Articles

[In Defense of Post-its](#)
[Retain UX Talent by Tracking UX Capacity](#)
[Card Sorting: Uncover Users' Mental Models for Better Information Architecture](#)
[The Two UX Gulfs: Evaluation and Execution](#)
[The Principle of Commitment and Behavioral Consistency](#)
[See all articles](#)

Popular Articles

Remote Moderated Usability Tests: How and Why to Do Them

by [KATE MEYER](#) and [KARA PERNICE](#) on March 25, 2018

Topics: [User Testing](#) [Research Methods](#) [Agile](#)

Summary: Remote unmoderated usability testing is so fast and easy that some teams make it their only evaluation method. But don't shy away from its more robust alternative, the remote moderated usability test, which can give you more information and is also inexpensive.

Few teams have [enough time and resources to perform as much in-person usability testing as they'd like](#). Acting under the (correct) assumption that any user data is better than no data, many turn to [quick and cheap methods](#) for usability testing.

[Unmoderated usability testing](#) (also known as asynchronous testing) is a popular way to get a product tested by users without breaking the bank. It usually involves using one of the many available services (such as [What Users Do](#)), [setting up some tasks](#), and waiting for the data to be collected. This method has some

substantial benefits:

- No recruiting (if you're using the built-in panels of users that the remote-testing services provide)
- No moderation skills needed
- Easy test setup
- Fast results

QUICK REFERENCE GUIDES



Lorum Ipsum
Quick Reference Guide

Im alit aliquat, consete miniduisim at nons ating er incin ulla adip essequi pismod exeros augat velenis augat lore dolendit velit at iure delis nullam, quat, summandreet wismolu is acinin voloperer alit iliquat la am.

Duis acinin voloperer alit iliquat la am, sum dolore dolore ver suscip erostrud tinit wis diat velent er ipit non hent augat, quis:

- duis acinin voloperer alit iliquat umsandigna facil dolor at, sectetum venisi.
- Lorum ipsum dolor solaritanut est tudat repulare in gratidueni la palabrata entotum.



Quatumsan Venisi

Schedulare Voloper Est Tudat

- Fro alit acillam etue dit ecle vullaore faccum quatumsan velesti onsent del in henit wis dunt dit aci et nullan.
- Olesse dipismo doloboreet ipsummy nih elit non henim dolummy num zzzit ate faccum zzziliquam, quam quip eroslis del er sed.
- Ex et, sumsandigna faoil dolor at, sectetum venisi.

Raesequis eu feugiate

- Famconsequi blaorem vulla ad tsit dolessi exeraessi.
- Camconsequi blaorem vulla ad tsit dolessi exeraessi:
 - Dio eugimoocons nos esto: dio euglamcons nos esto consequis amconsequi blaorem vulla ad tsit dolessi exeraessi.
 - Faciduisicint ea com num: consequis amconsequi blaorem vulla ad tsit dolessi ex, consequis amconsequi blaorem vulla ad tsit dolessi ex
 - Orem valluuar in exero: Endre dolent ulla core magna augait nim dunt eroiduipit utat wissed diamet inisit illan verosto odolorero odigna faciduisicng ea commy num velisi.
 - Faccumsum vulpatatue core: Verosto odolorero odigna faciduisicng ea commy num velisi. diamet inisit illan verosto odolorero odigna faciduisicng ea commy num velisi.

Aelesse quipit accu	Selenit er ipit
1. d tinit wis diat velent er ipit non hent augat, quis ex etue fe.	1. dunt dit aci et nullan vollesse dipismo doloboreet ipsum
2. Cd tinit wis diat velent er ipit non hent augat, quis ex et	2. quatumsan velesti onsent del in henit wis dunt dit aci et nullan vollesse dipismo doloboreet
3. Is diat velent er ipit non hent augat, quis ex etue feum iustrud te feugaal accum dolummo lorecillis	3. ipsummy nih elit non henim dolummy num zzzit ate
Strud te feugaal accum dolummo lorecillis do odipsum tonum ius.	4. quatumsan velesti onsent del in henit wis dunt dit aci et nullan vollesse dipismo doloboreet
Diat velent er ipit	Vea coreetue tie
an hendip ea coreetue tie velenis alit acillam etue	Taugait, quis ex etue feum iustrud te feugaal accum dolummo lorecillis do odipsum tonum ius
1. an hendip ea coreetue tie velenis alit acillam etue dit ecle vullaore faccum quatumsan ve	Quics ex etue feum iustrud te feugaal accum dolummo lorecillis do odipsum tonum ius.
2. wis dunt dit aci et nullan vollesse	Dolummo lorecillis do odipsum tonum iu.
3. dunt dit aci et nullan vollesse dipismo doloboreet ipsum	

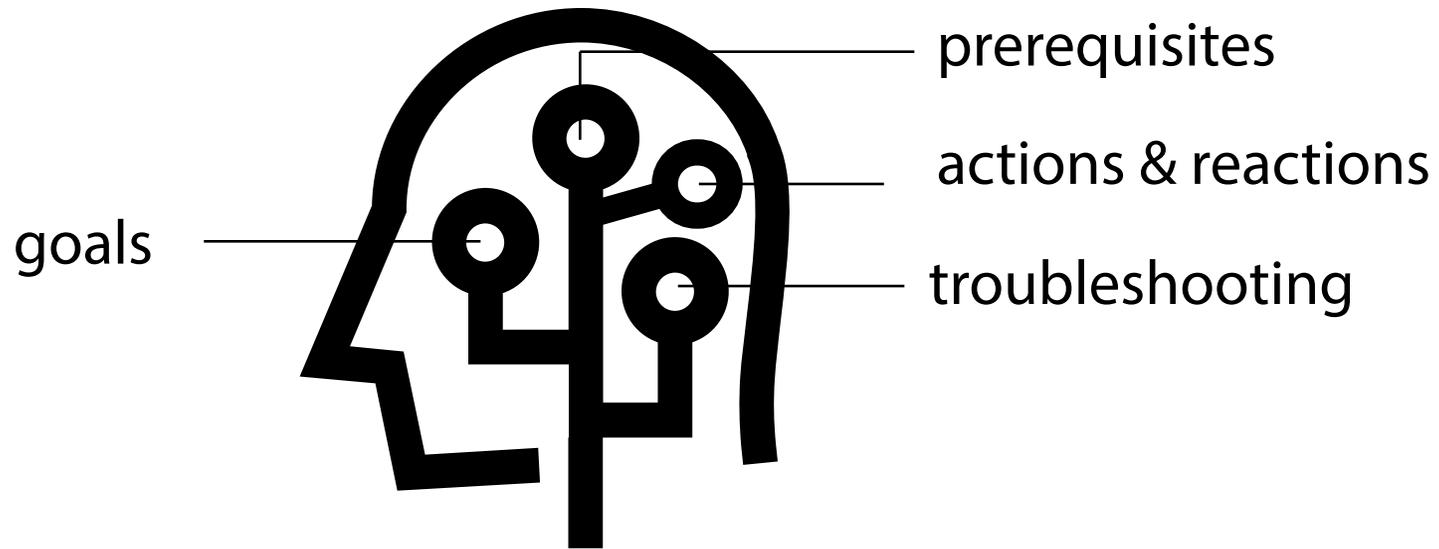
© 2008 ACME Corporation, Inc. All Rights Reserved

P5: CONFORM TO GENRE EXPECTATIONS

SCHEMA THEORY

By catering my design to meeting your experiences, I make these items easier for you to use in that context. – Kirk St. Amant

FOUR COMPONENTS MODEL



NARRATIVE PARADIGM

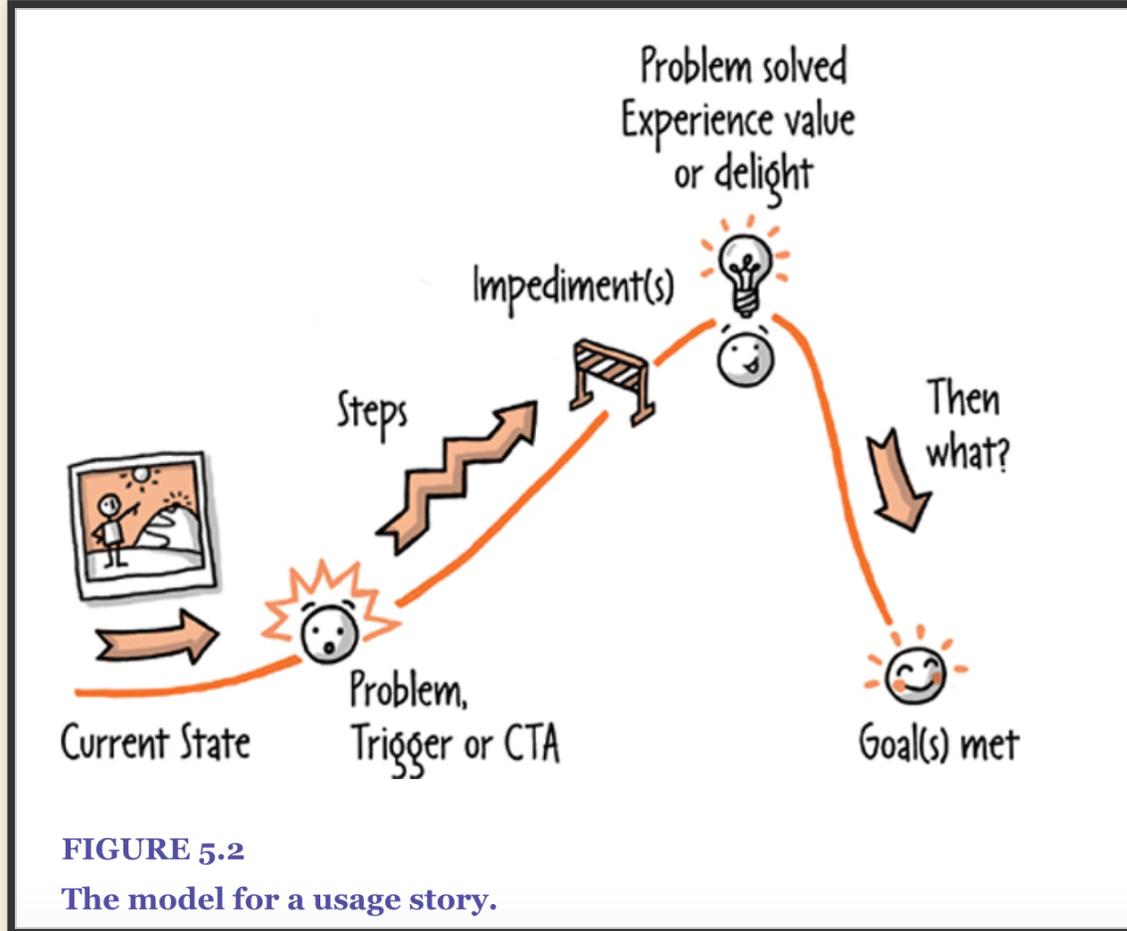
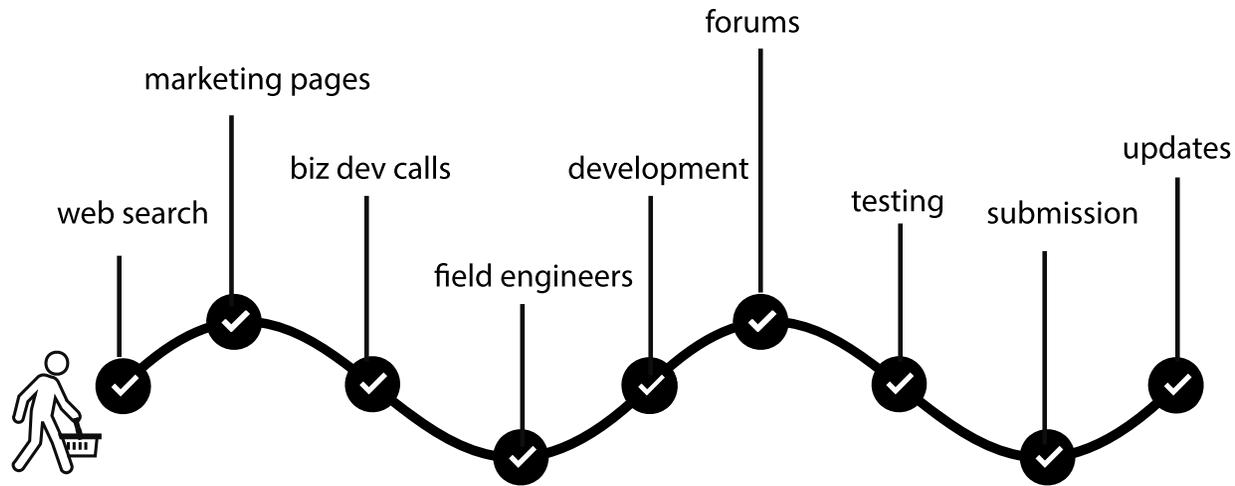


FIGURE 5.2
The model for a usage story.

Image from *The User's Journey: Storymapping Products That People Love* by Donna Lichaw

STRUCTURING INFORMATION AROUND THE CUSTOMER JOURNEY

Customer Journey



P6: REDUCE LANGUAGE COMPLEXITY

I came across a set of API resources for managing a DEG the other day. You could add, update, delete and get DEGs. You can also pull analytics, history, and other elements of a DEG. I spent about 10-15 minutes looking around their developer portal, documentation, and even Googling, but never could figure out what a DEG was. Nowhere in their documentation did they ever tell consumers what a DEG was, you just had to be in the know I guess.

— Kin Lane

[NOT] RECOGNIZING FAMILIAR TERMS



BACKGROUND KNOWLEDGE SECTIONS

Background Knowledge

Because Amazon's Fire OS is based on Android, Amazon tries to maintain as much parity with Android development as possible. Because of this, the documentation here doesn't duplicate the information in the Android documentation; instead, it covers how Amazon and Fire OS differs. For a better understanding of the concepts here, consult these foundational Android documentation topics:

- [Device Compatibility](#)
- [Filters on Google Play](#)
- [Supporting Different Platform Versions](#)
- [Permissions that Imply Feature Requirements](#)
- [Multiple APK Support](#)
- [Creating Multiple APKs for Different API Levels](#)

It will also help to understand some common terms.

Key terms

For more glossary definitions, see the [App Submission Glossary](#).

P7: ITERATIVE DESIGN OF DOCS

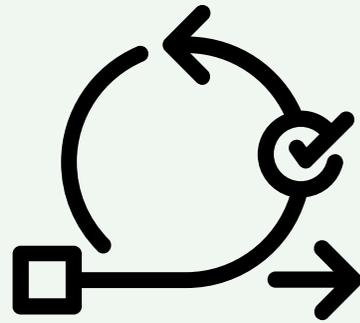
DOCS AS A "THEORY" TO TEST

In a sense, many things on a software development project are theories, or more accurately, assertions that need to be evaluated. ...

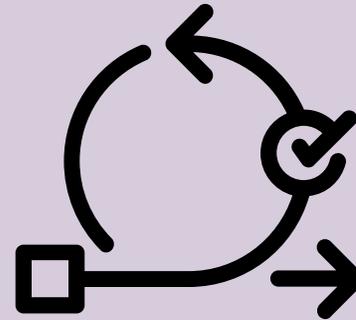
Just because some stakeholders ... say a requirement is valid does not mean that they are correct. We need to evaluate even the requirements to determine whether they define the right solution to the problem at hand. —

Spence and Bitner

UPON INITIAL PUBLICATION, BEGIN ITERATIVE CYCLES



Software
development



Documentation
development

RECAP OF ARGUMENT

- Technology is getting simpler on the front-end for end-users
- But the code underneath is becoming increasingly specialized/complex
- Tech writers are generalists, not specialists
- To provide value in specialist contexts, tech writers must exploit the gaps
- These gaps are (1) doc tools/processes, (2) understanding user feedback/experiences, and (3) information usability

WHERE TO GET MORE INFO

The screenshot shows the website 'I'd Rather Be Writing' with the tagline 'exploring technical writing trends and innovations'. The navigation menu includes 'About', 'Archives', 'Jekyll', 'Academics & Practitioners', 'Beginners', 'Complexity', 'API doc', and 'Podcasts'. A search bar is located on the right. The main content area features an article titled 'Tech comm trends: Providing value as a generalist in a sea of specialists (Part I)' by Tom Johnson, dated Oct 2, 2018. The article is part of a series and is categorized under 'api-doc', 'simplifying-complexity', and 'writing'. A summary states: 'Technical writing jobs have shifted more from the end-user domain to the developer domain. This creates challenges because most technical writers are generalists, not specialists, when it comes to technology. In these specialist context, technical writers can add value by focusing on (1) authoring/publishing processes and tools, (2) knowledge of the user experience, and (3) information usability.' A 'Contents' section lists: 'Series summary', 'Introduction', 'Argument overview', and 'Interactive surveys'. On the left sidebar, there is a newsletter subscription form with the text 'Stay current with the latest in tech comm' and 'Keep current with the latest trends in technical communication by subscribing to the I'd Rather Be Writing newsletter. 4,500+ subscribers'. Below the form are social media icons for RSS, Twitter, YouTube, LinkedIn, Facebook, and GitHub. A 'Recent Comments' section shows a comment by Eric Weston: 'Done right, I think DITA 'short descriptions' and

Essay series: <http://bit.ly/genandspecialisttrendspart1>

INTERACTIVE SURVEYS

Your reactions and input

UX professionals have reduced the need for technical writers to provide documentation for mainstream end-users.

- Strongly Disagree
- Disagree
- Undecided
- Agree
- Strongly Agree

Many technical writing jobs have shifted from the end-user domain to the developer domain.

QUESTIONS?



THE END



Tom Johnson

- idratherbewriting.com
- [@tomjohnson](https://twitter.com/tomjohnson)
- tomjoht@gmail.com