

# Starting Documentation Early in the Development Process

Linda Urban

Linda Urban Communication Services

[www.urbancreations.com](http://www.urbancreations.com)

Contact: [lurban@earthlink.net](mailto:lurban@earthlink.net)

# About the speaker

- ▶ 20+ years as a Bay Area Technical Communicator
- ▶ Consulting/contracting business
  - Analyze and plan what documentation is needed
  - Design and develop documentation
  - Work on user interface with programmers and others on development teams
  - Incorporate usability techniques
- ▶ Teach in Technical Communication certificate programs
  - UC Berkeley Extension
  - UC Santa Cruz Extension

# About you

## ▶ Students?

- Computer science (programmers)
- Writing or journalism
- Other

## ▶ Faculty?

- Computer science
- Other?

## ▶ Work in software or other product development (Industry?)

- Programmers
- Engineers
- Technical writers
- Product managers
- Other

## ▶ Size of organization?

- Large company
- Small company
- Start-up

# About you, cont'd

- ▶ Do you write documentation?
- ▶ Do you work with those who do?
- ▶ Are you a SME (subject matter expert) for writers?
- ▶ Familiar with user-centered design?
  - Know about it?
  - Use it?
  - Promote it?

# Overview

- ▶ Defining “documentation” for this talk
- ▶ Examples of documenting at the end
- ▶ Advantages to starting early
- ▶ Questions tech writers ask are key for product usability
  - Who the users are
  - What will product be used for?
  - Context in which they will use it
- ▶ Keep focus on user perspective
  - Surface features and concepts that are hard to explain, overly complex
- ▶ Result in improved product as well as documentation
- ▶ Techniques for planning and developing documentation as the product shifts

# What is documentation?

## “User documentation” Print and/or Online

- ▶ User manual (concepts and tasks)
- ▶ Quick Start
- ▶ Tutorial
- ▶ Reference manuals (varied)
  - Function Reference
  - Other “look ups”
- ▶ Administrator guide
- ▶ Installation
- ▶ Maintenance and Repair

## Other Documentation

- ▶ Requirements
- ▶ Specifications and design documents
- ▶ User Interface Style Guides
- ▶ Comments in code
- ▶ Text in the user interface
- ▶ Error messages
- ▶ Marketing materials (data sheets, whitepapers, etc)
- ▶ Training materials
- ▶ Support knowledgebase (internal or external)

# What is documentation?

## **“User documentation” Print and/or Online**

- ▶ User manual (concepts and tasks)
- ▶ Quick Start
- ▶ Tutorial
- ▶ Reference manuals (varied)
  - Function Reference
  - Other “look ups”
- ▶ Administrator guide
- ▶ Installation
- ▶ Maintenance and Repair

- ▶ **Text in the user interface**
- ▶ **Error messages**

## **Other Documentation**

- ▶ Requirements
- ▶ Specifications and design documents
- ▶ User Interface Style Guides
- ▶ Comments in code
- ▶ Marketing materials (data sheets, whitepapers, etc)
- ▶ Training materials
- ▶ Support knowledgebase (internal or external)

# Who writes it? (traditional)

## “User Documentation” Print and/or Online

- ▶ User manuals: **Tech writer**
- ▶ Online help: **Tech writer**
- ▶ Text in the user interface: **Programmer**
- ▶ Error messages: **Programmer**

## Other Documentation

- ▶ Requirements: **Product marketing, product management**
- ▶ Specifications and design documents: **Software developers (programmers)**
- ▶ User Interface Style Guide: **UI Designer, Programmer**
- ▶ Comments in code: **Programmer**
- ▶ Marketing materials: **Product marketing, Tech writer**
- ▶ Training materials: **Tech writer, Trainer, Professional services**
- ▶ Support knowledgebase (internal or external): **Technical support**

# Who writes it? (ideal)

## “User Documentation” Print and/or Online

- ▶ User manuals: **Tech writer**
- ▶ Online help: **Tech writer**
- ▶ Text in the user interface: **Tech writer**
- ▶ Error messages: **Tech writer**

## Other Documentation

- ▶ Requirements: **Product marketing, product management**
- ▶ Specifications and design documents: **Software developers (programmers)**
- ▶ User Interface Style Guide: **UI Designer, Programmer, or Tech Writer**
- ▶ Comments in code: **Programmer**
- ▶ Marketing materials: **Product marketing, Tech writer**
- ▶ Training materials: **Tech writer, Trainer, Professional services**
- ▶ Support knowledgebase (internal or external): **Technical support**

# Who writes it? (reality)

## *Reality varies...*

- ▶ Large company: Specialized work, more likely to be “traditional”
- ▶ Very small company (startup): people wear many hats
  - Tech writer only at end (if any)
  - Or Tech writer writes most docs
  - Programmer writes docs
  - Skip many docs
- ▶ Many companies: Limited resources, Rushed schedules
  - Specialized when possible
  - Multiple hats, double-up responsibilities
  - Shortcuts: Drop some documentation

# Who writes it? (limited resources)

## "User Documentation" Print and/or Online

- ▶ User manual: **Limited**
- ▶ Online help: **or at end**
  
- ▶ Text in the user interface: **Programmer**
- ▶ Error messages: **Programmer**

## Other Documentation

- ▶ Requirements: **Limited** ing,  
product manag
- ▶ Specifications and documents: **Limited** rs  
So  
(programmers)
- ▶ User Interface ~~Style Guide~~: **UI Designer, Programmer**
- ▶ Comments in code: **Programmer**
- ▶ Marketing materials: **??**  
marketing, Tech writ
- ▶ Training materials: **??**  
Trainer, Professiona
- ▶ Support knowledgebase (internal or external): **??**  
Tech

# Mindset when documenting at end

- ▶ Focus is on documenting *something*
- ▶ Purpose of documentation
  - Have documentation
  - Document what's in the product
  - Have it be accurate

# Examples



# January, not too many years back

- ▶ Need two manuals: Admin, End User
- ▶ New product shipping in 8 weeks. (Will NOT slip)
- ▶ Results:
  - “Ok” Admin Guide
    - ▶ Judged by readability, task-orientation, information coverage
    - ▶ No index
  - “Ok” End User Guide, same criteria
  - Address user needs - ??
  - Adequate explanation of technical concepts - ??
  - Ability to improve user interface – Zero
  - Hooked into user interface (context-sensitivity) -

# This past summer

- ▶ Update administrator guide with changes for new version shipping in 3 weeks
- ▶ Results:
  - Quality of information? – Not great
  - Screenshots updated
  - New information added...in places
  - Improvement of existing information? – Zero
  - Ability to improve user interface – Zero

# More typical (new product or major release)

- ▶ Shipping in 3 months, time to start the docs
  - (Product has been in development 3 to 9 months)
- ▶ At some point, project slips 2 weeks to 3 months
- ▶ Actual time for docs: 3.5 to 6 months
- ▶ Results:
  - “OK” docs (judged by readability, task orientation, information coverage)
  - Additional time has allowed additional coverage, maybe some user contact
  - Ability to impact user interface – Maybe
  - Hooked into user interface (context-sensitivity) – Doubtful

# Missing

- ▶ More than bare bones analysis of what the user needs in the product and the docs
- ▶ An opportunity to improve the product

# Documentation is part of the product

- ▶ It should be developed as such
- ▶ It's expensive to develop the wrong documentation

Technical communicator's skills are more than writing documentation

# Mindset when documenting early

- ▶ Understand user needs
- ▶ Understand goals for product and for documentation
- ▶ Understand concepts
- ▶ Design appropriate documentation
- ▶ Collaborate with product designers and developers

# High-level questions

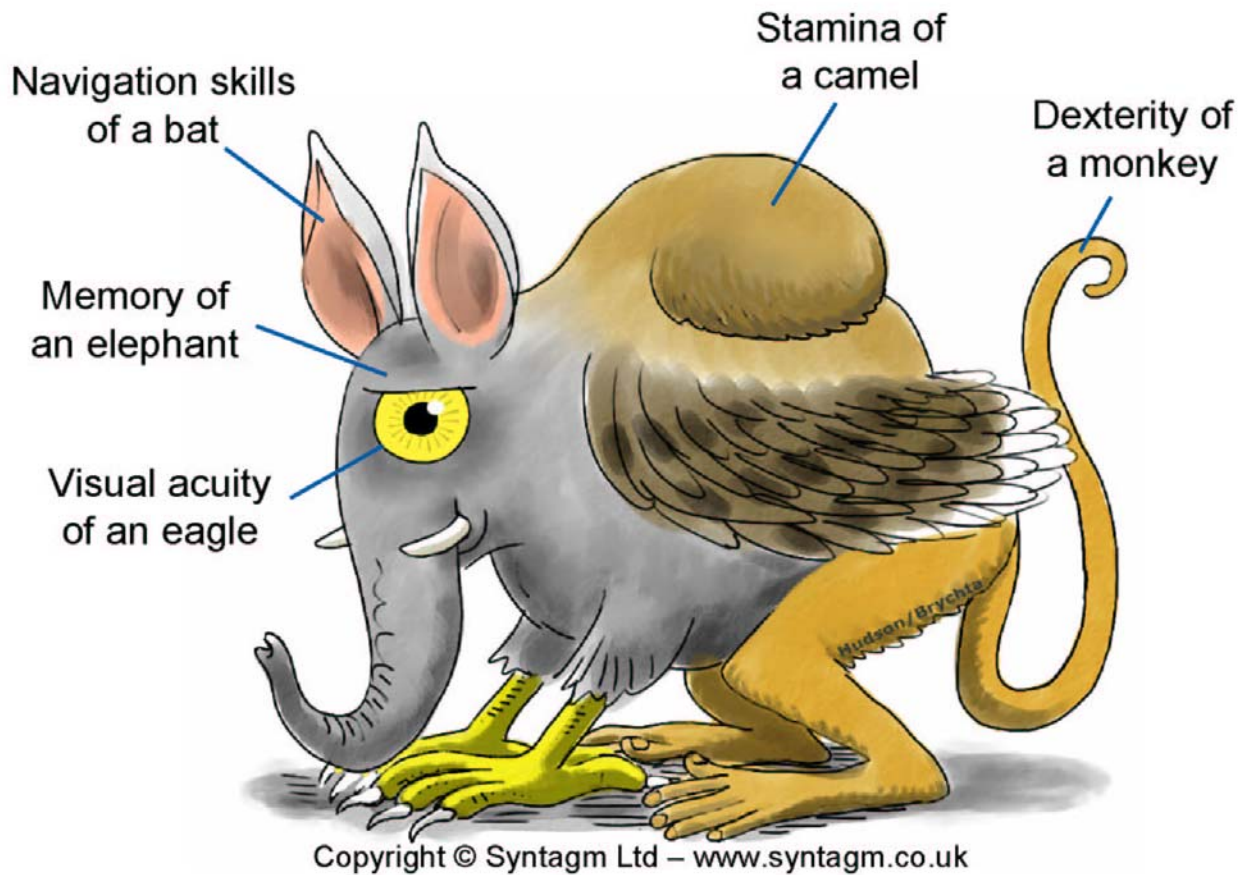
- ▶ What will product be used for?
- ▶ Who are the users?
- ▶ When and where will they use it?
- ▶ What words do they use to describe tasks the product lets them do?

# Questions when looking at requirements or specs

- ▶ What do those terms mean? (on the screen, in menus)
- ▶ Why are these features grouped together?
- ▶ What order are these functions done in?

# Focus on user experience

- ▶ User experience is of LEARNING and USING the product
- ▶ Not on mastering product features
- ▶ Reminder of context of use: user's context
- ▶ Get into the field, talk to users
  - Interviews
  - Task analysis
  - Observation



# The Perfect User

"This creature is known as an elecamonglebat, being an unusual hybrid of elephant, camel, monkey, eagle, and bat. It represents the perfect user during design meetings. Unfortunately, it is exceedingly rare in the wild." Image copyright William Hudson and Alex Brychta. From "User-Centered Design with Use Case Methods," a tutorial by William Hudson. <http://www.syntagm.co.uk/ucuml.htm>

# When learning about product, the writer discovers

- ▶ Features that are complex
- ▶ Concepts that are hard to explain

What to do with that information?

- ▶ Write better documentation?
- ▶ Yes, but...

# The sad truth

▶ No one *wants* to read the documentation...

...and mostly, they avoid it

So, what to do?

▶ Use the writer's skills and insight to improve the product

(As well as develop the documentation)

Much more satisfying, for everyone!

# Product Visions



As proposed by the project sponsor.



As specified in the project request.



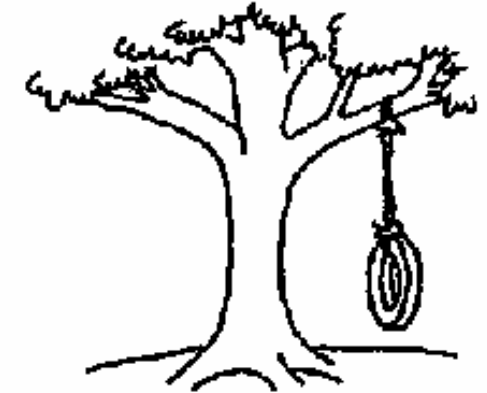
As designed by the senior analyst.



As produced by the programmers.



As installed at the user's site.



What the user wanted.

From Fred Tepfer's website at Oregon State:

<http://www.uoregon.edu/~ftepfer/SchIFacilities/TireSwingTable.html>

# What writers can do, when there's no product yet

- ▶ Analyze what's needed
- ▶ Understand audience
  - Identify
  - Observe
  - Interview
- ▶ Task analysis
- ▶ Design documents
- ▶ Create content spec
- ▶ Create prototypes
- ▶ Do usability analysis

## Improves

- ▶ Doc and Product
- ▶ Doc and Product
- ▶ Doc and Product
- ▶ Doc
- ▶ Doc
- ▶ Doc and Product
- ▶ Doc and Product

# Review the UI and workflow for clarity and usability

- ▶ Review terminology and grammar
  - Labels groupings
- ▶ Define terms, and reach agreement on what things are called
- ▶ Identify jargon, and help eliminate it
- ▶ Review task flow (map user tasks to product)

# How starting early helps the documentation

- ▶ Make sure you create the right documents, for the right users
- ▶ Clarify UI, thereby simplifying the job documentation needs to do
- ▶ Start in time to produce better documents
- ▶ Shorten production schedule

# How starting early helps the product

- ▶ Improve the UI and work flow
- ▶ Bring user perspective to the development team
- ▶ Develop tools the whole team can use  
(Example: Personas, UI Style Guides)
- ▶ Provide drafts early
  - When people read the manual is often when people begin to recognize problems
  - I didn't know it did that
  - It's not supposed to do that
  - It doesn't work that way (oh YES it does!)

# What if the writer has other projects?

- ▶ This is typical
- ▶ Still good to be involved, even if it's on a very partial basis
- ▶ Ask appropriate questions
- ▶ Bring user viewpoint to the design and development process

# Techniques for technical writers

Writing during development means writing about shifting content. So what to do?

- ▶ Focus on user needs and tasks
- ▶ Create a content spec
- ▶ Map user needs and tasks
  - to the product
  - to planned content
- ▶ Design and create sample pages (prototype)
- ▶ Develop a style guide
  - Include structure as well as look and feel, terminology
- ▶ Write concepts first
- ▶ Include tasks in the outline, but hold off on details

# Thank you!

▶ Questions?

▶ Discussion, Q&A

▶ Contact Linda at: [lurban@earthlink.net](mailto:lurban@earthlink.net)