"Everything" about scientific software documentation that wasn’t in the manual

Aleksandra Pawlik

supervisors: Judith Segal, Helen Sharp, Marian Petre
Department of Computing, The Open University, UK

Scientific Software Days
Austin, TX 17th December 2012
Talk overview

1. Empirical studies of (scientific) software development
2. Documentation in scientific end-user development
3. Documentation beyond scientific end-user development
4. Community’s role in producing documentation
5. Crowd-sourcing documentation - implementation
6. Benefits of crowd-sourcing documentation
7. Challenges in crowd-sourcing documentation

"Everything" about scientific software documentation that wasn’t in the manual
Empirical studies of (scientific) software development

• Real-life situations and activities
• The importance of the context
• Actual software development practices
• The human factor
Study data

33 Interviews:
Scientists who commercialized their research software
Scientists developing scientific software
Scientists using scientific software

Case study: SciPy/Numpy Documentation Project:
8 interviews with key stakeholders
10 years of 3 mailing lists archives
2 Progress Reports (SciPy Conference proceedings)
Documentation system data and logs
Documentation production in scientific software development contexts

Context 1:
Scientific end-user development
Documentation production in scientific software development contexts

Context 1:
Scientific end-user development

Context 2:
Scientific software developed for and used by a wider user community
Scientific end-user development (Context 1)

- Advancing research - the main aim
- The developer is the (sole) user
- Typically no other developers
- One-off use software

Therefore...
documentation production too big an investment

• If anything is well documented, it’s typically the scientific model

• Scarce or non-existent technical documentation

• Comments in the source code often understandable only to the original developer

• No user manuals
Seems reasonable but...

Documentation production supports reasoning process

"I never felt the need to document it [when developing for own use]. In hindsight I think it would have been a good idea because it makes you think about what the code is actually doing..."  [Scientist-developer A]
Seems reasonable but...

Documentation production supports reasoning process

"I never felt the need to document it [when developing for own use]. In hindsight I think it would have been a good idea because it makes you think about what the code is actually doing..."  [Scientist-developer A]

No documentation - reproducibility issues

"I reckon I repeated 3.5 years of work in 6 months at the end of my PhD. If stuff had been better documented, then it would have probably been more like 2 months. I probably wasted 4 months retrying the wrong thing because I had not made sufficiently good notes."  [Scientist-developer B]
Seems reasonable but...

Documentation production supports reasoning process

"I never felt the need to document it [when developing for own use]. In hindsight I think it would have been a good idea because it makes you think about what the code is actually doing..."  [Scientist-developer A]

No documentation - reproducibility issues

"I reckon I repeated 3.5 years of work in 6 months at the end of my PhD. If stuff had been better documented, then it would have probably been more like 2 months. I probably wasted 4 months retrying the wrong thing because I had not made sufficiently good notes."  [Scientist-developer B]

And what if....
...software is developed for a wider community? (Context 2)

- Users: manuals, tutorials, examples
- Users represent a continuum:
  
  **End-users**
  
  **User-developers**

| BLACK BOX USERS | WHITE BOX USERS |
...software is developed for a wider community? (Context 2)

- Users: manuals, tutorials, examples
- Users represent a continuum:
  - End-users
  - User-developers
- Software maintenance: technical documentation
- Developers often belong to the same community as users
Does that influence documentation production?

• Tacit knowledge - informal documentation

• Assumptions about users’ knowledge related to: scientific domain as well as IT & general computing
Problems with documentation are still there

"..it’s never fun to do the documentation. It’s boring. So you do it [develop the software] but you never properly document that. Then you forget yourself how it works.”  [Scientist-developer C]
Problems with documentation are still there

"..it’s never fun to do the documentation. It’s boring. So you do it [develop the software] but you never properly document that. Then you forget yourself how it works.”  [Scientist-developer C]

"We weren’t really spending time on the documentation. In the end it wasted a lot of time because we couldn’t remember quite what we did. (...) we couldn’t remember how we did things so when the program didn’t work we had quite a long time rectifying it."  [Scientist-developer D]
Problems with documentation are still there

"..it’s never fun to do the documentation. It’s boring. So you do it [develop the software] but you never properly document that. Then you forget yourself how it works.” [Scientist-developer C]

"We weren’t really spending time on the documentation. In the end it wasted a lot of time because we couldn’t remember quite what we did. (...) we couldn’t remember how we did things so when the program didn’t work we had quite a long time rectifying it." [Scientist-developer D]

It’s evident in all the data sources that lack of documentation is a major cause of problems!
Main challenges in documentation production

• Lack of time and resources.

• Nature of research - impossible to predict its direction

• Dynamic users’ needs

• Users finding new applications for the software... especially user-developers
Where do users get information about the software?

• Consult the community and share experiences

• Use research publications, conferences, mailing lists, internet forums....

• Deploy the potential of communities and networks of practice
Advantages of consulting the user community

- Cumulative knowledge about software
- Collection of different experiences coming from different viewpoints
- Peer-to-peer understanding
- Inspirational ideas

"Everything" about scientific software documentation that wasn’t in the manual
Challenges in consulting the user community

- Finding those who know
- Taking up people’s time
- Competitive research environment
Challenges in consulting the user community

• Finding those who know

• Taking up people’s time

• Competitive research environment

"If they working on the same problem, you don’t know that, then that may spur them to write the paper quicker. You may end up in a worse position. (...) Sometimes people are working on things and they discover that other people are working on the same thing and then it’s a bit of a race to finish. It’s not fun.”  [Scientist-user A]
But still, the user community generates a lot of useful documentation

"If it still doesn’t work, I will then look up examples. People often have forums where they ask questions and they do things which are similar. I see how other people have done it and try to understand what is going on." [Scientist-user B]

"Everything" about scientific software documentation that wasn’t in the manual
But still, the user community generates a lot of useful documentation

"If it still doesn’t work, I will then look up examples. People often have forums where they ask questions and they do things which are similar. I see how other people have done it and try to understand what is going on."  [Scientist-user B]

Crowd-sourcing documentation?
Is crowd-sourcing documentation feasible?

SciPy/Numpy Documentation Project

- “Scratching one’s personal itch”
- Securing resources
- Finding a leader / project coordinator
- It’s been out there since 2008: docs.scipy.org
Setting up the infrastructure

COMMUNITY

DEVELOPERS WITH WRITE ACCESS TO THE REPOSITORY

Edit docstring

Numpy Online Doc System

Generate patch

Docstring patch

Apply patch

Numpy code repository

"Everything" about scientific software documentation that wasn’t in the manual
Standards and quality control

- Numpy/SciPy Docstring Standard: the community discussion
- The workflow: Editing + Proofing + Reviewing
- Editors negotiating changes of docstrings
Engaging the community

"Everything" about scientific software documentation that wasn’t in the manual
Keeping up the momentum

• Documentation Marathon 2008
• Progress monitoring - automatic statistics
• Setting up milestones
• Reporting back to the community (annual SciPy Conference)
• The 1000 words T-shirt reward
The Pareto principle

Number of words edited by editors

- 1-499 words: 61.0%
- 500-999 words: 8.5%
- 1,000-4,999 words: 15.3%
- 5,000-9,999 words: 8.5%
- 10,000-19,999 words: 3.4%
- 20,000-29,999 words: 1.7%
- 30,000-39,999 words: 1.7%
Crowd sourcing documentation: benefits

- Boost in documentation production: >76% coverage; from 8,521 words to over 140,000 words
- Lowering entry barriers - expanding the community
- Documentation written by users for users
- New stakeholders = new opinions, views and concepts
Crowd sourcing documentation: challenges

• New stakeholders = new opinions, views and concepts
• Time & resources investment
• Making it work long-term
Conclusions

• Documentation in scientific software - extended definition
• Tacit knowledge and informal information exchange
• Documenting scientific model essential but not sufficient
• Addressing different needs of different users
• Crowd sourcing documentation - balancing challenges and benefits
Thank you for your attention.

Questions?