A Systematic Approach to Creating and Maintaining Software Documentation

Allison L. Powell, James C. French, John C. Knight

Presented at the Symposium on Applied Computing - SAC ‘96
Philadelphia, PA, February 20, 1996

WHY SOFTWARE DOCUMENTATION AS AN APPLICATION AREA?

- Problems with the current paradigm:
  - Difficult to write and hard to use.
  - Inconsistent between project revisions.
  - No assurance that effort will pay off for end users.
  - Not designed to provide high quality responses to queries.

- Standards:
  - Exist, but are often ambiguous and not universally adopted.

- Experimental Systems:
  - HyperCASE (Cybulski & Reifs), Document Integration Facility (Garg & Scacchi).
  - Both DIF and SSDOS integrated database management systems with the software documentation associated with the software lifecycle and allowed queries on pre-defined keywords.

- Search accuracy.
- Fragment size/what to return.
- How to approach the problem when searching source code.
- How effective/useful is keyword searching on source code?
- What’s a more effective way?

- How to organize on-line software documents.
- How to present software documents when browsing.
- How to search software documents.

- How to approach the problem when searching source code.
- How effective/useful is keyword searching on source code?
- What’s a more effective way?

- How to organize on-line software documents.
- How to present software documents when browsing.
- How to search software documents.

RESOURCE-INTENSIVE, BUT RESULTS ARE OFTEN INADEQUATE OR INAPPROPRIATE

OUR APPROACH

- Consider multiple views.
  - newcomer needing concepts.
  - software engineer needing detail.
  - manager needing overviews.

- Attack the problem in two ways:
  - focus on retrieval:
    - making the documentation readily available.
    - authoring and organizing documentation with retrieval in mind.
  - focus on authoring:
    - Hypertext links for over 40 words and phrases.
    - Approx. 25 figures and tables.
    - Glossary
    - 100+ pages of system documents.
    - Hypertext links for over 40 words and phrases.

- Investigate most useful fragment size when responding to query.
- Explore ways to more fully integrate source code.
- Investigate most useful fragment size when responding to query.

- How effective/useful is keyword searching on source code?
- What’s a more effective way?

- Investigate most useful fragment size when responding to query.
- Gather feedback from software engineers working on the MSS project.
- Begin a document collection for the University of Virginia research reactor project.
- Explore ways to navigate document hierarchy.
- Explore ways to more fully integrate source code.

- Investigate most useful fragment size when responding to query.
- Gather feedback from software engineers working on the MSS project.
- Begin a document collection for the University of Virginia research reactor project.
- Explore ways to navigate document hierarchy.
- Explore ways to more fully integrate source code.

- Investigate most useful fragment size when responding to query.