SOFTWARE DOCUMENTATION

Discussion Topics

1. Introduction
2. Documentation Requirements
3. Process and Product Documentation
4. Document Quality
5. Standards
6. Conclusion

Introduction

- This class provides an overview of the
  - Reasons for software documentation
  - Types of software documentation

Documentation Requirements

- General requirements of all software documentation
  - Should provide for communication among team members
  - Should act as an information repository to be used by maintenance engineers
  - Should provide enough information to management to allow them to perform all program management related activities
  - Should describe to users how to operate and administer the system

Documentation Requirements

- In all software projects some amount of documentation should be created prior to any code being written
  - Design docs, etc.
- Documentation should continue after the code has been completed
  - User's manuals, etc.
- The two main types of documentation created are Process and Product documents.

Process Documentation

- Used to record and track the development process
  - Planning documentation
  - Cost, Funding tracking
  - Schedules
  - Standards Etc.
- This documentation is created to allow for successful management of a software product

Product Documentation

- Describes the **delivered** product
- Must evolve with the development of the software product
- Two main categories:
  - System Documentation
  - User Documentation

Product Documentation

- System Documentation
  - Describes how the system works, but not how to operate it.
- Examples:
  - Requirements Spec
  - Architectural Design
  - Detailed Design
  - Commented Source Code
    - Including output such as JavaDoc
  - Test Plans
    - Including test cases
  - V&V plan and results
  - List of Known Bugs

Product Documentation

- User Documentation has two main types
  - End User
  - System Administrator
- In some cases these are the same people
  - The target audience must be well understood!

Product Documentation

- There are five important areas that should be documented for a formal release of a software application
  - These do not necessarily each have to have their own document, but the topics should be covered thoroughly

1. Functional Description of the Software
2. Installation Instructions
3. Introductory Manual
4. Reference Manual
5. System Administrator’s Guide

Document Quality

- Providing thorough and professional documentation is important for any size product development team

- The problem is that many software professionals lack the writing skills to create professional level documents

Document Structure

- All documents for a given product should have a similar structure
  - A good reason for *product standards*
- The IEEE Standard for User Documentation lists such a structure
  - It is a superset of what most documents need
- The authors “best practices” are:

1. Put a cover page on all documents
2. Divide documents into chapters with sections and subsections
3. Add an index if there is lots of reference information
4. Add a glossary to define ambiguous terms

Standards

- Standards play an important role in the development, maintenance and usefulness of documentation

- Standards can act as a basis for quality documentation
  - But are not good enough on their own
    - Usually define high level content and organization

Standards

- IEEE
  - Has a published standard for user documentation
  - Provides a structure and superset of content areas
  - Many organizations probably won’t create documents that completely match the standard

Standard Templates

- Ensures that all the essential information is included
- A number of different large organizations such as the US Department of Defense and IEEE have defined standards for requirements documents
- Probably the most acceptable of these standards is the IEEE/ANSI 830-1993
- The standard recognizes differences between systems, and allows for some variations in the structure.
- There is a list of stable and variant parts:
  - Stable parts
  - Variant parts

IEEE/ANSI 830 - 1993

1. Introduction:

- Purpose of the requirements document
- Scope of product
- Definitions, acronyms and abbreviations
- References
- Overview of the remainder of the document

2. General Description:

- Product perspective
- Product functions
- User characteristics
- General constraints
- Assumptions and dependencies

IEEE/ANSI 830 - 1993

3. Specific Requirements

- Covering functional, non functional & interface requirements, these should document
  - External interfaces
  - Functionality
  - Performance requirement,
  - Logical database requirement,
  - Design constraints,
  - System attributes
  - Quality characteristics

IEEE/ANSI 830 - 1993

4. Appendices
5. index

A template based on IEEE/ANSI 830 – 1993

- Preface
- Introduction
  - Definition of the product, its expected usage and an overview of its functionality
- Glossary
  - Definition of technical terms and abbreviations used
- General user requirements
  - The systems requirements from the perspective of the user
- System architecture
  - A high-level overview of the anticipated system architecture, showing the distribution of functions across system modules