Capturing and Documenting Requirements

SFWR ENG 2B03

2003

Robert L. Baber

Introduction

- Problem, solution and software difficult to understand, change, maintain
- Requirements capture: process of asking questions
- Tough part: knowing when you have asked all the questions and received satisfactory answers

Background

- Who are the readers?
- What do they need to know?
- What background knowledge do they have, need?
- How will they use the document? For what?

Objectives of a Requirements Document

A requirements document

- is a reference resource
- should answer specific questions quickly
- is not a tutorial or a general explanation of the program and what it does
- records foresight about the life cycle of the system, likely changes, new functions, etc.

Content of a Requirements Document

- Specify externally observable behaviour *only*, not how to implement
- Specify constraints on the implementation, e.g. system software, hardware, interfaces to the given environment
- Anticipate “undesired” events (“errors”) and specify acceptable responses to them. (“Errors” to the user are not “errors” to the designer.)

Requirements Capture: Principles

- Formulate questions fully before answering them
- Separate concerns (“divide and conquer”)
- Be as formal as possible
  - avoid prose
  - be precise, concise, consistent, complete