

A Recipe for Scientific Computing Software Development

W. Spencer Smith^{a,*}, ?^a, ?^a

^a*Computing and Software Department, McMaster University, Hamilton, Ontario L8S 4L7, Canada*

Abstract

abstract

Keywords: Software Certification, Literate Programming, Document Driven Design, Scientific Computing

1. Introduction

- technology and processes exist, but still not used - need something more prescriptive

- Recipe for dev SC software - includes technology, process, templates etc.
- goals:

- Correct
- Verifiable
- ...
- Reproducible
- Simple technology

Figure 1 shows a high level view of the documentation to be produced.

Bottom up pieces are shown in Figure 2.

Introduce Figure 3, which shows template for the SRS. Adding a section for likely changes.

2. Concluding Remarks

Acknowledgements

acknowledgements

*Corresponding author

Email address: `smiths@mcmaster.ca` (W. Spencer Smith)

URL: `http://www.cas.mcmaster.ca/~smiths/` (W. Spencer Smith)

1. Problem statement.
2. Software requirements specification.
3. Verification and validation plan.
4. Module guide.
5. Module interface specification.
6. Literate code.
7. Verification report
8. Validation report.

Figure 1: Top-down view of software documentation following a rational process

1. Problem Statement
2. SRS: a) Symbols b) Goals c) Theoretical Models d) Instanced Models e) Data Definitions
f) General Data Definitions g) Assumptions h) Requirements i) Likely Changes
3. Verification and Validation plan: a) V&V techniques b) System level verification test cases
c) System level validation test cases d) Traceability to SRS
4. MG: a) Modules (name, secret, description) b) Traceability to SRS
5. MIS: a) Interface b) Description (formal or informal) c) Trace to MG
6. Literate Code: a) Code b) Traceability to numerical algorithm c) Traceability to requirements
d) Traceability to MG
7. Verification Report a) Interface b) Unit tests c) System tests
8. Validation Report a) Interface b) Comparison to experiments

Figure 2: Constituent parts of documents

1. Reference Material: a) Table of Symbols b) Abbreviations and Acronyms
2. Introduction: a) Purpose of the Document b) Scope of the Software Product c) Organization of the Document d) Intended Audience
3. General System Description: a) System Context b) User Characteristics c) System Constraints
4. Specific System Description:
 - (a) Problem Description: i) Background Overview, ii) Terminology Definition, iii) Physical System Description, iv) Goal Statements
 - (b) Solution specification: i) Assumptions, ii) Theoretical Models, iii) General Definitions, iv) Data Definitions, v) Instanced Models, vi) Data Constraints, vii) System Behaviour
 - (c) Non-functional Requirements: i) Accuracy of Input Data ii) Sensitivity of Model iii) Tolerance of Solution iv) Solution Validation Strategies v) Look and Feel Requirements vi) Usability Requirements vii) Performance Requirements viii) Maintainability Requirements ix) Portability Requirements x) Security Requirements
5. Other System Issues: a) Likely Changes b) Open Issues c) Off the Shelf Solutions d) Waiting Room
6. Traceability Matrix
7. List of Possible Changes in the Requirements
8. Values of Auxiliary Constants

Figure 3: Table of Contents of the SRS for FP