

Department of Energy (DOE)

Software Engineering Methodology

March 1996



U. S. DEPARTMENT OF ENERGY

Assistant Secretary for Human Resources and Administration
Office of the Chief Information Officer

TITLE PAGE

Document Name: Department of Energy
Software Engineering Methodology

Publication Date: March 1996

Contract Number: DE-AC01-95AD34277

Project Number: Task: 26-433-08-7-01
CDSI: 4001

Approval: _____
Wayne Jones, HR-433

U.S. DEPARTMENT OF ENERGY

Assistant Secretary for Human Resources and Administration
Office of the Chief Information Officer

The compilation/development of the *Department of Energy Software Engineering Methodology* was performed as part of a continuing effort to improve the quality, performance, and productivity of Departmental information systems. Many Departmental and contractor personnel were involved in the development of the document as contributors or reviewers.

The following team of Departmental and contractor personnel served on the Steering Committee for the project. They generated the philosophical framework for the lifecycle methodology and provided oversight and guidance during the initial development process.

Paul Barkley	Molly Jekel
Brenda Coblentz	Wayne Jones
Roger Collins	John Kramer
Mark Gordy	Kathy McGlaughlin
Cheryle Greenaugh	Philip Quarry
William Greiner	Richard Tittle

The following individuals were the principal contributors to the document.

Andrew Bicocchi	Marilyn Morein
Kathleen Canal	Margaret Webber
Denise Miller	

Credit must also be given to the many reviewers who provided invaluable input and feedback, and made useful suggestions for improving the content and presentation of information.

Paul Asbury	Ron Lauzon
Lewis Bachman	Fahy Low
Lorin Beehler	Michael Mullan
Carl Brown	Pam Pagnotta
Jo Buxton	Clif Quinn
Alison Campbell	Deborah Reynolds
Ann Heeney	Daniel Rupinski
Kim Hunter	Toni Russell
Anne Imrie	Pat A. Simmons
Gail Kallmyer	Sammy Slough
William King	Ruth Ann Smith
Milton Kowalewski	Kimberly Wandersee
Robert Kuehne	Harry Yocum

Chapter	Page
1.0 Introduction	1.0-1
1.1 Implementation of Methodology	1.1-1
1.2 Submitting Change Requests	1.2-1
2.0 Lifecycle Model	2.0-1
2.1 Project Sizes	2.1-1
2.2 Adapting the Lifecycle	2.2-1
2.3 Development Methodologies	2.3-1
3.0 Planning Stage	3.0-1
3.1 Analyze User Environment	3.1-1
3.2 Define Project Objectives	3.2-1
3.3 Define Project Scope	3.3-1
3.4 Develop High-Level Project Requirements	3.4-1
3.5 Establish Communication With Functional Areas	3.5-1
3.6 Determine Project Feasibility	3.6-1
3.6.1 Investigate Software Alternatives	3.6-3
3.6.2 Investigate Hardware Alternatives	3.6-7
3.6.3 Formulate Platform Options	3.6-8
3.6.4 Conduct Feasibility Review	3.6-9
3.6.5 Conduct Analysis of Benefits and Costs	3.6-11
3.6.6 Conduct Feasibility Study	3.6-12
3.6.6.1 Analyze the alternatives	3.6-14
3.6.6.2 Determine feasibility recommendations	3.6-15
3.6.6.3 Develop feasibility study document	3.6-16
3.7 Develop Project Plan	3.7-1
3.8 Develop Software Quality Assurance Plan	3.8-1
3.9 Conduct In-Stage Assessment	3.9-1
3.10 Conduct Planning Stage Exit	3.10-1
4.0 Requirements Definition Stage	4.0-1
4.1 Develop Software Configuration Management Plan	4.1-1
4.2 Select Requirements Analysis Methodology	4.2-1
4.3 Define Project Requirements	4.3-1
4.3.1 Define Functional Requirements	4.3-3
4.3.2 Define Input and Output Requirements	4.3-5
4.3.3 Define Performance Requirements	4.3-6
4.3.4 Define User Interface Requirements	4.3-7
4.3.5 Define System Interface Requirements	4.3-8
4.3.6 Define Communication Requirements	4.3-9
4.3.7 Define Computer Security and Access Requirements	4.3-10
4.3.8 Define Backup and Recovery Requirements	4.3-13
4.3.9 Define Data Requirements	4.3-16

Chapter	Page
4.3.10	Define Implementation Requirements 4.3-17
4.4	Compile Project Requirements 4.4-1
4.4.1	Develop Requirements Traceability Matrix 4.4-2
4.4.2	Develop Software Requirements Specification 4.4-4
4.5	Establish Functional Baseline 4.5-1
4.6	Develop Project Test Plan 4.6-1
4.6.1	Identify Test Methodologies 4.6-3
4.6.2	Identify Test Phases 4.6-4
4.6.3	Identify Test Environment Requirements 4.6-6
4.7	Develop Acceptance Test Plan 4.7-1
4.8	Select Design Methodology 4.8-1
4.9	Revise Project Plan 4.9-1
4.10	Conduct In-Stage Assessment 4.10-1
4.11	Conduct Requirements Definition Stage Exit 4.11-1
5.0	Functional Design Stage 5.0-1
5.1	Determine Software Structure 5.1-1
5.1.1	Identify Design Entities 5.1-2
5.1.2	Identify Design Dependencies 5.1-4
5.2	Design Content of System Inputs and Outputs 5.2-1
5.3	Design User Interface 5.3-1
5.3.1	Design Menu Hierarchy 5.3-3
5.3.2	Design Data Entry Screens 5.3-5
5.3.3	Design Display Screens 5.3-6
5.3.4	Design Online Help 5.3-8
5.3.5	Design System Messages 5.3-10
5.4	Design System Interfaces 5.4-1
5.5	Design System Security Controls 5.5-1
5.6	Build Logical Model 5.6-1
5.7	Build Data Model 5.7-1
5.8	Develop Functional Design 5.8-1
5.8.1	Develop Functional Design Document 5.8-2
5.8.2	Conduct Functional Design Review 5.8-3
5.9	Initiate Procurement of Hardware and Software 5.9-1
5.10	Revise Project Plan 5.10-1
5.11	Conduct In-Stage Assessment 5.11-1
5.12	Conduct Functional Design Stage Exit 5.12-1

Chapter	Page
6.0 System Design Stage	6.0-1
6.1 Select System Architecture	6.1-1
6.1.1 Evaluate System Architecture Alternatives	6.1-2
6.1.2 Recommend System Architecture	6.1-5
6.2 Design Specifications for Software Modules	6.2-1
6.3 Design Physical Model and Data Base Structure	6.3-1
6.4 Develop Integration Test Plan	6.4-1
6.5 Develop System Test Plan	6.5-1
6.6 Develop Conversion Plan	6.6-1
6.7 Develop System Design	6.7-1
6.7.1 Develop System Design Document	6.7-2
6.7.2 Conduct Critical Design Review	6.7-3
6.8 Develop Program Specifications	6.8-1
6.9 Define Programming Standards	6.9-1
6.10 Revise Project Plan	6.10-1
6.11 Conduct In-Stage Assessment	6.11-1
6.12 Conduct System Design Stage Exit	6.12-1
7.0 Programming Stage	7.0-1
7.1 Develop Acquisition Plan	7.1-1
7.2 Develop Installation Plan	7.2-1
7.3 Establish Programming Environment	7.3.1
7.4 Write Programs	7.4-1
7.5 Conduct Unit Testing	7.5-1
7.6 Establish Development Baselines	7.6-1
7.7 Plan Transition to Operational Status	7.7-1
7.8 Generate Operating Documentation	7.8-1
7.8.1 Produce Users Manual	7.8-3
7.8.2 Produce Programmers Reference Manual	7.8-5
7.9 Develop Training Program	7.9-1
7.10 Revise Project Plan	7.10-1
7.11 Conduct In-Stage Assessment	7.11-1
7.12 Conduct Programming Stage Exit	7.12-1
8.0 Software Integration and Testing Stage	8.0-1
8.1 Conduct Integration Testing	8.1-1
8.2 Conduct System Testing	8.2-1
8.3 Initiate Acceptance Process	8.3-1
8.4 Conduct Acceptance Test Team Training	8.4-1
8.5 Revise Project Plan	8.5-1
8.6 Conduct In-Stage Assessment	8.6-1
8.7 Conduct Software Integration and Testing Stage Exit	8.7-1
9.0 Installation and Acceptance Stage	9.0-1

Chapter		Page
9.1	Perform Installation Activities	9.1-1
9.2	Conduct Installation Tests	9.2-1
9.3	Conduct User Training	9.3-1
9.4	Conduct Acceptance Test	9.4-1
9.5	Conclude Acceptance Process	9.5-1
9.6	Transition to Operational Status	9.6-1
9.7	Revise Project Plan	9.7-1
9.8	Conduct In-Stage Assessment	9.8-1
9.9	Conduct Installation and Acceptance Stage Exit	9.9-1
10.0	Software Maintenance	10.0-1
10.1	Problem/Modification Identification Stage	10.1-1
10.2	Analysis Stage	10.2-1
10.3	Design Stage	10.3-1
10.4	Programming Stage	10.4-1
10.5	System Test Stage	10.5-1
10.6	Acceptance Stage	10.6-1
10.7	Delivery Stage	10.7-1
 Appendix		
A	Glossary	
B	List of Abbreviations	
C	Conducting Structured Walkthroughs	
D	In-Stage Assessment Process Guide	
E	Stage Exit Process Guide	
F	Computer System Retirement Guidelines	
G	Site-Specific Information Systems Environment	