



NUI MAYNOOTH

Ollscoil na hÉireann Má Nuad

National University of Ireland, Maynooth
MAYNOOTH, CO. KILDARE, IRELAND.

DEPARTMENT OF COMPUTER SCIENCE
TECHNICAL REPORT SERIES

NUIM-CS-TR2003-09

Documents for Student Projects in Software Engineering

Declan Delaney and Stephen Brown

DOCUMENTS FOR STUDENT PROJECTS IN SOFTWARE ENGINEERING

Declan Delaney and Stephen Brown
Department of Computer Science,
National University of Ireland, Maynooth

Date: 19th May 2003

Technical Report: NUIM-CS-TR2003-09

1. INTRODUCTION

This technical report outlines a set of software development documents, tailored for use by students in software engineering projects. It is based on IEEE standards and, in general, IEEE terms and abbreviations have been used to provide exposure to professional terminology. This is a revised version of the document templates and guidelines described in technical reports NUIM-CS-TR2002-5 and NUIM-CS-TR2002-6. In this report the contents have been consolidated and the guidelines condensed to reflect experience with the previously defined document templates. The titles of each document are now preceded by the word *educational* to distinguish them from the actual IEEE documents from which the document titles are derived.

2. HOW TO USE THE DOCUMENT SET

These templates are not forms to be filled in, but rather a framework for reporting the results of activities carried out during a software development project. The content of these documents is what is important. Each project has its own emphasis and the templates may be changed to match.

3. OVERVIEW OF THE DOCUMENTATION SET

Document	Description	Activities [1]
Educational Software Project Management Plan (ESPMP)	Details the tasks, deliverables, and timetable for the project.	System requirement analysis Software requirement analysis
Educational Software Requirements Specifications (ESRS)	Documents the requirements elicited from the user, and specifies the required software functionality.	Process implementation
Educational Software Design Description (ESDD)	Documents how the software is designed to meet the requirements.	System architectural design Software architectural design Software detailed design
Educational Software Test Documentation (ESTD)	Documents the test approach, test cases, test data, and test results for the verification and validation of the software.	Software qualification testing System qualification testing

4. COMMON SECTIONS FOR EACH DOCUMENT

i. Cover page

- Name of Document
- Project Title
- Document Version Number
- Printing Date
- Location of electronic version of file
- Department & University

ii. Revisions page

Update for each revision to show:

- Version number
- Author
- Description of version
- Data completed

iii. Additional Material

- Additional Issues
- Definitions, Acronyms, and Abbreviations
- References
- Appendices

5. EDUCATIONAL SOFTWARE REQUIREMENTS SPECIFICATIONS (ESRS)

The ESRS defines the problem in the form of requirements – it should not give design solutions – it is the “what” and not the “how” of the project. This document is based on IEEE standard 830[2].

USER REQUIREMENTS

This is a numbered list of the user requirements: containing both the original statement of requirements, and additional requirements elicited from the user.

SOFTWARE PRODUCT SPECIFICATION

Software Product Functionality

This is a numbered list specifying the details of everything that the software will do. It documents the results of your requirements analysis.

User Interface

This specifies what the user interface must do (though not how to do it). This will include such details as: user-interface tasks, processing requirements, etc.

Interfaces to External Hardware and Software

This specifies the interfaces to any external hardware or software that your program must connect to.

Software System Attributes

This specifies the quality of the software product: security, reliability, maintainability, portability, performance, etc

Datastore Requirements

This specifies what data must be placed in persistent store, and the structure/inter-relationships that the stores must support.

6. EDUCATIONAL SOFTWARE PROJECT MANAGEMENT PLAN (ESPMP)

The ESPMP specifies:

- the objective of the project,
- the tasks required to complete the project,
- the project deliverables,
- the project timetable.

This document is based on IEEE standards: IEEE-1058[3], IEEE-1540[4].

PART 1: PROJECT PLAN

Gantt Chart

This shows all the anticipated tasks in the project and their expected start/completion dates. This reflects your planned software process!

Task Descriptions

These are brief descriptions of the tasks identified in the Gantt chart - unless the task name is self-explanatory.

Deliverables and Milestones

This is a numbered list of the project deliverables and milestones

Project Risks and Contingencies

A list to describe what can go wrong with the project and what you would do to cope.

Special Resources Needed for the Project

A list of any special equipment or software you need for the project.

PART 2: PROJECT REPORT

Gantt Chart

This shows all the tasks actually carried out in the project and their actual start/completion dates.

How the software was developed

A description of how you developed the software (software process), including strengths and weaknesses in your approach.

Tools and Techniques

Specify the development methodologies, notations, programming languages, techniques, and tools you used during the project.

7. EDUCATIONAL SOFTWARE DESIGN DESCRIPTION (ESDD)

The ESDD shows how your software will be structured to satisfy the requirements. It provides sufficient technical detail to write and test the code. The SDD is a model of the software system. It represents a partitioning of the system into components, describing their important properties and the relationships between them. You must define what you mean by a "Component". It may be one or more of: an object, a function, a sub-system, a library, etc. This document is based on IEEE standard 1016[5].

Requirements Analysis - Results

This section summarises the results of requirements analysis.

System Architectural Design

This describes the system architectural design, identifying the major component groupings and their interfaces. It shows how they work together to satisfy the requirements.

Detailed Description Of Components

This is a numbered list specifying the name and content of each design component. Some examples of this are:

- *for a class, specify name, attributes, methods and relationships to other classes;*
- *for a function, specify name, parameters, return values and what it does.*

User Interface Design

This describes the design of the user interface in detail. Show the design of layout and menus for each screen. Identify all the items on each screen, and define the actions to be taken for each event.

Requirements Traceability Matrix

This is a matrix showing the mapping between software product functionality (from the SRS) and the design components (i.e. the 'X's identify which design components support each required item of functionality, and vice-versa).

	Component	Component	Component	Component	Component	Component	Component	Component	Component	Component	Componentetc
Functional Requirement	X						X		X		X	
Functional Requirement		X	X						X			
Functional Requirement			X	X				X	X	X	X	
etc.....					X	X	X					

Implementation

This Section describes the mapping of design components to directories/files.

8. EDUCATIONAL SOFTWARE TEST DOCUMENT (ESTD)

The ESTD documents all the software testing. It is based on IEEE standards 829[6], 1008[7], 1012[8] and 1012a[9]. This document covers unit testing (the verification of individual sub-systems or components of the system against their specifications), integration testing, (the testing of inter-operating sub-systems or components against their specifications) and system testing (both verification against the system specification, and validation against the user requirements).

Test Summary

This shows the number of tests passed, the number of tests failed, and the overall test result.

Test Cases, Data, & Results

This is a numbered list of tests. Use tables to group similar tests. For each test specify:

- *Its name*
- *Additional description if test name is not descriptive enough*
- *The input data*
- *The expected output data*
- *The actual output data*
- *Result (pass or fail)*

Post Project Test Approach Analysis

This is a description of how you tested the software. It includes any tools you employed for automated testing, test scripts, the testing techniques employed, and a discussion on the success or failure of your approach.

Test Logs

Copies of any actual output files produced during testing.

9. REFERENCES

- [1] IEEE 12207.2-1997 Industry Implementation of International Standard ISO/IEC 12207: 1995 (ISO/IEC 12207) Standard for Information Technology - Software Life Cycle Processes - Implementation Considerations
- [2] IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications
- [3] IEEE Std 1058-1998 IEEE Standard for Software Project Management Plans
- [4] IEEE Std 1540-2001 IEEE Standard for Software Life Cycle Processes - Risk Management
- [5] IEEE Std. 1016-1998 IEEE Recommended Practice for Software Design Descriptions
- [6] IEEE Std. 829-1998 IEEE Standard for Software Test Documentation
- [7] IEEE Std. 1008-1997 IEEE Standard for Software Unit Testing
- [8] IEEE Std. 1012-1998 IEEE Standard for Software Verification and Validation
- [9] IEEE Std. 1012a-1998 IEEE Standard for Software Verification and Validation -Supplement to 1012-1998 - Context Map to IEEE 12207.1