

# An Alternative Process Documentation for Data Warehouse Projects

Jyothi Prasad K S S<sup>1</sup>, Smt. G Hima Bindu<sup>1</sup>, Smt. G Lakshmeeswari<sup>1</sup>

<sup>1</sup>Department of Computer Science  
GITAM University, Visakhapatnam, Andhra Pradesh, India.

## ABSTRACT

It is a well-known fact that software documentation is, in practice, poor, incomplete and flexible. Projects may wish to add, change, remove or ignore any part of any document. Some may also believe that aspects of one document would sit better in another. If this is the case then users of this document and these templates are encouraged to change them to fit their needs.

The paper describes a process for the documentation that describes the template in data warehouse projects. We focus our attention to develop a series of guides and checklists. This ensures that small teams of relatively skilled resources developing the system can cover all aspects of the project whilst being free to deal with the specific issues of their environment to deliver exceptional solutions, rather than a rigid methodology that ensures that large teams of relatively unskilled staff can meet a minimum standard.

**Keywords:** *Software Documentation, Template, Document Standards.*

## 1. INTRODUCTION

Software engineering projects, as defined by the IEEE/EIA, consist of a number of development activities [1]. Each activity is characterized by a set of deliverables, normally in the form of code or documentation. Providing a structured template for software documentation assists the software engineering project. These templates provide a guide to the expected format and content of the documentation deliverables based on international standards. They also provide a framework for the evaluation of the project.

For large software projects, it is usually the case that documentation starts being generated well before the development process begins. For some types of systems, a comprehensive requirements document may be produced which defines the features required and expected behavior of the system. During the development process itself, all sorts of different documents may be produced – project plans, design specifications, test plans etc.

It is not possible to define a specific document set that is required – this depends on the contract with the client for the system, the type of system being developed and its expected lifetime, the culture and size of the company developing the system and the development schedule that it expected [2].

However, we can generally say that the documentation produced falls into two classes:

### 1. *Process documentation:*

These documents record the process of development and maintenance. Plans, schedules, process quality documents and organizational and project standards are process documentation.

### 2. *Product documentation:*

This documentation describes the product that is being developed. System documentation describes the product from the point of view of the engineers developing and maintaining the system; user documentation provides a product description that is oriented towards system users.

Process documentation is produced so that the development of the system can be managed. Product documentation is used after the system is operational but is also essential for management of the system development. The creation of a document, such as a system specification, may represent an important milestone in the software development process.

Further more documentation standards act as a basis for document quality assurance[3]. Documents produced according to appropriate standards have a consistent appearance, structure and quality. The standards that may be used in the documentation process are:

#### a. *Process standards*

These standards define the process which should be followed for high-quality document production.

#### b. *Product standards*

These are standards which govern the documents themselves.

#### c. *Interchange standards*

It is increasingly important to exchange copies of documents via electronic mail and to store documents in databases. Interchange standards

ensure that all electronic copies of documents are compatible.

Standards are, by their nature, designed to cover all cases and, consequently, can sometimes seem unnecessarily restrictive. It is therefore important that, for each project, the appropriate standards are chosen and modified to suit that particular project. Small projects developing systems with a relatively short expected lifetime need different standards from large software projects where the software may have to be maintained for 10 or more years.

This paper has looked at a consistent set of documents developed at **process documentation phase** that reflects a desire to develop the right amount of documentation at the right time in the project lifecycle and stored in the right place. It is essential to the success of a data warehouse project that a culture of open access is fostered and that the documentation is seen as the entry point to the data warehouse projects.

Here we have identified three aspects to essential documentation:

- A roadmap that describes what documentation is required and how it fits together.
- Team members within the project to use the templates, create quality documents and store them to the project repositories.
- Easy access for people outside the project team to the documentation including publication or notification of changes, updates and new releases.

## 2. Process documentation

Effective management requires the process being managed to be visible. Because software is intangible and the software process involves apparently similar cognitive tasks rather than obviously different physical tasks, the only way this visibility can be achieved is through the use of process documentation.

Here the process documentation in data warehousing project proposes a six phase approach that maintains focus on the critical success factors along the development path:

- Committed user and technical staff involvement from the beginning
- Clear definition of scope to prevent paralyzing scope-creep,
- Early executive review and buy-in to ensure priorities are met,
- Careful attention to configuring a platform that will enable rapid response time to queries,

- Intense scrutiny of the data loading and cleansing process to ensure data integrity from source to data warehouse, and
- Documentation and training of technical and production staff and end users to guarantee active use, refinement, and custodianship of the data warehouse.

### 2.1 Data Warehouse Development approach

Data Warehouse Development in 6 phases		
Phase	Tasks	Results
1	Workshops & develop Prototype data warehouse	Data Warehouse Prototype
2	Procure the data warehouse Equipment & consulting Services	Hardware, Software and Implementation Plan
3	Develop the data Warehouse software & convert the Initial Data	Operational software, Initial Queries, Reports & Data
4	Install the Data Warehouse Hardware, Software & Converted Data	Data Warehouse Goes live
5	Train the Data Warehouse Users & Operational Staff	Software Docs and User Manuals
6	Refine the Data Warehouse Data, Queries & Report	Revised Queries & Reports

As per the phases mentioned in the above development approach we define a model for process documentation in below figure (Fig1 Process of Software Documentation). Here phases 2 and 3 define the SDD ,phases 4 and 5 define the technical document and further more phases 5 and 6 defines the testing document and the product document .

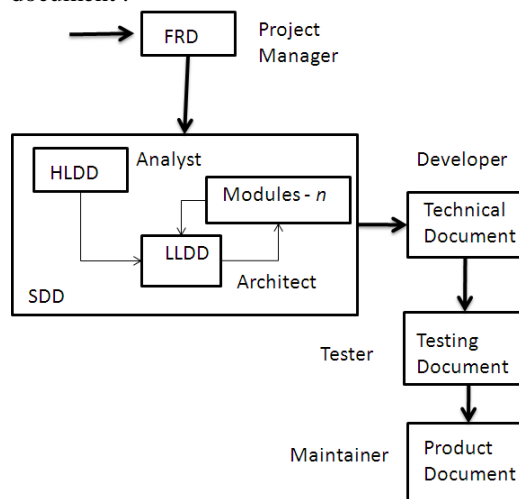


Fig 1: Process of Software Documentation

The process starts with collecting FRD (Functional Requirement Document) which contains Business Requirements and System Requirement This collection takes place at client location. These are submitted to the company or organization which should produce end product and this is handled by the Project Manger (PM)

The documentation contains the basic need of project i.e. scope, out of scope, the purpose, resource, risk factor and other considerations of project with cover page attached which contains project title, project code number, Team Members (Roles, experience), deadline, cost and etc. This FRD forms the input for next phase of SDLC for Analysis and Design phase. The analyst converts the FRD into Software Description Document which is turn divided into two ways[5].

First is the High Level design Document (HLDD), this document is produced by the analyst after a long instant of the analysis of the FRD.

Second is the Low Level Design Document (LLDD) developed by the architect who contains the information of the modules that exist in the project. Here the documentation contains the templates that represent the information of module-*n* with module identity, name of module, status of module etc.

Further more the above information about the Modules helps in easy development (coding) of project by the developer. The developer starts coding based on these modules. After the completion of coding the unit testing is performed by the developer and in resultant produces a Technical document which contains start and ending time schedules of module tested and the testing report with input provided and expected outputs resulted etc.

This Technical document serves as basis input for the testing document phase and makes tester task easy by not dealing with unnecessary errors i.e. while developing the project by the help of tools, there are some packages that show errors or upload errors at some constraints. The tester will skip such errors and these error are supported neither with hardware or software

Finally the Tester produces Testing document by performing various forms of testing the following various types testing carried out in testing environment.

- 1) Development Integration Testing
- 2) System Integration Testing
- 3) User Acceptance Testing

Further if there exists inconsistency with in the testing report, retesting take place for error free project satisfying all the needs in the Software Requirement Specification. This testing document

produced is finally submitted to PM. Based on the testing report the maintainer start producing product documentation which is used by end user.

\*Note this paper provides specific assessment criteria: it describes the development process of software documentation and it does not cover the product documentation which contains the following documentation (user manual, reference manual, installation manual etc.

### 3. Procedure of process documentation in Data ware house projects:

This process of documentation is produced at the end of phases. Now let us examine how this process is used in Data Warehouse (DWH) project. As the client approach the organization or company in bidding process, the Project manager (PM) of that selected organization is solely responsible for returning error free, quality product to client. DWH project neither be an existing or new one. The client do explain the requirements that gives a projection of the aspects of the project Based on these requirements FRD is produced by the PM which leads the process into the next phase of process documentation i.e. SDD. Here with in the SDD phase firstly, the DWH project deals with databases design, data sources design, data marts design, which is developed with the analysis of the design models such as E-R diagrams, dataflow diagrams, use cases etc.

Next the Low Level Design Document (LLDD) developed by the architect which contains the information of the modules that exist in the project which further helps the Based on design models the architect divided into set modules and developers for writing the code easily with the help of respective

To avoid risks in developing, the meetings are conducted once in a week to share Knowledge Transfer. This meeting follows a mesh wise procedure i.e. one of the team member starts the meeting by questioning or explaining the risks aspects and in turn any one of the team member can give solution or suggestions for that aspects. Concluded multiple views can be shared.

Knowledge Transfer and Testing are the main documents to be produced error free [4][6].

Each document phase follows a template basing upon the requirement here basing upon the

requirement some of the information within the templates can be ignored.

#### 4.TEMPLATES

**COVER PAGE (contents & Layout)**

**Name of Document**

**Project Title**

**Document Version Number**

**Printing Date**

**Location of Electronic version of file**

**Domain**

Fig 2: Template 1

**REVISION PAGE (Contests)**

Overview  
 Target Audience  
 Project Team Members  
 Version Control History:

Version	Primary Author(s)	Description of Version	Date Completed
Draft/final			

Signatures of Approval

Fig 3: Template 2

**FUNCTIONAL REQUIREMENT DOCUMENTATION (FRD)**

Cover page  
 Revision page  
 Table of contents

1. Introduction
  - 1.1 Purpose
  - 1.2 Scope
  - 1.3 Out of Scope
  - 1.4 Reference
  - 1.5 Assumptions and constraints
2. Project Manager and Methodology Selection
3. Functional Requirements
  - 3.1 Context
  - 3.2 User Requirements
  - 3.3 Data Flow Diagrams
  - 3.4 Logical Data Model /Data Dictionary
4. Other Requirements
  - 4.1 Interface Requirements
  - 4.2 Data Conversion Requirements
  - 4.3 Hardware/software Requirements
  - 4.4 Operational Requirements

Fig 4: Template 2

**HIGH LEVEL DESIGN DOCUMENT (HLDD)**

Cover page  
 Revision Page

1. Introduction
  - 1.1 Scope
  - 1.2 Assumptions
2. Database Architecture
  - 2.1 Metadata Design
  - 2.2 Fundamental Entities Design
  - 2.3 Transaction Entities Design
  - 2.4 Data marts Design
  - 2.5 Classification model design
  - 2.6 Transformation from logical to physical design
3. Mapping to the Data Warehouse Model
  - 3.1 Mapping requirements to DW model
  - 3.2 Mapping Fundamental Entities
  - 3.3 Mapping of Transaction Entities
  - 3.4 Mapping of classifications
4. Design Tools

Fig 5: Template 3

**LOW LEVEL DESIGN DOCUMENT (LLDD)**

Cover page  
 Revision page

1. Environment
2. Tools
3. Construction process
  - 3.1 Logic Design (Coding)
  - 3.2 Unit Testing and report
4. Meeting / Review (Internal)  
 Knowledge Transfer
5. Review Report

Fig 6: Template 4

**MODULE - n**

Cover Page

1. Module -n
  - 1.1 Module id
  - 1.2 Module Name
  - 1.3 Module Status
  - 1.4 Module Error
2. Report

Fig 7: Template 5

TEST DOCUMENT (TD)	
Cover page	
Revision page	
1. Introduction	
1.1 Scope	
1.2 Quality objective	
1.3 Roles and Responsibilities	
1.4 Assumptions for Test Execution	
1.5 Constraints for Test Execution	
2. Test Methodology	
2.1 Purpose	
2.2 Test Levels	
2.3 Data quality and accuracy Testing	
2.4 Test Completeness	
3. Test Deliverables	
3.1 Document	
3.1.1 Test Approach Document	
3.1.2 Test Plan	
3.1.3 Test Schedule	
3.1.4 Test Specifications	
3.2 Defect Tracking & Debugging	
3.3 Report	
4. Resource & Environment needs	
5. Terms/ Acronyms	

Fig 8: Template 5

that there is a lack of documentation, but just a lack of the right documentation in the right place. It is the quality and availability of the documentation that leads to an understanding of what is available and hence to the value and reputation of the data warehouse itself.

**7. References**

[1]. Garg, P. K. and Scacchi, W. 1990. ‘A hypertext system to maintain software life-cycle document’ IEEE Software, 7 (3), 90–8.

[2]. IEEE, 1987. IEEE Standard for Software User Documentation, IEEE-Std1063-1987

[3]. Chapter 30 from book Software Engineering, 4<sup>th</sup> edition, published by Addison Wesley in 1992.

[4]. IEEE Std. 829-1998 IEEE Standard for Software Test Documentation

[5]. IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications

[6]. IEEE Std. 1008-1997 IEEE Standard for Software Unit Testing

**5. TABLES**

Knowledge Transfer :						
Project Id:						
Project name:						
S.No	Module ID	Member name	Role	Risk Found	Suggestions/idea	Remarks

Table1: Knowledge Transfer

Module Report :						
Project Id:						
Project name:						
S.No	Module ID	Member name	Status	Start Date	End Date	Remarks

Table 2: Module Report

Testing Report :						
Project Id:						
Project name:						
Test Case ID	Test Case	Input	Excepted Output	No. of Error/Defects	Comments	Status

Table 3: Test Report

**6. Conclusion:**

Many data warehousing projects are both long running and poorly documented. It does’nt mean

Jyothi Prasad K S S, student of M.Tech(CST) in the Dept. of Computer Science and Engineering , GIT, GITAM University Visakhapatnam, AP, India.

G.HimaBindu,Asst.Professor, Dept. of Computer Science and Engineering , GIT, GITAM University Visakhapatnam, AP, India. Her research interests include Software Engineering ,image processing, Security and Computer Networks. She has 6 years of teaching experience.

G Lakshmeeswari Asst.Professor, Dept. of Computer Science and Engineering , GIT, GITAM University Visakhapatnam, AP, India. Her research interests include Software Engineering ,image processing, Security and Computer Networks. She has 12 years of experience and is now doing her Ph.D in Steganography.