Toward Single-sourcing of Project Documentation: A Preliminary Study

Chung-Yang Chen and Wen-Lung Tsai*  
Department of Information Management  
National Central University  
Jhong-Li (320), Taiwan

ABSTRACT

Project documentation is a requisite task to the success of a software development project. Project documentation strings up the context of development for a software project. However, preparing, writing and maintaining project documents are often considered a tedious and fragmental job, especially in the rapid project development with the stress of time. With the aim at achieving single sourcing of project documented contents, we present an object-oriented approach that scrutinizes the contents of a software project’s documentation.

Keyword: Project Documentation, Object-oriented, Single sourcing

1. INTRODUCTION

Software project management is a complex process and unfortunately is not always considered synonymous with the word “successful.” The Standish Group (2004) reported that approximately 50% over 9,000 software development projects in their report were classified as challenged, meaning these projects experienced cost and budget overruns or lacked functionality [1].

Project managers often rely on their experience to make decisions to keep the project on schedule and meet the budget, functionality, and quality targets. An experienced project manager is one of the keys to project success [2]. Organizations understand the importance of experience and often hire individuals based on experience rather than academic training and documentarily task [3]. No one can deny that experience doesn’t play an important role in a software development project but project documentation is a task to the success of a software development project. It is capable of assisting project managers in applying lessons learned from the former projects to group communication, receiving new information, storytelling [4], mentoring [5], etc. Project documentation strings up the essential context of development for a software project and enabling this type of content to be reused [6].

During the process of software project development, the documents that produced in every phase are the extension of front phase or the basis of next phase. All of these documents are indispensable foundation to ensure software quality, project efficiency, and future product maintenance. So, most project stakeholders often consult the former document format and then create new documents. But the contexts of the later documents are often accompanied with repeated information and even incompatible and inconsistent to the development project. Thus, this paper aims to achieve single sourcing of project documented contents by object-oriented approach.

2. REVIEW OF SINGLE SOURCING

2.1 What is Single Sourcing?

Traditionally, documentation is written in files that consist of chapters or sections. These files are then assembled to create a document, such as a user guide or reference manual. Single source materials, however, are created from one single location or file. All information types required by a particular project (e.g., product brochures, user guide, training materials, online Help) are created from the single file, hence the name-single sourcing [7].

In single sourcing, rather than assembling files to create documentation, writers break information down to the element level (e.g., section, paragraph, sentence), then identify which element is required for each information type [8]. For example, conceptual information and exercises may be identified as training materials, whereas procedural information may be identified as training, user guide, and online

* Corresponding Author: 974403007@cc.ncu.edu.tw
Help. However, all this information resides in one source.

2.2 Process for single sourcing

Single sourcing involves identifying all information requirements up front, then developing them from a single source. Rockley (2002) and her group (The Rockley Group) [9] defined the process of single sourcing as followings:

1. The Information Audit: The information, audiences, and the processes are examined currently in place to create documentation. That helps them to identify where information can be reused, what the audiences needs are, and how processes need to change to accommodate single sourcing.

2. The Customized Information Blueprint: Base on step 1, a too is selected to support the single sourcing requirements, and design the ways to use the tools and information models. The “big picture” that will guide the rest of the project is designed.

3. Proof of Concept: Working model and prototype are built, then usability test to be ensured by correct design decisions correct and the single sourcing blueprint they developed in step 2.

4. Full Implement: The rest of the information products are developed for the project, followed the information models and the prototype. Methodologies are in place and usability testing has been completed. The rest of documentation is developed by the processes.

5. Knowledge Transfer: The last step, The Rockley Group teaches clients to maintain their single source documentation on their own. They provide methodology, information models, templates, style guides and key information to everyone who will be involved in maintaining the single source file and producing the various outputs.

2.3 Benefits of Single Sourcing

Single sourcing requires more up-front planning, but once implemented it can result in consistent information and significant cost savings. Several function-related projects with information developed from a single source have shown savings of 25-60% [7] [9]. Translation costs are a significant area of savings as “core” information is translated once, then reused everywhere it appears rather than being retranslated in each document. Additionally, it will require fewer people to create the materials since less original material is developed [8].

3. METHODOLOGY

3.1. Elementary concepts of documents and project documentation

The concept “document” shares multiple connotations. The fundamental point is that a document carries information in some format that it could be shared, disseminated, stored and acted upon. Formally, we shall define a document as a set of metadata and data [10]:

\[
\text{Document} = \{\text{Metadata}; \text{Data}\}, \text{ where }
\]

\[
\text{Metadata} = \{\text{Ownership, Authorization, History, Status, Procedures, Links, Data}_\text{description}\}
\]

\[
\text{Data} = \text{any attributes associated in the document.}
\]

According to this definition everything that has been stored into an accessible source is a document, be the source a drawer or database, is a document. Things that are not accessible are not documents, like non-documented discussions in the previous week. By thinking in fundamental terms is everything that is stored in the database, as a file or data entity is in its broadest sense a document.

Typical documents are meeting minutes, drawings, emails, specifications, tendering instructions, test result, etc. In this paper, we shall not go into details of project documentation, i.e. on how project documentation is shared effectively or how changes in them are conveyed throughout the organization. Thus, we also define project documentation as a set of type, view and section:

\[
\text{Project Documentation} = \{\text{Type, View, Section}\}, \text{ where }
\]

\[
\text{Type} = \{\text{Plans, Proposals, Reports, Reviews, Statement, Lists, Minutes, etc. }\}
\]

\[
\text{View} = \{\text{PP, REQM, PMC, SAM, MA, PPQA, CM, etc.}\}
\]

\[
\text{Section} = \text{any attributes associated in the project documentation.}
\]

However, some of the key concepts deserve attention also in an application-oriented paper. What makes documents and project documentation attractive as tools for process management is the active functionality inherent. These characteristics are embedded in the capabilities of metadata or section. Besides the traditional document related metadata (e.g. ownership of the document), authorization (e.g. access and scripting rights) and status data the contemporary information and communication technologies enable automatic/semiautomatic maintenance of document history. What is even more interesting, is the gradually emerging object oriented nature of project documentation, meaning the project documentation
may have embedded functionality in the form of procedures to be activated by predefined triggers during the life cycle of project.

In a software development project, the project documentation and project management are closely connected together. The outcome of the project is typically a set of documents that describe either the function developed, process improvement or software applications to be delivered. If the organization is able to define the documents that form the output and the documents needed in various intermediate steps it can monitor the progress of the project using the project documentation. The document and project milestones are then directly linked and no milestone can be approved before all documents needed have been approved. The approach does not provide means to measure the quality of the documents and correspondingly the quality of the project. However, the experiences gained in various total quality management efforts provide a reason to assume that a punctual project is likely to produce good quality, i.e. a good process tends to produce good quality.

Documents are related to each other. An apparent feature of software development projects processes is the process flow seconded by the corresponding document flow. The status change of a document may trigger the instantiation of another document. Or a document status change may synchronize the flow of several work procedures, i.e. document flows. This process/document flow management may be enabled via the links or structural references between the documents.

The traditional administrative systems are implemented with the help of database management systems that do have data dictionary functionality to manage every single data item in the database. Project documentation does not have the means to cover and master all the data elements within the documents. Normally, no references to the contents of the documents are needed. However, there are some needs to have associations to a few critical data elements within the key documents reflecting the status of the actual real world business process.

3.2. Key functions of project documentation

There are three aspects of project documentation that prevail around the documents concerned:

1. Document flow: The intended way how activities are performed. These are usually stated in flow diagrams, which define the organizational routines to be performed in order to run the operations and development efforts as unified processes in the enterprise.

2. Document lifecycle: The chain of statuses through which each and every document passes once created. This lifecycle may vary from document to document, but nevertheless, at least document life span is dictated here. This lifecycle can share different status positions, various iterations and change processes. The document itself and its revisions and different versions follow physically the predefined sequence. Examples of documents with high relevance of document life cycle are meeting minutes, UML drawings and sales forecasts.

3. Document implementation: The means how the document is transferred, manipulated, and controlled. Document implementation comprises the tools and information networks. Examples of activities related to this layer are posting and mailing, all the software used to generate and share documents, and the direct transfer between people in the organization.

In brief, the set of aspects of project documentation above emphasizes the use of the documents, the documents themselves and all the tools and media used to share them. This three-aspect model follows also the various network protocols and their layer definitions in order to establish a connection through the network.

3.3 Object Oriented Approach

A major benefit of adopting the object-oriented paradigm in a software development project is that of real reusability: reusability of code, reusability of program portions, reusability of analyses and designs. To date, researchers and practitioners focus their efforts on reuse of the standard and general components or structures. Object-oriented reuse has been discussed in terms of the building of libraries of standard object classes, as well as cataloging of the design patterns [11] and frameworks [12].

During developing applications using object-oriented approach, we realize that the documents of a software development project, especially that written by reuse experts, are very useful for developing new applications in the same application domain. We can utilize the pre-built project documents as followings:

1. Learn by example. The all documents of a project provide a complete instance.

2. Develop a family of software products in the same application domain.

3. Reuse in all stages of the software life cycle, including analysis reuse, which is not considered in the general asset-centered reuse.

4. Get more domain specifics and designs context information which is needed to guide reuse.

The above approaches tend to record down only the general or relatively stable properties of classes. For example, in class library only the inheritance relationship is kept. Context information, which is not used as reusable assets, may help to understand and guide the reuse.
4. RESULT

4.1 Defining software development project documentation

Owing to plenty types of project documentation in a software development project we illustrate one type, plan, as Table 1. The column is process group of IEEE std 1074 (Standard for Developing Life Cycle Processes) [13]. The content of each process group could be consolidated to a view. The row is section covered in IEEE/ EIA 12207.1- 1997 Guide for Information Technology - Software Life Cycle Processes- Life Cycle Data [14].

4.2 The process of using single sourcing

Once we have defined that a single source file will be more beneficial than two separate ones, have estimated the work effort, and established the design of the structure, then the writing starts. Whether we create a single-source file for the first time or updating an existing one, the process is similar to working with a file for one single product. However, there are several important differences, which are evident in Figure 1.

![Figure 1: The process of using single sourcing](image)

For one thing, the file is reviewed once at least, once by each domain expert who is concerned with it. This means more meetings to attend and more perspectives to consider. We must consider the feedback from each reviewer and assess where comments are applicable to shared sections of the view. Any textual or graphics changes that are made to the shared sections of the document must be acceptable to each reviewer, usually meaning that there must be more compromises by everybody.

Table 1: One relation between view and section

<table>
<thead>
<tr>
<th>Section</th>
<th>View Development (Incl. Pre., Post-</th>
<th>Management</th>
<th>Integral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date of issue &amp; Status</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Scope</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Issuing organization</td>
<td>•</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Approval authority</td>
<td>•</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: in this example, degrees of involvement in a full-scale development project are illustrated as: ○ : low involvement(1) ● : medium involvement(2) ◎ : high involvement(3)

4.3 The Implementation

We expect a single sourcing environment to address project documentation. The authoring environment has to be user-friendly and easy to learn. This paper is accomplished a holistic illustration (Left part is view, and right part is section) as Figure 2. It is transparent to understand each part of single sourcing in a software development project. Most importantly, it is able to scale easily, so we could make enhancements and changes without major infrastructure changes going forward.

![Figure 2: A holistic project documentation](image)

We also want an environment that allowed authors and editors to be more productive. Our goals are to promote project efficiency by reducing
maintenance and overhead in the project documentation. We draw a clear distinguish between documentation and stakeholders in a software development project as Figure 3.

5. CONCLUSION.

The single-sourcing is particularly useful for the development of a family of software in the same application domain. Project documentation provides rich context information for understanding. This approach also provides a good basis for learning by example. Supporting by a powerful navigation tool, developers can learn from a well-documented project developed by the expert developer with high reuse ability, to develop his own project efficiently.

At present, a prototype system supporting this approach has been implemented. It includes object-oriented modeling tools, and a database for managing document information. A navigation tool will be implemented in the near future.

REFERENCES


ABOUT THE AUTHORS

Chung-Yang, Chen is an Assistant Professor in the Department of Information at National Central University (NCU), Taiwan R.O.C. He received his Ph.D degree in Industrial Engineering and Engineering Management at Arizona State University (ASU) in 2002. His current research and teaching interests are in the general area of CMMI and Project Management. In particular, he is interested in Product Data Management, Modeling of Meeting Flow Systems, and Software Changing Impact.

Wen-Lung, Tsai is a Ph.D graduate student of Information Management at National Central University (NCU). His research interests are Project Management and Product Data Management.
專案文件單一來源化：前導研究

陳仲儼、蔡文隆
國立中央大學資訊管理學系
中壢市中大路300號

摘要

專案文件化 (Project Documentation) 在軟體發展專案是不可缺少的工作任務。專案文件化具有串起整個軟體發展專案的脈絡的重要功能。然而，對於文件化的工作例如準備、撰寫及維護文件資料而言，常被視為一項繁重的工作，尤其是在有時間壓力下的快速發展的專案。本文研究目的是在於探索專案文件內容予以單一來源化 (single sourcing) 之可能性，並嘗試以物件導向 (object-oriented) 的方法詳究軟體開發專案的內容。本文為作者在專案內容管理全程研究中的一部分，整體目標期能協助專案更有效率地進行專案文件化的工作。

關鍵詞：Project Documentation, Object-oriented, Single sourcing
(*聯絡人：974403007@cc.ncu.edu.tw)