Document Concordance Generator

Software Project Plan

Eric Brickner
Chris Blanchard

20-260-495
Software Engineering Lab
Professor Carter
Winter, 1992

1. Introduction

The software project management process begins with a set of activities that are collectively called project planning. This document contains the logical framework, i.e. the project plan for the future development of the Document Concordance Generator. A description of all project objectives, estimates, risks, resources, staff organization, and schedule is contained within this document. Documents such as the Project Requirement Specification, Design Specification, Code, and Test Plan are to be developed according to the schedule presented here.

1.1. Scope and Purpose of Document

This document outlines the necessary time and cost estimates, risks and risk abatement, and resources required to carry out the development of the Document Concordance Generator. This document is primarily for use by the development group, and may be used by the project sponsor as information relating to the management and resource estimates of the project.

1.2. Project Objectives

1.2.1. Objectives

A.) Intended Use - The Document Concordance Generator is intended as an aid in the study of large textual works.

B.) Environmental Constraints - This software is being developed for use on the Sun SparcStation.

C.) Lifetime of the Software Product - The presumed life of the Document Concordance Generator is limited to the length of the Software Engineering Laboratory, which ends March, 1992.

1.2.2. Major Functions

Concordance - an alphabetical index that shows the locations within a document where each word may be found.

Given the name of a file containing a document of an output file, the Document Concordance Generator produces a concordance of the document in the output file. The concordance output file contains a heading at the top of each page, a footer at the bottom of each page with page number, and a title page showing the name of the input file. Provision is made for the exclusion of certain words from the generated concordance.

1.2.3. Performance Issues

No performance constraints have been specified by the customer, therefore, speed/performance is not a major concern of the group. Real- time performance is not an issue in this project.

The group, however, has developed certain internal speed/performance constraints:

A.) Performance constraint - The Document Concordance generator is to perform its function in an error-free manner.

B.) Speed constraint - The run-time of the Document Concordance Generator is to be directly proportional to the length of the input document.

1.2.4 Management and Technical Constraints

No management constraints have been identified to date. Technical constraints include the following:

A.) Finite quantity of secondary storage for input/output files.
B.) Finite quantity of RAM that may be incorporated into internal data structures.
C.) Finite quantity of computing time available to group.
D.) Longevity of present SUN workstations is yet unknown.

2. Project Estimates

2.1 Historical Data Used for Estimates

A) Members of the group have experience in the creation of programs involving sophisticated data structures and algorithms.

B.) Members of the group have experience writing Object-Oriented code.

C.) Members of the group have had to learn other computer languages in relatively short time spans before.

2.2. Estimation techniques

The estimation techniques employed by the group for the Document Concordance Generator include: LOC decomposition and estimation, Task decomposition, and the Constructive Cost Model (COCOMO). The organic mode of Constructive Cost Mode (COCOMO) was chosen as its description fits this group's situation quite well. The organic mode is defines as software projects that are relatively small and simple, in which small teams with good application experience work to a set of less than rigid requirements.

2.3. Estimates

LOC decomposition and estimation

Module Type	Optimistic LOC	Most Likely LOC	Pessimistic LOC	Expected LOC
Skip-Words Module	20	25	35	26
Search Module	35	50	75	52
Output File-Generation Module	30	40	65	43
Line Parsing Module	50	75	90	73
Storage Data-Structure Module	25	45	55	43

Therefore, based on LOC decomposition, the total comes to 237 LOC.

Time - Effort Estimates - COCOMO Method

Basic COCOMO - Organic Mode

$E = a_b(KLOC) \exp(b_b)$

$E = 2.4(0.237) \exp (1.05)$

E = 0.55 person-months

$D = c_b(E)\exp(d_b)$
D = 2.5(0.55)exp(0.38)
D = 1.99 chronological months

Therefore, according to the COCOMO method of estimation, this project will require two people to work approximately 48 hours apiece.

Time Estimates- Task Decomposition Method

Task#	Description	Est. Hours
1.0.0.	**Develop Project Plan**	**41.0**
1.1.0.	Attend Initial Meeting	0.5
1.2.1.	Attend Informal Walk Through Meeting	4.0
1.2.2.	Develop Rough Draft	18.0
1.3.1.	Attend Pre-Project Review Meeting	5.0
1.3.2.	Present Project Plan	0.5
1.3.3.	Attend Post-Project Review Meeting	3.0
1.4.1.	Prepare Final Plan	10.0
1.4.2.	Submit Final Plan	0.0
2.0.0.	**Develop Requirement Specs**	**28.5**
2.1.1.	Attend Informal Walk Through Meeting	4.0
2.1.2.	Develop Rough Draft	15.0
2.2.1.	Attend Pre-Requirements Review Meeting	5.0
2.2.2.	Present Requirements Specs	0.5
2.2.3.	Attend Post-Requirements Review Meeting	3.0

2.3.1.	Prepare Final Requirements	1.0
2.3.2.	Submit Final Requirements Specs	0.0
3.0.0.	**Develop Design Specs**	**28.5**
3.1.1.	Attend Informal Walk Through Meeting	4.0
3.1.2.	Develop Rough Draft	15.0
3.2.1.	Attend Pre-Design Review Meeting	5.0
3.2.2.	Present Design Specs	0.5
3.2.3.	Attend Post-Design Review Meeting	3.0
3.3.1.	Prepare Final Design	1.0
3.3.2.	Submit Final Design Specs	0.0
4.0.0.	**Generate Code**	**41.5**
4.1.1.	Attend Informal Walk Through Meeting	2.0
4.1.2.1.	Code Skip-Word Module	2.0
4.1.2.2.	Code Storage Data Structure Module	5.0
4.1.2.3.	Code Line-Parsing Module	4.0
4.1.2.4.	Code Search Module	5.0
4.1.2.5.	Code Output File Format Module	4.0
4.1.3.	Link All Modules	5.0
4.2.1.	Attend Pre-Code Review Meeting	5.0

4.2.2.	Present Code	0.5
4.2.3.	Attend Post-Code Review Meeting	2.0
4.3.0	Prepare Code Documentation	4.0
4.4.1.	Prepare Final Code	3.0
4.4.2.	Submit Final Code and Documentation	0.0
5.0.0.	**Develop Test Plans**	**25.5**
5.1.1.	Attend Test Plan Walkthrough Meeting	4.0
5.1.2.	Develop Test Plan	3.0
5.1.3.	Perform Preliminary Tests	3.0
5.2.1.	Attend Pre-Test Review Meeting	1.0
5.2.2.	Present Test Plan	0.5
5.2.3.	Attend Post-Test Review Meeting	1.0
5.3.0	Perform Additional Tests	3.0
5.4.0	Prepare Final Test Plans	10
5.5.0.	Submit Final Test Plans	0.0
6.0.0.	**Perform Demo**	**2.0**
6.1.0.	Attend Pre-Demo Meeting	1.0
6.2.0.	Consolidate All Reports On Disk	0.5
6.3.0.	Demonstrate Concordance Generator	0.5

Total Time Estimated to Complete Project = 167.0 hrs / person * 2 people = 334.0 hrs total

Cost Estimates - Overview

Based on the fact that the estimate of LOC is rather tentative, the group feels that the COCOMO - based estimates may be questionable. In a similar manner, the Task Decomposition -based estimates are possibly subject to under-estimation, as the group is inexperienced in estimation.
The co-op pay rate of approximately $10.00 per hour is used as a foundation for all calculations.

Cost Estimates - COCOMO

6.2 chron. months * 172 hours / month * $10.00 hour = $10,664.00

Cost Estimates - Task Decomposition

334 hours * $10.00 / hour = $3340.00

Cost Estimates - Summary

The overall cost estimate that the group is using for the Document Concordance Generator is based on a combination of the two figures.

Total Estimated Cost: $ 5000.00

3. Project Risks

3.1. Risk Analysis

3.1.1. Identification

A.) Time limited to one quarter
B.) Lack of Software Engineering experience
C.) Lack of experience with ADA
D.) Other time commitments outside of school
    1.) Interpersonal Relationships
    2.) Job Commitments
    3.) Family Emergencies
E.) Other in-school time commitments
    1.) Other Programs to Write
    2.) Tests in other classes
    3.) Homework
    4.) Reading
F.) Unfamiliar with Interleaf publishing software
G.) Unfamiliar with Primavera planning software
H.) Uncertainties associated with the development environment
    1.) Compiler
    2.) Debugger
    3.) Browser
I.) Uncertain of machine availability
J.) Uncertain of special resources availability
K.) Computer Viruses

3.1.2. Risk Estimation

Risk Type	**Severity**	**Probability**	**Product**

Unable to complete in limited time	.80	.10	.08
Lack of software engineering experience	.40	.25	.10
Lack of experience with ADA	**.75**	**.50**	**.38**
Time commitments outside school	.30	.10	.03
Time commitments inside school	**.80**	**.60**	**.48**
Unfamiliarity with Interleaf	**.75**	**.60**	**.45**
Unfamiliarity with Primavera	.75	.30	.23
Uncertainties associated with development environment	.80	.20	.16
Uncertain of development platform availability	.25	.70	.18
Uncertain of special resource availability	.10	.10	.01
Possibilities of computer virus	.99	.01	.01

3.1.3. Evaluation

After discarding the least significant 80% of all identified risks, the group arrived at three substantial risks:

1.) Time commitments inside school.
2.) Unfamiliarity with Interleaf.
3.) Lack of experience with ADA.

None of these three risks is likely to impede the development of, nor hinder the quality of the Document Concordance Generator.

3.2 Risk Management

3.2.1. Risk Aversion Options

1.) Other time commitments inside school

Option 1 - Carefully monitor time usage on other school assignments.
Option 2 - Drop all other classes.

2.) Unfamiliarity with Interleaf

Option 1 - Schedule additional time to learn Interleaf.
Option 2 - Find another tool to generate documents.

3.) Lack of experience with ADA

Option 1 - Organize a list of questions to present to special resources.

3.2.2. Risk Monitoring Procedures

1.) Other time commitments inside school - Group is unable to completely eliminate this risk, so each member of the group is expected to maintain a schedule of all assignments and due dates. In this way, any potential time conflicts can be avoided or minimized in advance.

2.) Unfamiliarity with Interleaf - Group has decided to completely avoid this risk, by choosing to use the more familiar Microsoft Word for Windows.

3.) Lack of experience with ADA- Group is unable to completely avoid this risk, however, this has been addressed by a plan to generate a list of questions and concerns by the end of each section's preliminary meeting.

4. Schedule

4.1. Project Work Breakdown Structure

Please refer to attached pages at the end of this document.

4.2. Task Network

The Task Network is broken down into approximately one week's work per page. A legend to the Task Network is as follows:

1.) Surrounding each task are four information fields. The uppermost left field contains the earliest starting date for that particular task. The field just below this field contains the latest allowable ending date for that particular task. At the lowermost left corner of each task box, the task reference number appears. These unique numbers correspond to the task decomposition table presented in this document. Finally, at the lower rightmost corner of each task box, there is a field which holds the approximate time to complete each task, as determined by the task decomposition table. The unit of time of this field is hours.

2.) The task boxes that have rounded corners correspond to those tasks which the development group feels are significant task milestones.

4.3. Timeline Chart

For clarity, the development group has broken down the timeline chart into two separate documents. The first, which consists of one page, contains only the significant tasks at the most general level. The second, which consists of three pages, contains each elemental task in the overall project. Please note that the darkened vertical line is a current date line, and is intended as a reference line with regards to current project progress. Also, those tasks which are considered milestones are marked with a diamond-shape to indicate that fact. In addition, at the bottom of each individual timeline, there is a small box which is darkened by the development group to signify completion of that task.

4.4. Resource Table

ID#	Activity	IBM PC	Suns	Mac IIfx	Over-heads	Carter/ Insko	B.Baker	J . Brickner
1.0.0.	**Develop Project Plan**	**x**		**x**	**x**	**x**		**x**
1.1.0.	Attend Initial Meeting	x						
1.2.1.	Attend Project Plan Walkthrough Meeting	x						
1.2.2.	Develop Rough Draft	x						x
1.3.1.	Attend Pre-Project Review Meeting	x			x			
1.3.2.	Present Project Plan				x			
1.3.3.	Attend Post-Project Review Meeting	x						
1.4.1.	Prepare Final Plan	x		x				

1.4.2.	Submit Final Plan						
2.0.0.	**Develop Requirement Specs**	**x**			**x**	**x**	
2.1.1.	Attend Req. Spec. Walkthrough Meeting	x					
2.1.2.	Develop Rough Draft	x					
2.2.1.	Attend Pre-Requirements Review Meeting	x			x		
2.2.2.	Present Requirements Specs				x		
2.2.3.	Attend Post-Requirements Review Meeting	x					
2.3.1.	Prepare Final Requirements Specs.	x					
2.3.2.	Submit Final Requirements Specs.						
3.0.0.	**Develop Design Specs**	**x**			x	x	
3.1.1.	Attend Design Walkthrough Meeting	x					
3.1.2.	Develop Rough Draft	x					
3.2.1.	Attend Pre-Design Review Meeting	x			x		
3.2.2.	Present Design Specs				x		
3.2.3.	Attend Post-Design Review Meeting	x					
3.3.1.	Prepare Final Design	x					
3.3.2.	Submit Final Design Specs						
4.0.0.	**Generate Code**	**x**	**x**		x	x	x
4.1.1.	Attend Code Gen.		x				

	Walkthrough Meeting						
4.1.2.	Develop First Prototype		x			x	x
4.1.3.	Link All Modules	x	x			x	x
4.2.1.	Attend Pre-Code Review Meeting	x	x		x	x	x
4.2.2.	Present Code				x		
4.2.3.	Attend Post-Code Review Meeting	x	x				
4.3.0.	Prepare Code Documentation	x	x		x		
4.4.1.	Prepare Final Code	x	x				
4.4.2.	Submit Final Code						
5.0.0.	**Develop Test Plans**	**x**	**x**		**x**	**x**	**x**
5.1.1.	Attend Test Plan Walkthrough Meeting	x	x			x	x
5.1.2.	Develop Test Plan		x				
5.1.3.	Perform Preliminary Tests	x	x				
5.2.1.	Attend Pre-Test Review Meeting	x	x		x		
5.2.2.	Present Test Plans				x		
5.2.3.	Attend Post-Test Review Meeting	x	x				
5.3.0	Perform Additional Test	x	x				
5.4.0	Prepare Final Test Plans	x					
5.5.0.	Submit Final Test Plans						
6.0.0.	**Perform Demo**	**x**	**x**				

6.1.0.	Attend Pre-Demo Meeting	x					
6.2.0.	Consolidate All Reports On Disk	x	x				
6.3.0.	Demonstrate Concordance Generator		x				

5. Project Resources

5.1. People

Eric W. Brickner - Junior in the Computer Engineering program at U.C. Computer experience includes programming in C, Assembler, and some Pascal; COOP experience mostly in PC support working in MIS department. Management experience includes 4 years active duty in Army and 3.5 years in Ohio National guard as a squad leader. Married 4.5 years to Jennifer who has 9 years experience with computers and is currently a Systems Engineer working for CBIS using CASE Tools.

Chris Blanchard - Junior in the Computer Engineering program at U.C. Computer experience includes programming in Assembler, C, FORTRAN, and Object Pascal. Co-op experience has involved mainly the specification and development of small to medium-sized software projects, and associated documentation.

5.2. Hardware and Software

A.) IBM-compatible Personal Computers equipped with :
1.) Microsoft Windows 3.0 and Word for Windows
2.) 3.5 inch disk storage capability
3.) Laser Printer

B.) SUN SparcStations equipped with:
1.) ADA language compiler and source debugger
2.) 3.5 inch disk storage capability
3.) Laser Printer

C.) Apple Macintosh IIfx equipped with:
1.) Claris' MacProject
2.) Laser Printer

5.3. Special Resources

A.) Overhead Transparencies
B.) Burt Baker - ADA language Consultant
C.) Darren Insko - Software Engineering Lab Teaching Assistant
D.) Professor. Carter - Software Engineering Lab Instructor
E.) Jennifer Brickner - Systems Engineering Consultant

6. Staff Organization

6.1. Team Structure

Team Leader -Eric Brickner

Team Analyst - Chris Blanchard
Test Engineering - Responsibilities are shared among the group members.

6.2. Management Reporting

A.) Each team member is responsible for generating a list of questions to provide to the team leader. The team leader is responsible for seeing that the questions are directed to the appropriate resource.

B.) Team members exchanged phone numbers and class schedules for easy informal contact.

C.) Each team member is responsible for, at the time of code development, generating a list of possible tests to submit to the Test Engineer.

D.) Project sponsor is informed of project progress and milestones accomplished at each review.

E.) Informal contacts with sponsor or team members when questions arise.

F.) At each meeting, the team leader presents milestones recently achieved, and those on the horizon.

G.) At rough draft meetings, the team can decide if the schedule needs adjustment.

7. Tracking and Control Mechanisms

A.) Milestones - A system of milestones corresponding to the achievement of significant accomplishments is currently implemented.

B.) Master checklist of accomplishments - A master checklist of all steps in the overall project is currently implemented. For more detail, refer to Appendix A.

C.) Course syllabus - At all times, the Software Engineering Laboratory course syllabus will be adhered to.

8. Appendices