A STRUCTURE GUIDE FOR SOFTWARE DEVELOPMENT PROJECTS

CHAPTER 1: INTRODUCTION

This section contains a brief generic presentation of the project undertaken, its main phases, as well as an outline of the remaining document.

CHAPTER 2: PROJECT SCOPE

1. *Purpose and Scope of the Project*
    a. Motivation

        The motivation behind the project is described (automation or upgrading of an existing information system, bespoke software, part of an overall hardware system).

    b. Existing System Description (Optional)

        If the software will replace an existing manual or semi-automatic information system, the elements of the existing system need to be described (organization, people, work, environment).

    c. Project scope

        A rough-cut description of the software is presented. Major inputs, processing functionality and outputs are described without regard to implementation detail.

    d. Performance/Behavior issues

        Any special requirements or conditions for performance or behavior are noted here.

    e. Management and technical constraints

        Any special constraints that affect the manner in which the project will be conducted, the technical approach to development, and any concerns that the developer might have regarding any system aspect are noted here. Included here should be any concerns that the developer might have regarding any system aspect

2. *Feasibility Study*
    a. Financial Feasibility

        A cost-benefit analysis of the proposed software implementation is provided.

    b. Operational Feasibility

        This section illustrates how the proposed implementation will satisfy user and/or organizational objectives.

    c. Technical Feasibility

        A preliminary examination of available technology for the implementation of the project is conducted.

d. Schedule Feasibility

This section investigates if the estimated project completion time lies within the available time resources.

CHAPTER 3: ANALYSIS & DESIGN

1. *Requirements Elicitation Methodology*

   The methodology and techniques that were utilized for the elicitation of customer requirements are described in this section (interviews, brainstorming, scenario analysis, prototyping).

2. *Software Requirements Specification*

   a. *Usage Scenarios*

      A number of usage scenarios are provided that describe the typical functionality of the software, as this was understood through the process of requirements elicitation. UML diagrams can be employed to assist the understanding of scenarios.

   b. *Software modeling*

      1. Database modeling

         A description of the database model that will support the implementation of the software is provided. This constitutes the Entity-Relationship (ER) diagram in a pictorial form, together with a textual description of its structure.

      2. Functional modeling

         A modeling tool such as Dataflow diagrams or UML should be used to provide a detailed description of the major software functions. The context diagram and Level-1 DFD diagrams are presented in pictorial form and explained with the help of narrative text. The lower-level DFD diagrams are provided in the relevant Appendix section. In the case of Object-Oriented modeling, a description of the primary classes of the solution should be provided in this section. The reader should be referred to the relative Appendix for a complete and detailed listing. ADG

      3. Behavioral modeling (Optional)

         The time-dependent behavior of the software is presented with the help of a State-Transition Diagram or a similar modeling tool. Both a pictorial representation and a (brief) textual description of the respective diagrams should be provided.

3. *System Architecture*

   A pictorial representation of the software architecture in the form of a structure chart or an equivalent modeling tool is provided. If the software has been developed as a part of a larger system implementation, the architectural diagram of the system should be provided as well.

CHAPTER 4: IMPLEMENTATION & TESTING

1. *Implementation issues*

   The software development environment is described in this section (programming language, programming tools, interface implementation, hardware requirements).

2. *Testing*

   This section describes the effort and the type of tests that were contacted in order to uncover software errors. Specific sections can be written for unit testing, integration testing, validity testing, system testing (if software has been developed as part of a larger system), and acceptance testing (if software has been developed for a customer). Any known bugs or other related issues about the delivered software should also be included in this section.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

Every step of the development methodology should be critically analyzed to deduce conclusions about the processes of customer communication, requirements elicitation, analysis, design, coding and testing. All difficulties that were encountered should be documented, as well as the procedures that were followed to overcome these difficulties. Comparison with alternative development methodologies should be given wherever it is appropriate.

This section should also discuss future recommendations about potential extensions of the developed software. Students, based on their project experience, should suggest enhancements and modifications on the development methodology (too optimistic perhaps!)

APPENDIX A

The Data Dictionary of the implemented software in a compact form

APPENDIX B

The complete set of Data Flow Diagrams or UML diagrams for the implemented software

REFERENCES

References to all book, article, and network resources that were used for the implementation of this project.

ADDITIONAL DOCUMENTS REQUIRED (on the same or separate binding):

1. *Installation document*

   This document provides detailed instructions about the installation of the system. Describes the installation device (floppy disk, CD-ROM), the files on these disks, and the minimal hardware configuration required for the installation of the system. A step-by-step description of the installation process (including screen captures) together with advice on configuring systems' files (where appropriate) should also be provided. The screen captures must depict the details and not the entire screen. Specific elements need to be detailed.

2. *Introductory user manual*

   It provides an informal description of the standard usage of the system. It describes the start-up process and the utilization of the facilities offered. Heavy use of examples (illustrated where possible) is strongly suggested. Basic error-recovery functionality should also be described.