A Practical Approach To Data Management For Minimizing Software Product Delays

Jagrut T. Sharma
Senior Information Systems Officer
Haldia Refinery, Indian Oil Corporation Limited
(CSI Life Member; IEEE Member, CSI-SIGAI Member)
jagrutsharma @ {iocl.co.in, gmail.com}

Abstract

Software development projects are not just about software. Data modeling, collection, verification, validation, formatting, storage and retrieval are important aspects of the software development process. Often, these data-related aspects are ignored or given low-priority. This results in software projects being delayed due to unavailability of crucial production data at the integration and testing phase. This paper describes the software development life cycle of a product, e-Samadhan, which has been developed by me at Indian Oil Corporation Limited. The different phases of the SDLC – requirement analysis, specification analysis, planning, design, implementation, integration and testing – are described. A comparison of the product development at two different locations – Mathura Refinery and Haldia Refinery – is shown. In this paper, I show some important observations related to the software development process of e-Samadhan. Firstly, I show that software reuse in the right manner reduces the time and cost of software development (till implementation phase) to a great extent. Secondly, I show that integration and testing phase rely to a great extent on availability of accurate production data. Ignoring the importance of data at this final phase can lead to annoying software delays. I also give a brief overview of what constitutes a good data model. Finally, I propose a checklist/questionnaire to be followed by software project managers during the various phases of the software development life cycle to ensure that their data modeling and management is done properly in parallel with the software development process. This will help reduce data-related problems and eliminate avoidable software product delays.

Index Terms - data modeling, e-samadhan, software engineering, software reuse

I. INTRODUCTION

Indian Oil is a public sector Fortune 500 company (rank no. 153) in India. Mathura Refinery and Haldia Refinery are two of Indian Oil’s flagship refineries catering to a major portion of petroleum demand in North India and East India respectively. The capacities of the refineries are 8.0 MMTPA and 6.0 MMTPA respectively. Every year, an Annual Maintenance Contract (AMC) for maintenance of computer hardware and network at various units of IOCL is given to a suitable vendor at the corporate level. Depending upon the size of the units, service engineers are posted by the AMC vendor at the sites for taking care of the hardware, software and networking-related problems. The number of assets under maintenance at Mathura and Haldia are approximately 1400 and 1100 respectively. This includes PCs, printers, servers (RISC and CISC), scanners, plotters, network switches and routers. An effective software management system is required at each of the units to ensure proper monitoring of the AMC and efficient user service as per the terms of the contract. In this paper, I give the case study of the AMC management software e-Samadhan, which was developed by my team and me. The software was initially developed for Mathura Refinery in 2002. Thereafter, it was implemented at IOCL Research & Development Center (Faridabad) and Barauni Refinery (Bihar). Currently, the work of implementing it for use at Haldia Refinery (West Bengal) is under progress by my colleagues and me. This paper presents the software development life cycle of e-Samadhan, the challenges faced at various stages, the solutions adopted for various problems and suggestions to make the software development (with emphasis on efficient data modeling and management) more efficient. It is important to note that the development environment is highly heterogeneous. Also, it gives an account of the reuse of the software at another location. A comparative study of the implementations at both locations is done. The study results are interesting. Results show that:

1. Software reuse drastically reduces the time (and cost) till the implementation phase of software development.
2. Integration and testing phase are highly dependent upon the system data requirements and less dependent on software reuse. Data plays an important role in the final phase of software development.

I propose a questionnaire/checklist to be followed during the various software development life-cycle phases to ensure that the data is ready when the final product is completed. This will help in minimizing software product delays due to data-related problems and poor data modeling.

My future efforts will be targeted towards further
improvement of the software development life cycle and reduction of the implementation time at other locations.

II. BACKGROUND OF E-SAMADHAN

The need for an efficient software solution for effective monitoring of AMC was anticipated at Mathura Refinery of Indian Oil Corporation Limited. There was a lot of computer and network hardware under maintenance. Also, after ERP implementation, it was critical that the problems related to these areas were attended quickly and efficiently. Thus, was born the idea of e-Samadhan. It was planned to build a comprehensive software system that would be used for improving the performance of Information Systems dept. The time frame for system development was estimated between 7 and 8 months. After spending considerable time and effort, a system was developed which had the following features:
- Unmanned complaint logging from any point on the network
- Online complaint status monitoring by all concerned
- Better record keeping of AMC assets and network devices
- Generation of complaint history and more efficient control on Information System department services to end users
- Analytical data generation

III. E-SAMADHAN DEVELOPMENT AT MATHURA REFINERY

Table 1 shows the time taken for different phases of development of the software and the number people involved during the different development phases at Mathura. Figure 1 shows this information in a pie chart. Table 2 and Figure 2 depict the figures for Haldia implementation.

Table 1: Time taken for various phases of the system development at Mathura

<table>
<thead>
<tr>
<th>Phase</th>
<th>Days</th>
<th>From</th>
<th>To</th>
<th>% time</th>
<th>People involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement Analysis</td>
<td>16</td>
<td>15-Dec-01</td>
<td>31-Dec-01</td>
<td>6.72</td>
<td>15</td>
</tr>
<tr>
<td>Specification Analysis</td>
<td>14</td>
<td>1-Jan-02</td>
<td>15-Jan-02</td>
<td>5.88</td>
<td>4</td>
</tr>
<tr>
<td>Planning</td>
<td>15</td>
<td>16-Jan-02</td>
<td>31-Jan-02</td>
<td>6.30</td>
<td>4</td>
</tr>
<tr>
<td>Design</td>
<td>42</td>
<td>1-Feb-02</td>
<td>15-Mar-02</td>
<td>17.65</td>
<td>2</td>
</tr>
<tr>
<td>Implementation</td>
<td>91</td>
<td>16-Mar-02</td>
<td>15-Jun-02</td>
<td>38.24</td>
<td>1</td>
</tr>
<tr>
<td>Integration &amp; Testing</td>
<td>60</td>
<td>16-Jun-02</td>
<td>15-Aug-02</td>
<td>25.21</td>
<td>8</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>238</td>
<td><strong>100.00</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IV. E-SAMADHAN DEVELOPMENT AT HALDIA REFINERY

Table 2: Time taken for various phases of the system development at Haldia

<table>
<thead>
<tr>
<th>Phase</th>
<th>Days</th>
<th>From</th>
<th>To</th>
<th>% time</th>
<th>People involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement Analysis</td>
<td>4</td>
<td>1-Feb-07</td>
<td>5-Feb-07</td>
<td>3.48</td>
<td>15</td>
</tr>
<tr>
<td>Specification Analysis</td>
<td>9</td>
<td>6-Feb-07</td>
<td>15-Feb-07</td>
<td>7.83</td>
<td>4</td>
</tr>
<tr>
<td>Planning</td>
<td>5</td>
<td>15-Feb-07</td>
<td>20-Feb-07</td>
<td>4.35</td>
<td>4</td>
</tr>
<tr>
<td>Design</td>
<td>12</td>
<td>21-Feb-07</td>
<td>5-Mar-07</td>
<td>10.43</td>
<td>2</td>
</tr>
<tr>
<td>Implementation</td>
<td>19</td>
<td>6-Mar-07</td>
<td>25-Mar-07</td>
<td>16.52</td>
<td>1</td>
</tr>
<tr>
<td>Integration &amp; Testing (in progress)</td>
<td>66</td>
<td>26-Mar-07</td>
<td>31-May-07</td>
<td>57.39</td>
<td>8</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>115</td>
<td><strong>100.00</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
V. DESCRIPTION OF ACTIVITIES DURING VARIOUS SOFTWARE DEVELOPMENT LIFE CYCLE PHASES

A combination of waterfall model and rapid prototyping model was used for development of the system.

A. Requirement Analysis Phase

The exact requirements were determined during this phase. The desired functionality of the system was examined. Lot of difficulty was encountered since no such system existed at the time. The end users of the system were the company employees, service engineers and the Information Systems (IS) dept. Users had difficulty in conveying their requirements in a systematic way. A variety of requirement analysis techniques were used. A series of open-ended (unstructured) and close-ended (structured) questions were posed before the users. Interviews were conducted for getting domain knowledge from users. A standard questionnaire was prepared and sent to the end users for determining user requirements. Other documents, such as functional organograms and job descriptions, were used to find exactly what is done and how. The terms of the annual maintenance contract were helpful in determining the crucial requirements that the system needed to satisfy. After a certain unanimity was reached among the IS dept, a rapid prototype was developed using Visual Basic 6.0 (Enterprise Edition) as the front-end and Oracle 8i (Enterprise Edition) as the backend database. A rapid prototype is a hastily built software that exhibits the key functionality of the target product. It reflects the functionality that the client sees, such as input screens and reports, but omits “hidden” aspects such as actual database updating. Based on hands-on experience, employees and service engineers told us how the rapid prototype satisfied their needs and identified the areas that needed improvement. We changed the rapid prototype until their requirements were accurately encapsulated in the rapid prototype. The rapid prototype was then used as the basis for drawing up the specifications.

Time requirements for this phase at Mathura and Haldia were 16 and 4 days respectively. The reason for this reduction in time can be attributed to similarity in requirements at both locations due to a common AMC. The system was to cater to similar functional constraints at both places. Certain minor changes in practices being followed were identified in 4 days at Haldia.

B. Specification Analysis Phase

A specification document must satisfy two mutually contradictory requirements:
1) It should be sufficiently non-technical to be clear and intelligible to the client.
2) It must be precise, complete and detailed enough to result in a fault-free product being delivered to the client at the end of the development cycle.

A DFD (Data Flow Diagram) was drawn which depicted a pictorial representation of all aspects of the logical data flow of the system. The DFD was considered as an ideal technique for specification analysis since the system did not have specific any timing constraints. Otherwise an alternate specification technique like Petri Nets would have been suitable. The end users and service engineers approved the DFD.

Time requirements for this phase at Mathura and Haldia were 14 and 9 days respectively. The specification document for both locations had quite some similarity. But the specifics of the practices being followed at Haldia had to be reflected in the DFD. Hence, 9 days were utilized for incorporating the changes in the original DFD. This also indicates that minor changes in requirements require changing the specification document (and hence DFD) in numerous ways to reflect these variations.

C. Planning Phase

During this phase the software project management plan (SPMP) was drawn up. Components of IEEE software project management plan are specified in IEEE 1058.1(1987). It served as a comprehensive blueprint for the remainder of the software development process. The cost and duration of the project was accurately estimated. The bottom-up approach was used to arrive at the figures. In this approach, the entire product was broken into smaller components (modules). Estimates of duration and cost were made for each component separately and then combined to provide an overall figure.

This phase took 15 and 5 days at Mathura and Haldia respectively. During development at first location, the entire plan had to be worked out from scratch. For the latter, the plan was almost similar to the first plan with minor modifications. Hence, the experience gained at the first location drastically reduces the planning time required for the second location. The plans for both locations were duly approved by their respective management officials before proceeding further with system development.

D. Design Phase

The design was done with the objective of decomposing the final product into modules with high cohesion and low coupling. This means that all the modules were inter-related, but each could be developed and modified without affecting the other. The database schema design was done during this stage. The design technique used by us was Data Flow Analysis (DFA). It is an action-oriented design technique whose underlying foundation is stepwise refinement for achieving modules with high cohesion. The input to the
technique is the DFD (which was developed during the specification analysis phase). The flow of the data in the product is considered. The product is transforming input into output using some transformation. We used the DFA because of the following reasons:
1) DFA used DFD as input (which we already had)
2) DFA is appropriate when the design team is familiar with the application area. (We were familiar with the application domain of our product.)
3) There is more possibility of reuse when DFA is used (in comparison with other design techniques like Jackson System Development). We had anticipated the possibility of the system being deployed in future at other locations of IOCL.

The complete product design for Mathura took 42 days to complete. But when the product was redesigned for Haldia, the design time was cut down to just 12 days. Again, this can be attributed to the many design similarities between the two locations, which saved design time for Haldia.

E. Implementation Phase

Implementation is the process of translating the detailed design into code. The programming platforms available for development were Active Server Pages, Visual Basic, Oracle Developer 6i, Java Server Pages and Java. The database systems that were considered for storage were Oracle, Ingres and Access. The cost-benefit analysis was done to compute the monetary cost of implementation of the system on each of these platforms, as well as the monetary benefits (present and future) of using each of these platforms. Finally, Active Server Pages as the front-end and Oracle database as the back-end technology resulted in the largest expected gain, that is, the difference between benefits and estimated costs, and were finalized for implementation. We also did risk-analysis for selection of implementation technologies. For each of the available technologies under consideration, a list of the potential risks was made along with ways of resolving them. The overall risk was smallest in case of Active Server Pages and Oracle. Hence, these were finally chosen as the platform for system implementation. Recommended programming practices were followed to the maximum extent possible during product coding. Throughout the entire coding phase, reusability was kept in mind. Coding was done in such a manner that a majority of the code could be reused in entirety. Only a small portion of the code needed to be changed if the requirements changed. Also, provisions were provided for extending the functionality by additional coding at a later stage, if required.

This was a time-consuming phase. It took 91 days for implementation at Mathura. But once, the basic code was in place, the system could be implemented at Haldia in just 19 days. Hence, we were able to drastically cut down the implementation time at Haldia by software reuse.

F. Integration and Testing Phase

A top-down implementation and integration approach was followed for a majority of this phase. This means that if a module Level1Module called module Level2Module, then Level1Module was implemented and integrated before Level2Module. Parts of the product also followed the bottom-up implementation and integration approach. A couple of modules did not fit properly and had to be re-implemented. Correctness, robustness, performance and documentation of the product were tested thoroughly as a part acceptance testing. Some of the modules had been completely implemented before the integration phase. In these cases, isolation of faults was difficult. Hence, we learnt a lesson that it is best to perform implementation and integration in parallel. It becomes easier to test inter-dependent modules.

This phase taught me a number of lessons as a software engineer. Availability of actual data was crucial for this phase. At Mathura, we had been working on the data preparation from the start of the software project. Hence, during this phase, we had a fully validated database. We also developed some standard tests to check the accuracy of the system. This enabled the system to be comprehensively integrated and tested at Mathura in 60 days. Due to under-estimation of the work required for compilation of valid data, we encountered a difficult phase at Haldia. Due to poor data modeling and management, this phase has been delayed at Haldia and is projected to be completed by 31-May-07. Hence, it is worth pondering that this phase took 25% of the product development time at Mathura and 58% of the product development time at Haldia (which should have been much lower).

VI. A COMPARISON OF IMPLEMENTATIONS AT MATHURA AND HALDIA: OBSERVATIONS & DISCUSSION

![Comparison of Phase-Wise Requirements for e-Samadhan at Mathura and Haldia refineries](image)

Fig 3: A comparison of time taken (in days) for different phases of development of e-Samadhan at Mathura and Haldia refineries
Since the system was envisioned and developed from scratch at Mathura, the data preparation for the system started in the early phases of system development. The design of the system took the data-handling capabilities into consideration. On one hand, design changes were made to suit the type of data available. On the other hand, data formatting was done frequently to adapt it to the system requirements. The important point to note is that data preparation and formatting was being done simultaneously with system specification and design. Hence, when the coding was completed, the system had a fully validated database too. This reduced the time in the integration and testing phase. Hence, this phase took 60 days at Mathura. But the situation at Haldia was very different. The product development till the implementation phase went on smooth and fast. This was because these phases gained from the maturity and experience of the previous implementation at Mathura. But the final phase of integration and testing relied heavily on the availability of properly formatted and validated data. This was an area that had received very little focus. The reason was the obvious lack of interest in the tedious and time-consuming task of data collection, validation and formatting. A lot of data-related jobs were pending till the last phase. Data collection about the various types of hardware had to be done by field visits and interaction with users. Asset tagging needed to be done in the field. Network devices had to be discovered and a complete network diagram to be formed. This involved the checking and tagging of of UTP and fiber-optic cables, ports, I/O boxes, switches and routers. All this data had to be compiled properly, validated, formatted and uploaded to the database. The cost of ignoring the importance of data for complete product development was incurred in the form of delay in the final phase of integration and testing. Extreme time pressure during the final stage lead to decreased productivity. This is attributed to the reasoning that time pressure leads to a faster rate at which errors are made, which leads to a further delay in the completion date. Consequently, this phase is still under progress and is projected to be completed by 31-May-2007, taking a total time of 66 days. The above data shows that even after following the principles of software engineering, software projects can get delayed due to lack of attention to the importance of data. After all, the software code will ultimately depend upon data for achieving the designed functionality.

Therefore, based on the above discussion, it can be observed that:

1. **Software reuse in the right manner reduces the time and cost of software development (till implementation phase) to a great extent.**

2. **Integration and testing phase rely to a great extent on availability of accurate production data. Ignoring the importance of data can lead to software delays.**

**VII. HIGH QUALITY DATA MODELS: AN OVERVIEW**

A data model defines the structure and meaning of data. It provides support to data and computer systems by providing the definition and format of data. If this is done consistently across systems, compatibility of data can be achieved. If the same data structures are used to store and access data then different applications can share data. This can result in systems and interfaces that are easier and less costly to build, operate and maintain. Figure 4 shows the standards required for creating high-quality data models.

**VIII. PROPOSED DATA MANAGEMENT SOLUTION: CHECKLIST/QUESTIONNAIRE**

Hence, I propose an additional checklist/questionnaire, which deals with the data requirements of software projects. This must be referred during the various phases of software development life cycle. Following this list will help ensure that the data is ready when the software coding is complete.
and the product is ready for integration and testing.

### TABLE 3: PROPOSED CHECKLIST/QUESTIONNAIRE TO BE REFERRED DURING SDLC

<table>
<thead>
<tr>
<th>Phase</th>
<th>Proposed Checklist/Questionnaire</th>
</tr>
</thead>
</table>
| Requirement analysis | 1) What is the type of data that the system would be managing?  
2) Where is the real-world data currently available (e.g., it can be in files, documents, CDs etc)  
3) Is the real-world data heterogeneous or homogenous?  
4) How much effort would be required to convert the real-world data into system-specific data?  
5) What efforts would be required to modify the real-world data as specified by the constraints of the system? |
| Specification analysis | 1) Has a conceptual schema (data model) been developed? [A conceptual schema describes the semantics of an organization. This consists of entity classes (representing things of significance to the organization) and relationships (assertions about associations between pairs of entity classes)]  
2) Is an ER diagram required for effective specification?  
3) Have the exact contents of each data store (along with their representation format) been defined?  
4) What are the queries that are going to be put to the product? The issue of immediate data access depends on these queries. |
| Planning | 1) If any data conversion is required, has it been taken into consideration?  
2) If data is to be collected for the system from the real world, what provisions have been made for the same?  
3) What resources (man, machine, money, time) are required for data collection? How are these resources to be provided?  
4) What approvals from higher management are to be taken for gathering of data at a central location? |
| Design | 1) Has a logical schema (data model) been developed? [A logical schema describes the semantics, as represented by the chosen data manipulation technology. This consists of descriptions of tables and columns, object oriented classes, and XML tags, among other things]  
2) Are the required licensed software for data storage (e.g., database management system) available?  
3) Is any training to be provided to the developers for using the selected data management system?  
4) Has the methodology for testing of database design and (also the data itself) been developed? |
| Implementation | 1) Has a physical schema (data model) been developed? [A physical schema describes the physical means by which data are stored. This is concerned with partitions, CPUs, tablespaces, and the like]  
2) Are proper naming conventions being followed for developing data structures for storage of data?  
3) Is the data structured properly to provide support for testing of code?  
4) Is the product coding able to handle all the kinds of data to be used in the final production system?  
5) Are there any specific data requirements for the database management system being used? If so, does the available data fulfill those requirements?  
6) Are there any modifications to be done in the real-world with respect to the fulfilling the data requirements? Have they been done?  
7) If the software needs to be reused on another hardware platform, are there any portability issues with respect to data? |
| Integration & Testing | 1) Is the data flowing smoothly throughout all the modules of the product?  
2) Are all the modules designed to handle similar data in a uniform manner?  
3) Is the real-world prepared to accept the constraints that would be imposed by the system with respect to data?  
4) Is there some type of data which is not currently uploaded into the system but may eventually be encountered? Is the system robust enough to handle this type of scenario? |

**IX. CONCLUSION**

Software projects are increasing in size and complexity. With the new business models of outsourcing, physical boundaries have disappeared resulting in intense competition among software companies. Hence, it is of vital importance that software projects be managed in a timely and professional manner. This paper described the software development process of the product e-Samadhan, which is an AMC management software being used at four major units of Indian Oil Corporation Limited. It described the complete product development life-cycle at two sites – Mathura Refinery and Haldia Refinery - which have been handled by me. Also, a comparison of the software development process at both locations has been given. Brief overview of data models has been provided. Finally, a checklist/questionnaire to be referred during the software development process to minimize data-related problems has been proposed. This work is important since it can be related to a number of scenarios, particularly in context of Indian companies. Also, data management is crucial in Indian context since lot of data is available in a variety of heterogeneous legacy systems.

**ACKNOWLEDGMENT**

I would like to specially thank Mr. P.K. Mittal [Chief Manager-Information Systems] and Mr. P.N. Sharma [Senior Manager-Information Systems] for their valuable guidance regarding both theoretical and practical aspects during the entire development cycle of e-Samadhan. Also, I convey my thanks to the entire team of CMC Computers Ltd., Accel ICIM, Wipro Ltd. and CMS Computers Ltd. posted at Mathura and Haldia for their efforts in making this system a success. Lastly, I extend my thanks to the IS group at Mathura and Haldia for their cooperation and team effort in making e-Samadhan a core strength of both refineries of Indian Oil.

**REFERENCES**

2. Software Engineering Institute, Carnegie Mellon University. Available at URL: http://www.sei.cmu.edu  
5. Indian Oil Corporation Limited website. Available at URL: http://www.iocl.com  
8. Chrysanthos Dellorocas, “Toward Exception Handling Infrastructures for Component-Based Software” of Adaptive Systems and Evolutionary Software Group, MIT  