

# Strategic Design Issues Encountered with the SAIV Process Model: A Case Study

**LiGuo Huang**

Computer Science Department  
University of Southern California  
Henry Salvatori Computer Science Center  
Los Angeles, CA 90089-0781  
+1 213 747 1814  
liguohua@usc.edu

## ABSTRACT

Ensuring on time, within-budget delivery is increasingly difficult in the information technology (IT) field because of the increasingly rapid rate of requirement volatility of IT systems under development. This paper reports some strategic design issues encountered in a digital library project applying the Schedule as Independent Variable (SAIV) approach and how they are solved in this case study.

## Keywords

SAIV, stakeholder, architecting, schedule, requirement, core capability, software process, strategic design

## 1 INTRODUCTION

We have been applying strategic design approaches [1] to the development of digital library applications using the Schedule as Independent Variable (SAIV) Process model [2] in CS577ab (Software Engineering) course projects in the University of Southern California. The approaches include application of the Benefits Realization Approach Results Chain to identify key initiatives and stakeholders [3], business case analysis, value-based stakeholder win-win feature prioritization, architecting for ease of rapid change, and using dynamic in-process information to modulate feature content to meet the schedule target. Here we summarize an example application and identify strategic design issues involved in implementing it.

## 2 PROJECT DESCRIPTION

University of Southern California (USC) subscribes to many different vendors' databases that index and abstract popular and scholarly articles and make them available over the World Wide Web. In some cases these databases

include the full-text of the articles as well. A major problem faced by faculty, students, researchers and librarians is identifying where a particular periodical title is indexed, the dates of coverage and whether it is available in full-text. Title lists are available from the vendors but are large files and can be difficult to use on the fly.

The project is named Fulltext Title Database. By applying the Benefits Realization Approach Result Chain in the Life Cycle Objective (LCO) Operational Concept Description (OCD) stage, we worked out with the USC library stakeholders that the initiative of this project is to retrieve and consolidate the information in vendors' title lists into a searchable database that can be accessed via the web. Currently, users of USC's electronic resources spend a great deal of time trying to find sources in full-text rather than focusing on their research topics. Having a comprehensive, easily searched collection of full-text titles would fill a serious gap in the tools available to researchers and students.

The Fulltext Title Database project is one of the USC digital library projects using the USC Model-Based (System) Architecting and Software Engineering (MBASE) approach. It is proposed and implemented for faculty, students, researchers and librarians at USC. Its target customer is the Information Service Division (ISD) in USC. Specifically, the clients are Greg Fleming and Anne Lynch from USC ISD public service division. The developer is a five-student team in the CS577ab (Software Engineering) class in USC. The future system administrator will be Greg Fleming and the future software maintainer will be an ISD employee. The data sources are from vendors' databases that USC subscribes to. The Fulltext Title Database is a web-based full-text journal title searchable database. It retrieves and consolidates journal title information from different vendors' databases that USC subscribes to. Unlike the more general Jake project at Yale University and full-text journal and newspaper search engine at Indiana University, this product is focused on

full-text journal title information for full-text journals available to the USC community.

### 3 SHARED VISION AND EXPECTATION MANAGEMENT

Many software projects lose the opportunity to assure a rapid, on-time delivery by inflating client expectations and over-promising on delivered capabilities. In this project, we avoided this by obtaining stakeholder agreement that meeting the 24-week schedule for delivering the system's Initial Operational Capability (IOC) is the most critical objective, and that the other objectives such as the IOC feature content can be variable, subject to meeting acceptable levels of quality and post-IOC scalability.

Often, the clients and developers (i.e. computer science students) have unrealistic expectations about what is easy or hard for each other to do. We analyzed the developer and client "simplifiers and complicators" in order to improve our ability to converge on a realistic set of expectations for the Fulltext Title Database system. Simplifier/Complicator analysis will reduce over-expectations and the frequency of projects specifying unachievable requirements.

Our project belongs to the domain of Automated Reference Services and FAQ's. In general, the simplifiers of our project are:

- Restricted query vocabulary and format;
- Simple and stable filter policy;
- World Wide Web (WWW) interface.

The five complicators are:

- Natural language processing of query

This was not a client expectation.

- Complex or volatile filter policies

Regarding the corresponding simplifier, clients only wanted the exact key word mapping when searching journal titles. And they required us to build web interfaces for both users and the system administrator. In the WinWin agreement, client Greg required us to implement searching by keywords in journal titles first, and leave searching by ISSN and date in evolution requirements. Based on his requirements and the original WinWin, we designed the software architecture and made the life cycle plan for our project. In the Life Cycle Architecture Review Board, Anne Lynch proposed a new requirement for our project. She hoped that we could also implement searching by ISSN, volume, number and date as the core capability. Maybe she thought it was not so hard to implement. But this requirement became a complicator for this project since the flexibility of users' searching requests will complicate the filter policy in our system. We didn't commit to deliver the

new feature in the following spring semester because (1) those new requirements would change the software architecture and (2) those new requirements would increase the difficulty to finish the project in the short 12-week construction phase next semester. We temporarily didn't include those improvement suggestions into our Life Cycle Architecture (LCA) package but we agreed to negotiate this new requirement in Rebaselining Life Cycle Architecture (RLCA) phase. The result was to put this as a lower-priority capability requirement to be discussed in the RLCA phase.

- Single-criterion optimization

Our client didn't propose any searching performance requirement. But they only cared about the correctness of searching results. Every time we asked them about the search performance requirement, they always said they wouldn't have any specific performance requirement. However, after they begin to use the actual product, they might complain about the searching speed if it's not fast enough. We accepted their requirements as desired, but identified performance scalability as a potential risk to manage.

- Use vendor-specific architecture

We built the web interfaces mostly using the MS FrontPage 2000 which sometimes has difficulty being interpreted by browsers except for Microsoft Internet Explorer itself. But the web-based GUI should support both Netscape Communicator 4.0 or higher and Microsoft Internet Explorer 4.0 or higher. Again, we identified this as a potential risk to manage.

- Scope crosses organizational boundaries

The maintenance of our proposed system will be the responsibility of ISD infrastructure division. However, our client Greg who will be responsible for updating the Fulltext Title Database is from ISD public service division. Our ISD management client Anne Lynch is helping us in resolving this issue.

The resulting shared vision has enabled the stakeholders to rapidly renegotiate the requirements as we encountered changing conditions.

### 4 FEATURE PRIORITIZATION

With MBASE at USC, stakeholders in this project used the USC/GroupSystem.com EasyWinWin requirements negotiation tool to converge on a mutually satisfactory (win-win) set of project requirements. One step in this process involves the stakeholders including the clients and developers prioritizing the requirements by assessing their relative importance and difficulty, each on a scale of 0 to 10. This system provided WinWin forms for stakeholders to express their win conditions for the system, to define

issues dealing with conflicts among win conditions, to support options for resolving the issues, and to consummate agreements to adopt mutually satisfactory (win-win) options. There were three stakeholder roles in this project which are developer, customer and user. The ISD clients played the roles of both customer and user during the WinWin negotiation. This process was carried out with initial system prototyping, which helped ensure that the priority assessments were realistic.

## 5 CORE CAPABILITY DETERMINATION

Based on the prioritized features in EasyWinWin negotiation results, the developers used a mix of expert judgement and parametric cost modeling to determine how many of the top-priority features can be developed in 24 weeks under optimistic and pessimistic assumptions. For the parametric model, we used USC COCOMO II to do the business case study, which estimated 90% confidence limits on both cost and schedule [4]. We decided the core capability following the guidelines in SAIV process model. First, the core capability must be selected so that its features add up to a coherent and workable end-to-end operational capability. Second, the remainder of the lower-priority IOC requirements and subsequent evolution requirements must be used in determining a system architecture facilitating evolution to full operational capability [2]. In the Fulltext Title Database project, we finalized four core capabilities by grouping the prioritized features of the proposed system. The four capabilities of the Fulltext Title Database system are as follows:

- Provide a full-text journal title search capability

It aggregates two of the topmost-priority features. One is to use exact keyword mapping when searching the Fulltext Title Database; the other is to provide complete full-text journal title and the hyperlink to its location (i.e., the vendor's database which stores the journal's full-text). In the RLCA phase, the client wants to add a new feature of being able to search by the combination of title and volume, title and number or title and date besides searching only by title keywords. However, we only assign a lower priority to this feature in case that we can't meet the schedule.

- Update the Fulltext Title Database

It incorporates another two topmost-priority features. One is to enable the system administrator to update the Fulltext Title Database using new vendors' title lists; the other is to provide security to restrict the update of Fulltext Title Database to only system administrator.

- Administrator password maintenance

It provides the feature to let the system administrator change his/her password in the future. This feature has the high priority.

- Accommodate vendors' title lists in various formats.

It aggregates three medium-priority features which include adding the new vendor's title list format, modifying existing vendor's title list format and deleting existing vendor's title list format.

The first three of those capabilities form a coherent, end-to-end Core Capability which provides value to users, and which COCOMO II estimates even pessimistically can be developed in 12 weeks. In the most-likely case, we will have enough time to add the fourth capability into the IOC. The core capability also includes a solid underlying architecture supporting which is easy to scale up to the full feature set and workload.

## 6 INCREMENTAL DEVELOPMENT

We have planned to finish implementing the system core capabilities, that is the first three capabilities on March 22, 2001. There is about 20-30% of the schedule remaining. An important step at this point is to provide the operational stakeholders (clients, users, system administrator, software maintainers) with a Core Capability Demonstration. If we succeed in passing the Core Capability Demo, we are going to use the remaining time to add the medium-priority features into the IOC.

## 7 ADAPTATION TO CHANGE

There are three major sources of change, which may require reevaluation and modification of the project plans:

1. Schedule slips. This is mainly caused by the unforeseeable technical difficulties, staffing difficulties, customer or supplier delays, etc. In the Fulltext Title Database project, because the clients were not from the ISD infrastructure core and they couldn't provide the developers the server themselves, it took us almost three months to get an unsupported server to work on. It has already delayed the start time of implementation. And the technical difficulties for us are that we can't physically access the unsupported server and ISD doesn't allow CGI scripts running on ISD server for security purpose. However, we need to install and configure Tomcat, MySQL and JDBC. Furthermore, when we encounter some technical problems when implementing the system, it'll be difficult to tell whether it is the problem of our program or the unsupported server itself.
2. Requirement changes. These may include changes in priorities, changes in current requirements, or needs for new high-priority requirements.
3. Project changes. These may include staffing changes, COTS changes, or new marketing-related tasks (e.g. interim sponsor demos).

Firstly we tried to accommodate the possible changes within the existing plan. If not, renegotiation and

reconstruction of the plan is needed. If this involves the addition of new tasks on the project's critical path, some other tasks on the critical path must be reduced or eliminated. We encapsulated the foreseeable sources of change within modules [5] in order to adapt the change in the future. For example, we incorporated one low-priority feature into the searching capability and it has no much relationship with other features in this capability. If clients proposed new high-priority requirements in the future, we may drop this feature. And we put three medium-priority features in the last capability to accommodate vendors' title lists in various formats. If needed, we may drop or defer this capability to adapt to the new requirements and meet the schedule.

There are two evolutionary requirements for this project. One is to support more searching options, e.g., search by ISSN of full-text journals; the other is to support partial keyword searching. For example, if user wants to search "Wall Street" but he/she enters "Wal Street", the system can still find and locate the "Wall Street" full-text journal title. We also provided architecture supporting for them so as to make sure to be able to implement when needed. If our progress becomes well ahead of schedule, our architecture will enable us to promote evolution requirements into the IOC, beginning with search by ISSN.

## 8 ONE PROBLEM ENCOUNTERED

The development team believed that the database automatic recovery feature is basic for Fulltext Title Database system. However, because of the limitation of schedule, the ISD clients would rather sacrifice this feature for others they thought currently more important for them. The development team couldn't persuade them to add this feature into the proposed system since there was a 24-week schedule limit. If they can find a system administrator or software maintainer, it will not be a problem. But if not, they may feel the system is not so convenient to use as they thought. As a result, we have identified this to Anne Lynch as an issue to be addressed by ISD management.

## 9 SUMMARY OF ISSUES

The Fulltext Title Database project has successfully applied the six-step SAIV process model in order to deliver highly client-satisfactory applications on a 24-week schedule in a climate of rapid change. The main issues involved in are:

- Working with stakeholders in advance to achieve a shared product vision and realistic expectations; "simplifiers and complicators" analysis are very helpful in preventing overexpectations;
- Getting clients to develop and maintain prioritized requirements;
- Determining the core capability to fit within the high-payoff segment of the a application's production function for the given schedule;

- Architecting the system for ease of adding and dropping features;
- Implementing the system incrementally;
- Monitoring progress and taking corrective actions to adapt to change.

## REFERENCES

1. K.J. Sullivan, P. Chalasani, S. Jha and V. Sazawal, *Software Design as an Investment Activity: A Real Options Perspective*, Real Options and Business Strategy: Applications to Decision Making, L. Trigeorgis, ed., (London, England: Risk Books), 1999, pp. 215-261.
2. Barry Boehm and A. Winsor Brown, *Mastering rapid delivery and change with the SAIV process model*, Proceedings, ESCOM 2001, April 2001.
3. J. Thorp and DMR's Center for Strategic Leadership, *The Information Paradox: Realizing the Benefits of Information Technology*, Mc-Graw-Hill, 1998.
4. Boehm, B., Abts, C., Brown, A.W., Chulani, S., Clark, B., Horowitz, E., Madachy, R., Reifer, D. and Steece, B., *Software Cost Estimation with COCOMO II*, Prentice Hall, 2000.
5. Parnas, D., *Designing Software for Ease of Extension and Contraction*, IEEE Trans. Software Engr., March 1979, pp. 128-137.