



Proceedings of the ISCA  
16<sup>th</sup> International Conference

---

---

## COMPUTERS AND THEIR APPLICATIONS

---

---

Seattle, Washington USA  
March 28-30, 2001

---

---

Editor: C-C. Hung

---

---

A Publication of  
The International Society for  
Computers and Their Applications - ISCA

---

---

ISBN: 1-880843-37-4

## **SESSION 11A: SOFTWARE RELIABILITY, REUSABILITY AND MAINTENANCE**

1. **A Methodology for Developing a Probabilistic Security Measure using a Software Reliability Model**  
Charlie Yong-Sang Shim and Rex E. Gantenbein (University of Wyoming, USA) ..... 517
2. **An Aspect-Oriented Approach to Supporting the Design of System Software**  
P. Netinant (Bangkok University, Thailand, and Illinois Institute of Technology, USA),  
C. A. Constantinides (Loyola University Chicago, USA) and T. Elrad (Illinois Institute  
of Technology, USA) ..... 521
3. **A Design Oriented Approach to Software Maintenance**  
Jaehyoun Kim, Sargon Hasso and C. Robert Carlson (Illinois Institute of Technology, USA)..... 525
4. **An Approach to Remote Software Project Management and Development**  
Vladimir Zanev (Columbus State University, USA), Avram Eskenazi (Bulgarian Academy of Sciences,  
Bulgaria) and Seong-Moo Yoo (Columbus State University, USA) ..... 531

# An Approach to Remote Software Project Management and Development

Vladimir Zanev  
Dept. of Computer Science  
Columbus State University  
4225 University Avenue  
Columbus, GA 31907  
[zanev\\_vladimir@colstate.edu](mailto:zanev_vladimir@colstate.edu)

Avram Eskenazi  
Dept. of Software Engineering  
Institute of Math and Comp. Sci.  
Bulgarian Academy of Sciences,  
1113 Sofia, Bulgaria  
[eskenazi@math.bas.bg](mailto:eskenazi@math.bas.bg)

Seong-Moo Yoo  
Dept. of Computer Science  
Columbus State University  
4225 University Avenue  
Columbus, GA 31907  
[syoo@colstate.edu](mailto:syoo@colstate.edu)

**Abstract:** This paper describes an approach to remote software project management and development using thin client/fat server model. The customer and the developer are remote distributed. The authors refer to their experience with several remote developed projects and try to improve the cost effectiveness by reducing the number of the joint meetings between customer and developer. The developer is using web-based interface to develop and maintain a software project using the customer Intranet.

**Keywords:** remote software project, client/server model, Intranet, remote database access, Web application, Active Server Pages.

## 1. Introduction

Making effective use of Internet and Intranet technologies has been critical for software companies to lower the cost of software project development and of obtaining competitive advantages. The software industry was (and often still is) drowning in complexity and failing to deliver many software projects on time or to budget. The software development cycle (SDLC) [4] phases: concept phase, requirement phase, design phase, implementation phase, integration and test phase, and maintenance phase can be carried out in different locations geographical distributed. The customer initiating a software project could be located in one country and the design and/or programmer team could be located in another country where the labor cost is relatively low. Several software development projects have been implemented by the Software Engineering Department (as developer) of the Institute of Mathematics and Computer Science at the Bulgarian Academy of Sciences in collaboration with and as proposer (or customer) DaimlerChrysler Dornier, Germany.

The following constraints were imposed on these projects [2]:

1. The contractual relationship is at fixed price, and the project deadline is determined in advance without an opportunity to be reconsidered.
2. The customer is responsible for the:
  - request for proposal,
  - requirement definition,

- external (general) specifications,
  - independent testing.
3. The developer is responsible for the:
    - project management scheme and the quality assurance plan,
    - detailed specifications,
    - programming (coding) and internal testing,
    - maintenance documentation.

The primary challenge in such projects is to develop software with the three famous objectives in mind:

1. Within budget (fixed price)
2. On schedule
3. According to requirements

In an environment where the customer and the developer are located remotely the Software Quality Assurance (SQA) approach with appropriate plan and policy in remote software development environment [2,5] is a milestone and a background for successful software project development. The experience and the careful coordination of these software projects require 3-4 joint meetings of both parties - the customer and the developer during the project life time: a joint meeting to conclude the contract, a joint meeting to design the prototype, and one or two joint meetings for acceptance tests [2]. The joint meetings comprise an essential part of the overall remote project expenses. Working in a client/server environment with appropriate working procedures to support the remote development of the project will allow the number of the joint meetings to be reduced. In the same time the established SQA plan and policy [2] will be unchanged and maintained in the new client/server environment. A distributed software development environment where the main communication tools have been only e-mail, fax and telephone, raised the necessity of moving to a new model of software development environment using client/server methodology, concepts, and creating new working procedures.

## 2. Client/Server Model

Over the years a series of architectures have been devised in an attempt to integrate various distributed environments within an organization. Client/server computing is one of the most dominant technologies



of remote information access and has developed as the computer industry moved from centralized shared logic based system to a network servers and workstations. The most obvious example of client/server architecture is the World Wide Web. The Web architecture is based on a client/server model and uses a Web Browser (client) to retrieve information from a Web Server which may be located around the world on the Internet. Client logic can be embedded on the Web page by means of scripting languages.

Organizations are using Intranets to make it easier for their staff and partners to collaborate and locate/process information. While Internet is global and publicly accessible, an Intranet is closed and accessible only to those people who have permission to use it. In case of remote software project development the customer team (a project manager and testers) and the developer team (a project manager, SQA manager, chief programmer and 3-4 programmers) [2] are those people who have permission to work on and use requirement definitions, external and internal specifications, program code files, tests, and documentation. Special attention is paid to standards and a other legal documents, as the developer has to observe foreign standards (those of the customer) concerning documentation, some formal procedures and quality characteristics. This is the reason to organize a small library (Fig.1) containing such legal documents and standards of the customer's country and company, relevant to the project. An Intranet for remote software project development and management operates in just the same way as the Internet, using browsers to provide the customer and the developer team with remote access to corporate software project information and access.

Nowadays, Web browsers support web pages with images, sound, video, and animation. Scripting languages, Java applets, and ActiveX controls can be added also to web pages enabling user interaction and remote client/server processing. The web pages can be constructed to ensure remote access to software project files and documents, a software project database, and applications located on the Web server. The scripting languages as JavaScript and VBScript can be included within an HTML document to integrate different entities within a Web page. This enables the client to be event driven, for example, a script can be used to detect an event being fired by one control (e.g. a button click).

The Microsoft technologies ActiveX Data Objects (ADO) and Remote Data Services (RDS) [1] provide access to remote files and database contents. They enable a client Web browser to retrieve information from a database on a Web server, process that information on the client computer and return

modifications of the data to the Web server so that data can be updated in the database. The client/server architecture of a software project development allows to separate complex, centralized applications into smaller, more manageable tasks or application logic ensuring in the same time remote access to the software project files and database. These tasks can be split up into three layers [3]:

#### 1. Presentation logic

The presentation logic is handling how a customer or a developer team member interacts with an application. Usually it is implemented through a graphical user interface (GUI).

#### 2. Software project logic

The software project logic is handling the mechanics of the software project development and management. It includes software project development rules and procedures, software project requirement files, external and internal specifications, documentation files, and a relational database with stored procedures, triggers, and constraints.

#### 3. Data access logic

The data access logic manages the storage, the retrieval of data, and ensures data integrity.

We are considering relatively small software projects where the customer team consists of not more than 3-4 members and the developer team is not more than 5-6 members. In our model we are using Intranet access to a small file sharing system containing software project requirements, specifications, and documentation and a small relational database management system (RDBMS) supporting the software project development.

### 3. Implementation

The client/server architecture reduces network traffic. In addition, today's databases provide many features that enable the development of sophisticated multi-user applications. For example, multiple users can access and update the same set of data safely, or to be notified for upcoming events in time. The processing tasks of presentation logic, software project logic, and data access logic can be split between layers – the workstations (personal computers) and the database server. Such architectures are referred to as being two-tier client/server [3]. We have chosen thin client/fat server implementation in which the management and the support of a software project (the software project logic) is pushed down to the database server using techniques as stored procedures, triggers, and constraints.

When organizing the implementation we tried to follow two principles (contradictory to some extent):

1. Preserve the usual type of communication and organization in order not to change the habits and reactions of the project participants



2. Take into account the peculiarities of the remote operation and the specific organizational measures it requires

This is the reason we proposed two types of formalization:

1. Each document in the software project files (Fig.1) has a fixed format

2. A rather strict procedure for communication when using these documents was elaborated.

However, to keep the opportunity for informal communication among the staff members, the e-mail and chat-room components (Fig.1) are provided. As an example we could quote the testing procedures. It is clear that they are probably the most intensive from a communication point of view. The programming code is prepared by the developer's staff, internally tested and then delivered to the customer. The independent testers, after having performed the appropriate testing operations have to fill in a fixed form, which immediately becomes available to the developer's staff. Each of the programmers reacts to the content of this form by correcting the errors described and marks on the form his reaction and the results of this reaction. There are two points to be considered here.

1. The programmer is obliged formally to react by using the form, but (s)he is also entitled to informally communicate with the tester by using the e-mail and chat-room in order to comment or clarify some items.

2. The chief programmer is obliged to supervise how the form is proceeded and to resolve conflicts in case some of the errors are not covered, e.g. in case no programmer feels responsible for a particular error; as far as the program manager is concerned he is interested in this form from the point of how terms for reaction are observed.

A stored procedure is invoked by sending a request to the database server that includes the stored procedure name and any parameter values. A trigger is a stored procedure that is invoked automatically on the Add/Update/Delete of items in a database table or under some time constraints and events. Figure 3 below shows the structure of the software project database. The primary keys are in bold and the relationships from one table to another table are defined through foreign keys. The relational SoftwareProject database consists of tables that store software project management and development information (Fig.2). Only the customer and the developer project managers have full access (read and write) to the SoftwareProject database. The other team members have separate views and limited read/write access depending on their database view. Web-based procedures using Active Server Pages (ASP)[3] have been developed for remote database

access: user registration and login, add/delete/update tables, and database reports. Figure 4 below shows an example of a Web form for developer tests. It consists of developer name (from the login), two drop-down menus to choose a program module and a test assigned to a developer, due date, done test, and test description. The developer has access only to his/her program modules and tests and can update only his/her test description. The developer name, due date, and done fields are informative and cannot be changed. The manager and the chief-programmer have access to these fields. The developer can send a message to the manager, a team member, or through web forms to work with his/her documentation and program modules.

### 3. Conclusion

This paper has described an approach to remote software project management and development based on client/server model and implemented as Web application with ASP technology. At this stage the customer's and the developer's project members have positive subjective estimations and impressions. Experience shows that this approach has improved the quality of the software developed and lowering in the same time the cost of the software project. We hope to do a further development and refinement of the approach and its implementation described. We do believe that the approach presented in this paper can be applied to lower the cost of the software project labor force, management, and development in organizations using remote software project development and management.

### References

- [1] H. Deitel, P. Deitel, and T. Nieto, "World Wide Web. How to Program", Prentice Hall, 2000
- [2] A. Eskenazi, et al, "Project Management and Quality Assurance in a Distant Software Development Project", International Journal Information Theories and Applications, pp. 3-10, Vol 6, No 1, 1999
- [3] B. Francis, et al, "Beginning Active Server Pages 2.0", Wrox Press Ltd., 1998
- [4] J. Somerville, "Software Engineering", Addison-Wesley, 1992
- [5] I. Gorton, S. Motwani, "Towards a Methodology for 24 Hour Software Production Using Globally Separated Development Teams", Proceedings of the Third International Conference on Achieving Quality in Software, pp.50-55, London, 1996



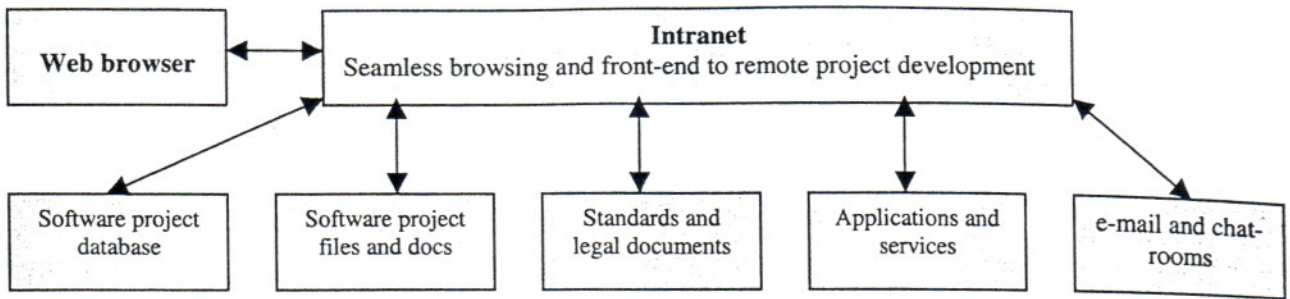


Figure 1. Intranet and Web solution of client/server software project development

Table	Description
TeamMember	Stores information about each customer and developer team member.
ProjectStructure	Describes the software project structure – program subsystems and program modules.
WorksOn	Used to establish who is responsible to develop a program module, how many hours are needed and a due date.
TeamStructure	Stores information about the team structure (project managers, SQA manager, chief programmer, programmers, and testers).
Dtests	Describes a set of internal program module tests needed to debug the software.
Ctests	Describes a set of external program module tests for software acceptance.
Ddocumentation	Program module internal documentation
Department	Department name and location
Dpayment	Developer project payments.

Figure 2. SoftwareProject database table description

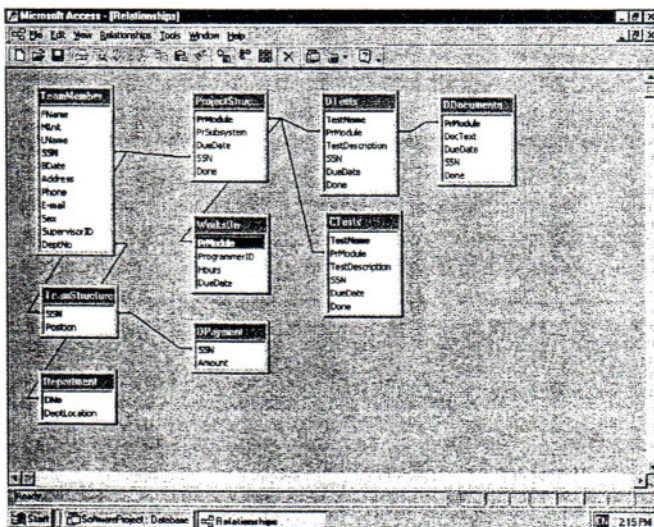


Figure 3. Table relationships in the SoftwareProject database

Figure 4. Developer tests Web form