Operational Auditing And Testing In Client/Server Systems

Parag C. Pendharkar, (E-mail:pxp9@psu.edu), Penn State University, Middletown
James A. Rodger, (E-mail: jrodger@pitt.edu), University of Pittsburgh, Johnstown
Shishir Kumar, (E-mail: skumar@pop.iscob.sei-it.com), SEI Information Technology, Chicago

Abstract

Client/Server(C/S) systems have revolutionized the systems development approach. Among the drivers of the C/S systems is the lower price/performance ratio compared to the mainframe based transaction processing systems. As most Fortune 500 companies are moving quickly to the client server systems, it is increasingly becoming important that a C/S system project be audited several times during its development so that potential errors can be avoided. In the current paper, we describe operational auditing in the C/S systems during various phases of its development.

Introduction

Client/Server(C/S) systems offer a "divide and conquer" approach to systems development [4]. The lower price/performance ratio and the power of distributed processing has made C/S systems popular in both large and small businesses. Most Fortune 500 companies have already developed or considering developing a C/S application.

The move from mainframe based applications to the C/S environment requires companies to redesign and evaluate their centralized mission critical applications. Redesigning the mission critical applications around the C/S framework is enticing to most organizations as C/S systems offer better graphical user interface (GUI), and response times compared to the traditional systems [6]. Some companies such as, Allnet Communications have already developed a strategic information system using the C/S framework [5]. Some of the C/S applications can be termed as strategic applications as these applications help corporations accomplish "fundamental competitive objectives" [2]. Given that more strategic applications will be built around the C/S framework, the errors and delays in a C/S project can have serious consequences to a company.

Among the risks in developing the C/S systems are the delays in implementation and the inability of an organization to abandon a troubled project. Our consulting experience in developing three different C/S projects for different government and private sector companies show that most C/S projects have a schedule and budget overrun. The schedule overrun of a C/S project means a weakened competitive position of the company. Furthermore, for a strategic application built using C/S framework, schedule overrun can eliminate any strategic advantage that the

Readers with comments or questions are encouraged to contact the authors via e-mail.
application may have by the time the system is operational. As the basic business need that mandated the need for C/S doesn’t change with the delays and budget overrun, it becomes extremely difficult to abandon a C/S project. The strategic importance of the project, corporate visibility, focus on short term goals and the pressure to succeed, leads some project managers to take risky shortcuts in a C/S project.

The risk proneness of the C/S project, new development methodology and the strategic importance of the C/S project, necessitates auditing and testing of the C/S project at various phases at development. In the current paper, we describe the different types of auditing and testing approaches that can be used to ensure the overall success of a C/S project. The rest of the paper is organized as follows: In the next section, we provide an introduction and a brief review of C/S systems in the industry. After the review, we detail a procedure for auditing and testing in C/S systems. We then conclude this paper with a section on discussion.

**Client/Server (C/S) Systems**

Client/Server (C/S) system is a type of computing model that allows developers to split the application processing load between two logical processes: the client and the server. The client in the C/S system is usually a desktop computer running a front-end software (Powersoft’s PowerBuilder, Borland’s Delphi, or Microsoft’s Visual Basic). A client can send a request to a server and process information returned from the server. The server (Oracle, Sybase, SQL server) receives and processes a request from the client and performs the requested operation. The processing in C/S environment takes place at two places: the client and the server.

There are three components in a C/S system: the front-end client software, middleware, and the server software. The front-end client software includes application development tool such as PowerBuilder, Delphi, or Visual Basic and reporting tool such as a spreadsheet or a word processor. The application development tools facilitate rapid application development (RAD) which allow a developer to create user interfaces by painting screens using prebuilt GUI controls. The reporting tools allow users to create ad hoc reports and graphs. Middleware is a communication mechanism that exists between the client and the server. Middleware provides navigation through the front-end GUI, operating system, network, and the database layers. There are different types of middleware. Different type of middleware is required for database access.

![Figure 1: The Basic and two-tier Architecture for Client/Server Systems](image-url)
(ODBC) or object request brokers (ORBs), communication (RPC or message oriented middleware) and security (distributed computing environment (DCE)).

There are different types of architectures for C/S systems. Figure 1 shows the different types of architectures. The basic architecture splits the processing load between the client and the server. Although the basic model provides better performance by decoupling the application processing, problems occur when more users are added. The basic model provides limited scalability.

As complex mission critical applications are being built using C/S systems there is a need for the C/S system to be scalable. Two-tier and n-tier C/S architectures provide the scalability and improve the performance of C/S systems by a large magnitude. In two-tier and n-tier C/S architecture an extension to the basic architecture is made to distribute application processing load on one or more application servers to best optimize the available resources. Among the factors governing the type of architecture for C/S system are: user load, processing load, application type, cost of development, maintenance, and performance expectations.

Auditing in C/S Systems

Auditing is defined as "... A systematic process of objectively obtaining and evaluating evidence regarding assertions about economic actions and events to ascertain the degree of correspondence between those assertions and established criteria and communicating the results to interested users. [1]". Auditing in information systems includes system testing and substantive testing (testing transactions and balances). There are two different types of auditing: independent auditing and internal auditing. Independent auditing is conducted by auditors who are independent of management. Internal auditing, however, is conducted by personnel internal to the organizations and it serves management in evaluating the efficiency of resource utilization (operational auditing), and the effectiveness of the management decisions (management and financial auditing).

C/S system auditing is a type of internal auditing. There are four types of internal auditing that can be conducted for C/S systems. These four different types of auditing are: Financial auditing, Compliance auditing, Fraud auditing, and Operational auditing. Our major focus in this paper is on operational auditing but, first we will discuss financial auditing, compliance and fraud auditing very briefly.

Financial auditing is need in a C/S project to ensure that project is on schedule. Financial auditing is the responsibility of the C/S project manager. Each project manager should regularly check the project team members time sheets to ensure that team members are billing the project for the work that is completed on the project. A project manager, in turn, can relate the percentage of project completed to the percentage of the budget used. Compliance and fraud auditing is required to evaluate the management procedures employed in the development and maintenance of the C/S. Special attention should be given to evaluate the compliance for C/S applications that are mission critical and/or are considered strategic. Fraud auditing is necessary to control the programmers from sabotaging the software and using the pirated software in rival organizations.

The key internal auditing in C/S is operational auditing. Most of the C/S projects skip the operational auditing phase and face problems in performance expectations, and general protection faults [3]. The reason project managers tend to neglect operational auditing is that they assume rapid application development (RAD) tools to protect them from application development problems. Furthermore, with budgets getting tight, most managers throw caution to the wind. Operational auditing in C/S is an important element for the deployment of quality applications. We propose four different types of operational auditing in C/S system. Figure 2 illustrates the four different types of operational auditing.
Operational Auditing in Client Server Systems

Portability auditing, in general, is auditing the C/S application to ensure that certain changes in C/S application don't cause an application to become unstable. For example, if a user upgrades his or her operating system from Windows 3.1 to Windows NT workstation then portability auditing should be able to identify any incompatibilities. Furthermore, if C/S application is designed using Visual C++ then it is likely that newer version of Microsoft Foundation Classes (MFC) may cause older version of client application to be inoperable. In one of our projects, we noticed that some of the stream functions work well in MFC version 4.0 whereas in newer version of MFC 5.0 the stream functions were inoperable. The solution to our problem was to define standard namespaces. Our auditing procedure revealed that some sections of the client applications are dependent on the version of the software that is used. After proper portability auditing, we were able to develop an application that was portable across different versions. On the server side, portability auditing provides guidelines for allowable changes to database schema so that front-end application is not affected. Given the technical nature of the portability auditing, there are some tools such as, PreVue-X that allows the auditor to test the impact of adding new features to an application.

Processing load auditing is a type of operational auditing where an application is audited and tested for its capability to handle the expected processing loads. Processing load auditing is conducted by running an application continuously using a large number of testing procedures that simulate many users [3]. The audit report for processing load testing should contain the saturation limit of the server, memory leaks of the front-end application, concurrency control problems, reports on disk and processor performance, system deadlocks and crashes.

Components auditing is conducted to ensure overall quality of the software engineering process. Under the components auditing, individual components are tested for proper documentation, version control statements, and formatting such as indentation and consistency of loops. In certain cases, C/S environment can be simulated on one machine by running a database server on client to do the component testing for the front end application. On the server side, a similar approach can be adopted, to test different database schema, by using SQL scripts to simulate client connections. Component testing ensures the overall system quality by ensuring quality of individual components. Among the available tools to facilitate component auditing, SQA Inc.’s SQA Robot is one example of the tool that is available that allow auditors to test new cases to simulate a particular condition.
Integration auditing involves linking the individual components to form an overall C/S system. Integration auditing is a comprehensive audit of middleware compatibility issues, ODBC driver issues, and network performance issues. Integration audit ensures that two clients use compatible TCP/IP protocol. Among other things, integration testing also needs some of the research into the information technology infrastructure that is used for the C/S system.

Discussion and Future Work

We have described basic and tier architectures of C/S systems, auditing in C/S system and focused on operational auditing in C/S systems. Auditing is an important aspect of C/S development as careful auditing keeps a C/S project on track in terms of schedule and cost. Furthermore, sound auditing policies will allow a company to avoid mistakes such as unrealistic expectations of the delivered system, and using wrong set of tools for developing applications using current system. Auditing C/S, in general can improve consultants in C/S area to better estimate costs of future projects and avoid unexpected cost over runs.

Future work in auditing in C/S can extend the current framework to audit C/S systems that may use JAVA or ActiveX and web technology. As more companies are developing intranets using corporate firewalls, it appears that architecture shown in Figure 3 is the extension of current C/S architecture that will be prevalent in the future.

The web based C/S architecture is similar to the two-tier architecture, however, the development tools for the front-end and the server are different. In web based C/S architecture, a client sends a request using HTTP protocol to the web server. The web server contains CGI (common gateway interface) protocol that interacts with the database server to obtain information requested by the client and then returns the information to the client as HTML (hyper text markup language). JAVA is a language designed by SUN microsystems that can enable a web based client to access web based server using SQL calls. It is likely that different operational auditing and testing tools will be available as more companies start developing Internet based client/server applications. Future research is needed to address this issue.

Overall, auditing and testing in Client/Server systems is beneficial to both project managers and to the companies. Proper auditing and testing would help managers improve their visibility in their companies by delivering quality systems without schedule overruns.

References

2. Lederer, A.L. and Mendelow, A.L. "Convincing Top


