

Group: 3D'S
Project: Media Server
Document: Final Report
Date: April 25 2005

1.0 Product-related information:

1.1 Current status of product:

What is the current status of your product?

A basic usable framework for a home media system.

Are there parts of the product that have not been adequately tested?

We did not test the Database Server's support of bonus features because their extraction has not yet been implemented in the Media Conversion Tool.

Are there product features that could not be implemented due to time or resource constraints?

Yes. The system currently does not support the extraction of bonus features from DVD's, music, and advanced file placement. Although the recommendation system does provide recommendations, it is less intelligent than we had originally hoped.

1.2 Recommended work:

Known faults include: The movie being viewed must be completely downloaded from the streaming server before it can be viewed. A movie appears in the database before it has been encoded and downloaded to the proper location.

Possible enhancements include: Making recommendations based on more than a title (i.e. favorite actor, genre, etc.). The current file placement system is a round robin selection on a list of computers (ip addresses) hard coded into the Database Server. This could be made enhanced to support a dynamic list of computers on the network as well as query them for available space and make a reasonable decision based on this information. The media conversion tool could be enhanced to extract bonus features. Additionally a file format that is more conducive to streaming could be used in place of the current format (.avi). The conversion tool currently has a local copy of the IMDB database that has a built in update feature. This feature does not work through cygwin but could be modified to do so.

Suggestions for new features include: Add support for music. This includes ripping CDs, storing their location in the database, setting up the recommendation system to create playlists based on user input, and anything associated with music. A user friendly installation process could be implemented. For a computer engineering project you could additionally create hardware to interface the media server with a home entertainment system. This could include creating a box that runs a client and supports streaming which plugs into a TV. Allow the user to view movie trailers through the recommendation system.

1.3 Advice to teams continuing this project:

Finish all documentation early, particularly any that involves the design of the application. Write test cases early and make sure they are thorough. If at all possible work with people you know and have worked with before.

2.0 Project team information: (*Postmortem analysis*)

2.1 Management objectives and priorities

Our objectives and priorities did not change so our original plan was accurate. It is as follows, "Each person will be assigned tasks and follow up will take place during each of our scheduled meetings. We want to have an open management style where suggestions and input are expected from each member of the group, and the group respects and considers the input given by others. The purpose of the management is to have a single person who is responsible for the follow up."

2.2 Final team structure.

Who was responsible for which tasks?

Drew: Primary: Project Leader, UI Leader, Source Code Librarian
Secondary: Website Coordinator, Conversion Tool, Network

Dave: Primary: Schedule Coordinator, Recommendation Leader,
Database Co-Leader
Secondary: Project Leader

Danny: Primary: Website Coordinator, Conversion Tool
Secondary: Database Leader, Document Coordinator,
Source Code Librarian

Steve: Primary: Document Coordinator, Network Leader,
Database Co-Leader

Did you rotate any of the roles?

No, however roles evolved through the course of the project due to the interaction between various roles.

Pretend you are just starting the project armed with your current experience. What would your team change about the team structure (if anything)?

Nothing, from previous projects we already knew how to work together.

What advice do you have for teams wanting to use the same structure?

Be sure to take interests and strengths into account when assigning roles.

2.3 Schedule and planning:

Which aspects of your team's scheduling and planning procedures worked well?

Creating both weekly and semester schedules. Creating a development plan goals for each release.

Which problems did your team have with scheduling and planning?

None.

What tools did your team use to assist you in planning and tracking the project?

Our schedule links from the webpage (weekly and semester schedules and development plans)

How could your team have improved its scheduling and planning?

Creating a time line for each release at the beginning of the project.

2.4 Support functions:

Discuss the effectiveness of your team's Quality Assurance.

We didn't do any explicit Quality Assurance. It was accomplished implicitly through testing and integration.

What procedures did the team establish for defect tracking? How successful has the team been in faithfully using these procedures?

Our original procedure for documenting defect tracking was inadequate. However, defects were tracked through comments in the code as specified in section 5 of our Coding Convention. These comments were removed as the defects were corrected.

What support function lessons did the team learn that would have helped if you had only known them earlier in the semester?

Thoroughly test components for defects before integration. Document all defects and when they were corrected in a persistent manner.

2.5 Work with the clients:

Not Applicable.

2.6 Work with project mentors.

Discuss your team's experiences with the project mentor arrangement.

Not Applicable

What advice does your team have for how this program can be made more useful from the point of view of the student teams?

The scope of the documentation was too large for the length of this course. We feel the documentation's scope would have been beneficial for a year long class but it was too stressful for a single semester.

Which aspects should remain unchanged?

The idea of a 10 minute meeting with the Professor as opposed to weekly lectures was good.

Other issues.

None at this time.

Feedback from the mentors:

Not Applicable.

Three general pieces of advice to future teams as they start out.

Test, Test, Test - early, often, and thoroughly.

Watch out for less than obvious problems (i.e. implementation details, bugs, etc.)

Implement the Instructor's advice in a timely fashion.