Project plan

Relevans

Helsinki 2nd March 2005

Software Engineering Project

UNIVERSITY OF HELSINKI
Department of Computer Science

Course
581260 Software Engineering Project (6 cr)

Project Group
Antti Nyberg
Päivi Suomalainen
Teppo Ahonen
Jussi Kurki

Client
Miro Lehtonen

Project Masters
Juha Taina
Turjo Tuohiniemi

Homepage
`http://www.cs.helsinki.fi/group/relevans`

Change Log

Version	Date	Modifications
1.0	7.2.2005	First version
1.1	10.2.2005	More details added
1.2	27.2.2005	Updates in responsibilities

Contents

1 Introduction

The INEX project at the University of Helsinki has developed tools and methods for XML Information Retrieval since 2002. The complete system consists of an index of XML documents and a search engine that finds relevant answers to a given query. For query processing and basic low level score computation, working scripts have been written. A new tool is needed for combining the most relevant scores.

2 Organization

2.1 Developers

The ”Relevans” is a group of four computer science stundents. Jussi Kurki is acting as a Project manager and as a chair man in weekly meetings. Teppo Ahonen is acting as a secretary.

For each of us, specific areas of responsibilities have been assigned, but in the end we try to co-operate and divide tasks as much as possible.

Developer	Area of responsibility
Jussi Kurki	Project manager
Antti Nyberg	Requirements
Teppo Ahonen	Design
Päivi Suomalainen	Web-pages, tools and testing

Figure 1: Developers

The project manager is responsible for leading the group. He writes the Project plan with help of Teppo Ahonen in risk analysis. Project manager plans the schedule for project. Managing the CVS-repository used in the project is also he’s work.

Requirements are Antti Nyberg’s area of responsibility. He leads the discussion with clients in client meetings and writes the requirements document with help from Päivi Suomalainen.

Teppo Ahonen is responsible of system and software design. He makes the architecture design and writes the design document with help of other group members.

Päivi Suomalainen designs the web-pages for group. With Teppo Ahonen she also decides on various tools that are used for the project, e.g. UML-tools, IDE etc. She is also responsible of testing the software.

Documents are written through out the process and checked and updated as needed. All members are participating on implementing and testing the software.

2.2 Management

Antti Tevanlinna is supervising the project. He gives us feedback and guidance if needed, but does not directly influence on developement.

Juha Taina is supervising the whole course and each sowftware engineering project group.

2.3 Clients

Antoine Doucet and Miro Lehtonen are members of Team INEX at the University of Helsinki. The group is doing research on structured XML-documents. The group is part of Doremi reseach group, also working in University of Helsinki.

Clients are only expected to provide knowledge and the example data needed for creating the tool. They will be checking requirements and design document. Finally they will grade the product.

3 Overview

The XML search engine developed by INEX group performs four distinct subroutines for each query:

1. Query processing.
2. Score computation for each element in the index.
3. Score combination.
4. Listing the best answers.

The score of an XML element indicates its relevance with respect to the query: the best answers should have the highest scores.

Perl scripts have been written for each of these routines. The score combination (3) is the weakest part of the system. For this, we are aiming to provide help.

3.1 Score combination

The input file for the score combiner contains the relevance score of every document fragment in the index. The fragments in the index are disjoint, e.g. no overlapping content is found. The size of these fragments corresponds to the size of a paragraph, subsection, or a small section.

The output file should contain a ranked list (sorted by relevance) of document fragments of any size. For example, if all the subsections of one section are highly relevant, the whole section is relevant and should be ranked high on the list. The score combiner tool computes the score of the whole section based on the scores of the subsections. In general, the score combiner can compute the score of any parent element when the scores of the child elements are given as input. We have implemented one method for combining the scores called Upward Propagation which weights the scores of each child element by its size.

All of the following features must be implemented in the score combiner:

- A user interface, possibly graphical
- Choice of the scorelist.
- Choice of the score combination method.
- Setting the appropriate parameters for the chosen method.
- Selected score combination methods.
- Support for additional score combination methods.

3.2 Typical use case

In a typical use case, the user selects a scorelist and a score combination method, then sets the parameters, clicks a button, and views the results. This is repeated with different settings. Optionally, several runs can be launched simultaneously.

4 Working methods

The project will follow the waterfall model. The process of creating software will be done in several stages. First, requirements are collected and defined. Then the software is designed. This is followed by actual implementation of the product and some unit testing. Finally the whole system is tested and delivered to customers. After each stage, a document is produced.

5 Risk analysis

ID	R0
Risk	The risk
Description	More precise description of the risk
Probability	small, moderate or large
Seriousness	trivial, tolerable, serious or devastating
Precautions	Precaution(s) done to prevent the risk or to minimize the effect.
Actions	Action(s) taken if the risk occurs.

ID	R1
Risk	Not understanding the problem field
Description	Group either doesn't understand or misunderstands the problem field of the project.
Probability	small
Seriousness	devastating
Precautions	Studying the problem field well beforehand. Everything must be understood in the requirement analysing phase, since afterwards it is too late.
Actions	-

ID	R2
Risk	Changes in the problem field
Description	Problem field changes for unpredictable reasons.
Probability	small
Seriousness	serious
Precautions	Making the product as modular and modifiable as possible. Discussing with the client of what can be done.
Actions	Rearranging the project schedule.

ID	R3
Risk	Not understanding the quality requirements
Description	Group either doesn't understand or misunderstands the quality requirements of different parts of the project.
Probability	small
Seriousness	serious
Precautions	Making a well-crafted requirement analysis: clarifying the gap between the number of actions and the quality of them. This demands a lot of communication with the client.
Actions	-

ID	R4
Risk	Weak requirement analysis.
Description	The requierement analysis does not cover everything that client requires or covers them falsely, but is still accepted by the client.
Probability	small
Seriousness	serious
Precautions	Writing the requirement analysis carefully and making the client to read it carefully.
Actions	-

ID	R5
Risk	Changes in requirements
Description	Client's requirements toward the product change.
Probability	moderate
Seriousness	tolerable
Precautions	Trying to understand client's goals better so changes won't be as comprehensive. Clarifying the requirements with the client.
Actions	Rearranging the project schedule and tasks: are the new requirements implementable in the time that is left.

ID	R6
Risk	Group's skills in software engineering
Description	Group's skills in software engineering are too limited.
Probability	moderate
Seriousness	serious
Precautions	-
Actions	One member of the group studies carefully the subject that causes problems and then guides the rest.

ID	R7
Risk	Group's technical skills
Description	Group's skills in used technologies (java, xml) are too limited.
Probability	small
Seriousness	serious
Precautions	-
Actions	One member of the group studies carefully the subject that causes problems and then guides the rest.

ID	R8
Risk	Problems in outer communication.
Description	Problems in communication between the group and the clients.
Probability	small
Seriousness	tolerable
Precautions	Putting more effort into the communication, writing a proper communication plan.
Actions	-

ID	R9
Risk	Problems in inner communication.
Description	Problems in communication between the group members.
Probability	moderate
Seriousness	serious
Precautions	Strict reporting: regular reporting of the situation.
Actions	-

ID	R10
Risk	Changes in the group's lineup
Description	A group member withdraws from the project.
Probability	small
Seriousness	serious
Precautions	Spreading the knowledge inside the group so that none of the group members is critically irreplaceable. Putting more effort into communication.
Actions	Rearranging the project schedule and tasks: prioritizing the undone parts of the software and discussing with the client of what can be left undone.

ID	R11
Risk	Sickness etc.
Description	One or more of the group members gets sick during the project.
Probability	small
Seriousness	serious
Precautions	Spreading the knowledge inside the group so that none of the group members is critically irreplaceable.
Actions	Prioritizing the tasks and rearranging them between the healthy members of the group.

ID	R12
Risk	Group member "steals" a task.
Description	A member of the group grabs a task and doesn't let other members to understand / work on the task.
Probability	moderate
Seriousness	serious
Precautions	Improving the project managing.
Actions	-

ID	R13
Risk	Closing different phases of the project
Description	Being unable to close a phase of the project before moving into the next phase.
Probability	large
Seriousness	serious
Precautions	Deciding several critical milestones and holding onto them. Each project member must estimate the work left on current phase. The tasks of different phases must be accurate (no lapping with other phases).
Actions	Updating schedule and priorizing the tasks left.

ID	R14
Risk	Poor testing
Description	Either no time for sufficient testing or testing methods are inaccurate.
Probability	moderate
Seriousness	serious
Precautions	Planning project schedule carefully and studying software testing. Making one of the group's members responsible of the testing. Planning testing already in implementation phase and automating testing as much as possible.
Actions	-

6 Hardware and software requirements

There are no special hardware or software requirements for this project. The program will be written in Java the programming language using version 1.4 of Java Standard Edition. It can be run on several platforms, e.g. Linux, Windows or Mac OS with standard hardware.

7 Project schedule

Task	Date
First version of the project plan is ready	10.2.2005
Requirements document is mostly ready	18.2.2005
Designing of the software starts	24.2.2005
Validation of the requirements document	24.2.2005
First draft of design document is ready	9.3.2005
The validation of the design document	18.3.2005
Implementation stage begins	23.3.2005
The design document is ready	26.3.2005
First demonstration of the software	6.4.2005
Project ends	8.5.2005

Figure 2: Important dates

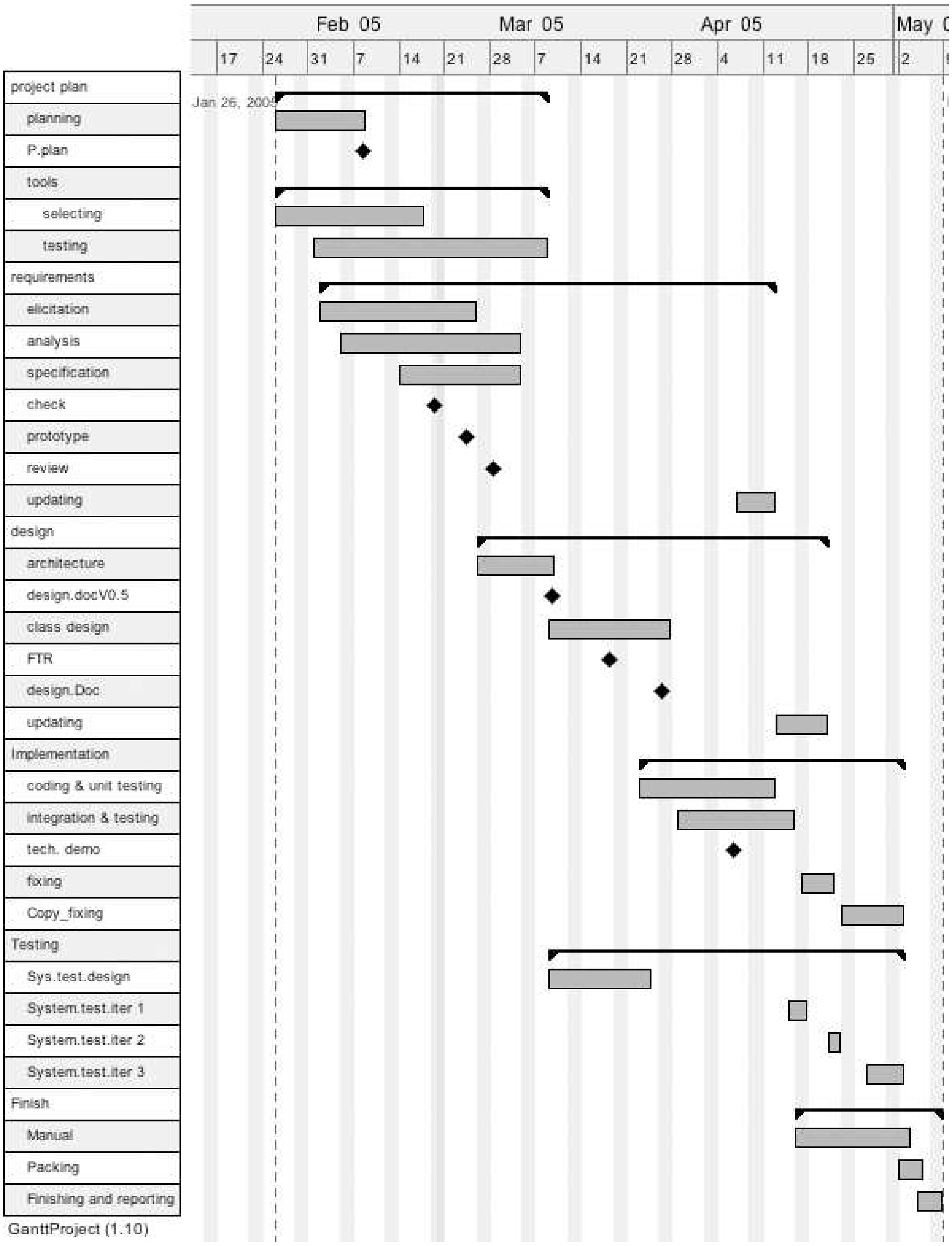

Figure 3: Timetable