

1 Teams

Team Access Barnes, Bhava, Chalasani, Desiraju, Gupta, Ren

Team Consistency Agrawal, Avera, Ding, Ironside, Rice, Smith¹, Zhou

Team Metrics Alapati, Cain, Gibbs¹, Movva, Rao, Silvia, Tillman

Feel free to change the team name, it was only chosen to reflect the required bid for your team. Teams should choose good times to meet and what type of team structure they would like to utilize. The times and places of your meetings should be accessible on WebCT (I may stop by from time to time.)

2 Project

The class as a whole will work on a single project broken up into three parts, one for each team. Each bid will contain the team's assessment of what functionality would be worth an A, B, and C. I will choose among the bids submitted and we will work to refine the bid into a suitable project. A team may bid on as many of the projects as it likes, as long as it submits at least 2 bids, one of which has been assigned.

The overall project will be a tool to support software engineering. The primary functionality will be to track the requirements through the different documents associated with the a software project. The documents will be written in a XML variant that is most appropriate. Initial information about XML can be found at: <http://www.xmlhack.com> and <http://xml.startkabel.nl>

The project descriptions below are very general and high level. They are not meant as specifications of what the project will be, they are only suggestions about what might be done. More specific information will be given in class and in response to questions. You are able to use any tools or libraries that are freely available on Linux, so long as you clearly state that you will be using them.

2.1 Consistency

- maintain a graph of nodes, where each node is a part of a document and the links between the nodes indicate a dependency between those parts
- check the graph for consistency
- allow versioning of nodes
- when a node changes, notify/indicate what other nodes need to be reviewed

¹Distance learning student

- integrate the nodes into different documents, either horizontally or vertically
- allow links from nodes to external resources

2.2 Metrics

- associate with each node various metrics associated with the project, process
- keep a historical database of metrics of past projects
- track risks associated with a project
- keep a historical database of the risks of past projects
- provide a search engine like capability to find related projects
- calculate metrics and risks from the projects found

2.3 Access

- have different categories of users with different associated permissions
- have a permission granting capability, where permission is changed by the creator/authorized user of a node grants access to others
- provide users the ability to see aggregate data but not individual pieces of data
- allow anonymous creation of a node, with different permissions about who is allowed to edit/view the node
- associate with each user various user specific metrics
- allow an anonymous author to maintain their own identifier and receive information anonymously

2.4 GUI

- create a GUI to allow editing of the other projects
- have the GUI be context sensitive providing help and suggestions
- provide a scripting capability to automate tasks or simulate a user
- Computer Supported Collaborative Work
- integrate this tool with other tools such as web browsers, databases, scheduling software, etc