Team Democracy
Project Proposal

1. Roles and Abilities:

Brad - Document Manager - good technical writing, Good at design (esp. Idea Generation)
Connor - Evaluation Manager - Good Programming skills, HTML, CSS, AJAX, Java
Madge - Design Manager - Decent programming experience, Java, Comfortable with backend, has sense of "pretty"
Jeff - Group Manager - Lots of programming experience, PHP, HTML, CSS, Java, JavaScript

2. Problem Statement:

At the moment it can be difficult or tedious to create "project groups" in classes or to group people in other situations. Because of this difficulty, our group intends to create a computerized system to improve this process. It will be targeted for small to medium-sized groups, rather than large groups of hundreds or thousands of people.

Groups can be made for widely varying situations. Our program will need to be flexible enough to accommodate this. The program should automatically create groups based on the constraints entered by the participants. However, because we want this program to be used in a small community like Olin, where organizers have unquantifiable knowledge of the individual participants, the organizer will be able to manually manipulate the groups created by the computer.

3. Characteristics of Primary Users and Their Goals

Our primary users will be the teachers or organizers who need to split participants up into groups. These users currently have varying approaches to the problem. Some may use computer algorithms, while others use index cards. All of them are busy and do not want to spend more time on the task than necessary, especially if it needs to be done on the spot with the participants.

Our secondary users are the participants themselves. They will interact with our program by submitting their information and preferences, probably in the form of a survey. They are also busy people and want the survey to be as simple as possible, although they do want to have as much input in the process as they can, especially if the teams will be long-term.

The main goal for both sets of users is to sort the participants into groups that will generate the highest overall satisfaction for the participants. This can be assessed in many ways. For example, it can be measured in terms of who the participants are working with, what project they are working on, and how large each of the groups is. Different organizers will have different ideas about which measure of participant satisfaction is the most important, so our program must allow for that.

Another goal for the users is to accomplish the sorting in a reasonable amount of time, which varies with the importance of the teams to be formed. If the teams will only be together for a day, the users probably care more about getting the sorting done quickly...
than about making it perfect. On the other hand, if the groups will be together for a year, the users do not mind spending more time to get the groups right.

4. How we will find participants

We have identified several groups of potential users, each of whom have explored some unique aspect of group-creation:

- Resident resources and others involved in Freshman Orientation are tasked with dividing new students into activity groups.
- Employees of the Office of Student Life manage Freshman roommate pairings based on constraints freshmen enter in a short survey.
- Olin Professors regularly divide their students into project groups. Team-making may be done informally among 20 students or so with relatively few constraints, as in ICB or HFID, or it may be a deliberate and lengthy process that takes into account many student preferences, like UOCD or SCOPE team formation.

We will be emailing members we have identified in each of these groups and arranging for short interviews to be conducted. By taking all of the different situations into account, we hope to be able to further define the shape of our project by identifying problems we would like to constrain or expand it to fit.

In addition, we realize that our project may be useful in a myriad of other situations and will be seeking input from interviewees and others on problems likely to benefit from our project.

5. Initial suggested design

We have some preliminary ideas on the services our software will provide. The program will accept the following inputs to determine group choices: number of groups; maximum and minimum size of each group; participants’ preference for groups; and participants’ preference for other members. Information may be added either by a single form, which allows for entering all participants and preferences at once by the organizer, or by surveys sent to each participant. Once the information is entered, the organizer will click a button on the screen which starts the sorting algorithm.

The back end of the software will probably use Gecode/J, a Java wrapper of the C++ constraint satisfaction library Gecode. This library allows for relatively simple representations and solutions of complex optimization problems. It allows solving for only solutions which satisfy all constraints, or solutions which satisfy all hard constraints and an optimal number of soft constraints. If it is necessary for time constraints, we may output a ‘good’ rather than optimal solution.

The preliminary groups will then be displayed, probably on the same page as the data entry. Participant and group names will be highlighted if they break preference or size constraints. Hovering the mouse over the highlighted area will display the specific constraint violated. The organizer will be able to move participants between groups through a click and drag interface. This will change the highlighting appropriately. Additional constraints can be added, and the algorithm rerun to produce different results. When the organizer is satisfied, he or she will be able to export or print the final group listings.