Project
Submission deadline: 2007-10-26

Problem Description ("Advert Consultancy Inc", introduced in Question 5 of Homework 2 and based on example on page 206 of textbook)

Let's call an advertising consultancy company as "Advert Consultancy Inc". The company comprises of general administration (Office manager, Personnel Assistants, Secretaries etc), Accounts (purchasing assistant, account clerks etc.), Creative (Graphics designers, editors, audio/video technicians, photographers etc.) and computing (Network support staff, operators and technicians etc.) departments.

Advert Consultancy Inc. deals with other companies and assists them in their Advertisement campaigns. Such companies are hence forward referred to as Clients. Records are kept for all Clients (currently active and previous). Client records comprise of concerned/representative person or manager from Client Company and other particulars of Clients. Clients have ongoing advertising campaigns, which are coordinated by one representative person from Advert Consultancy Inc. Such person could be from any of the departments depending upon the nature of consultancy required by clients. Such person can potentially work on multiple projects and is answerable to his department manager.

Generally Advert Consultancy Inc only provide consultancy for Creative (creating advertisements, planning an advertisement campaign) and Accounts (feasibility of campaigns, financial consultancy in terms of budget and coverage/effectiveness of campaign) domain. Estimated cost of campaign, starting and finish date and such terms are agreed at both Client and Advert Consultancy before campaign starts. The agreement at Advert Consultancy involves the respective department directors. Now each campaign can have a multiple advertisements, which could be of several types. For instance, newspaper/magazine, internet, TV, radio, leaflet adverts etc.

The actual cost of campaign at "Advert Consultancy Inc" is computed by staff time, resources used, advertising costs, service charges etc. The system also holds salary information. Amount of work by a staff member is calculated by timesheets which they fill out. Currently all this system is done manually.

Consider yourself an Analyst whose task is to model a Software System for this System.

In general the System should have:

1. Details of Client, Details of advertising campaigns active with the respective client.
2. Status and finance related details about Campaign. Details of payment for campaigns, for which we have a separate Accounts System.
3. Details of Staff working on Campaigns. Detail of all the staff in the Advert Consultancy Company. Staff pay-grades, Annual Bonus policy.
4. Advert scheduling information and Staff schedules, so that Directors/Managers can plan resources for future activities.

In general the System should handle this:

1. You do not need a database. It is ok if data is stored in memeory and disappeares when the program is shut down.
2. You should have a graphic user interface.
3. Department Director should be able to see current active campaigns with his department; He/She should be able to check schedules of Campaign, Staff from Advert Consultancy Inc. currently involved in the campaign. He/She should be able to see their other assignments, which will help him to assign them with other clients. Of course he/she also needs to see their skill set before assigning them to a certain task.
4. Department Director uses a gui which requires authorization. It provides a negotiation template (client information, type of campaign needed, resources, and campaign budget information, start and end time etc). Based on such template the Director performs negotiation with Clients. For that he/she needs to do '3'.
5. Once a campaign is negotiated then Client's information (some general information and some specific to Campaign like concerned person's contact information at client) is updated. Campaign might become active immediately, depending upon its negotiated start and end date.
6. Depending upon the type of campaign, nature of expertise required and importantly on availability of staff, Director will assign staff to clients. If he/she needs to assign staff member from another department then he/she has to mark a note to respective department director about his need for particular staff member. Unless the request is acknowledged he/she can't assign such staff member.
7. If a director receives a note from another department director about his need for his department staff then he can perform '3' and also consider his ongoing negotiations. Based on these he can send ACK or refuse the request by giving some comments (for instance about future potential activities of that staff member).
8. Staff also uses the system to see their updates for schedules and new appointments. Moreover every staff member can see how many hours of consultancy he/she has performed. Such has further classification as in-office or out-of-office. In case of out-of-office spent hours, client should be charged extra. All such information is presented by Staff at the end of week using a Timesheet.
9. Staff's skill set is constantly improving depending upon the nature of his tasks.
10. Board of governors need to see a summary of all active campaigns, utilization of staff, and current negotiations in progress.
11. … There could be number of other things which could be done, but we limit our problem to this description only...

Project Description

In this project you are supposed to apply several elements of the Extreme Programming approach (XP) to solving the problem of implementing the system described above. Only a *subset* of XP practices/elements will be used, because of the difficulties with complete modeling an XP process in the framework of a small course project. For more details about XP we refer to the lecture notes and to `http://www.extremeprogramming.org/start.html`. **Remember that you must solve this problem in groups of two or three students. We will not accept any project done by only one student.** XP practices depend on this to be meaningful.

Note that although the problem description is the same as in the the homeworks you must not reuse the work you did during the homeworks, since that did not follow XP. Instead you are supposed to redo the entire development the XP way.

The main elements of XP in this project will be:

1. Developing *user stories*,
2. *Release* planning,
3. *Iteration* planning,
4. Selecting a *system metaphor* that could be suitable for the problem solving,
5. Developing a system in a *test-driven* fashion for selected user stories,
6. *Refactoring* the programs,
7. *Pair programming*.

Project Task

1. Develop a set of user stories (remember that they are *not* use cases and they are much simpler). You should develop at least 20 stories.
2. Make an estimation of the time it takes to implement each user story.
3. Select a set of user stories for the first release. A release is a version of a system that is stable enough and has enough new features to be delivered to end users.
4. Divide the development of the first release into at least 3 iterations. Select stories to be implemented in each of these iterations. When you plan the iterations you should plan for typical XP length of iterations, that is one to four weeks (specify how long you want your iterations to be).
5. Write a metaphor.
6. Perform the planned iterations.
    - Since there is not so much time, you are supposed to program only a little bit of what you planned for each iteration. When programming, your iterations should be eight hours long (neither more nor less).
    - It is very important that you actually finish each iteration. When an iteration is finished there must be no “almost done” stories. The stories of the iteration must be written and successfully tested. It is no disaster if your estimate for the first iteration is wrong and you do not have time to complete all planned stories. In that case you can learn from your misstake when planning the next iteration.
    - Use test-driven programming. First write a test, then the implementation necessary to compile and run the test. You only have to write tests for the model, not for the user interface.
    - Do at least 3 refactorings in each iteration.
    - Use pair programming.

Deliverables

You should deliver the following:

1. A set of developed stories with time estimates.
2. A set of selected stories for the first release. Explain your selection by considering importance and risk factors.
3. Iteration plan, that is a list of which stories you implement in which iteration.
4. The metaphor.
5. Description of your test-driven pair-programming process and applied refactorings. Also describe how well you managed to estimate what should be done in each iteration. Write the truth! In order to pass you must show that you can draw conclusions of your misstakes, not that you have done everything perfectly.
6. The source code and a readme.txt file that explains what is needed to compile and run the program.
7. A comparison of your experience to develop the system with the *requirement-design-implementation* process in the homeworks and with the XP approach (about 1 page of text).

The project must be done in groups of two or three students. We will not accept any project done by only one student.

Bonus Tasks

1. **3 bonus points**
   Write acceptance tests for two of the user stories you implement. You must choose stories that take input and give output through the user interface. The acceptance tests shall be fully automatic, that is they shall give input to the user interface and evaluate the results displayed by the user interface. Report this task by submitting the acceptance tests and a short description of them (including how they are started).
2. **2 bonus points**
   Which more XP elements, except those used in the project, would you like to try in software devolpment? Why would you like to try them and why do you think they would be of help? Write about 1 page of text.

   For a list of XP elements see for example Beck, Kent: *Extreme Programming Explained: Embrace Change, 2nd Ed* (ISBN:978-0321278654)