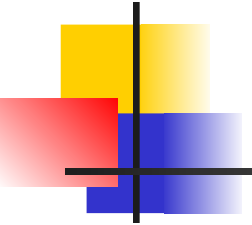


Capstone Projects

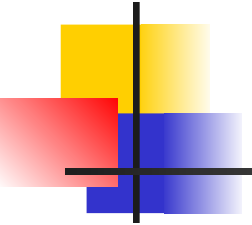
RoviSys (<http://www.rovisys.com>) located in Aurora Ohio, has 4 project that we will help them develop. In the process you will have an opportunity to experience how software is developed in a real company. RoviSys will be represented by Chris Conry, a graduate of this department who suffered through this course while at Kent State.



Previous Capstone Project

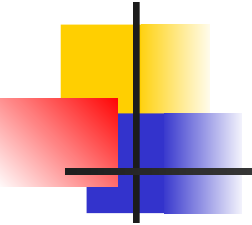
Customer Relationship Management System

RoviSys would also like built a Customer Relationship Management system. This system will allow sales representatives to manage a list of contacts. This will be a web application that has mobile optimized views.



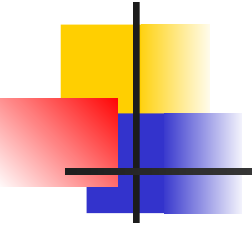
RoviSys Project 1

- Time Tracker
 - Overview
 - For many companies they bill customers based off of time and materials. Tracking an employee's time spent on a project is important to know how much to bill the customer, and where their efforts went.
 - Students will create an application for people to enter their time for Projects (jobs), parts of execution (sub-jobs), and the individual tasks for implementation.
 - Implementation
 - Angular.JS
 - Angular Material
 - Microsoft Web API



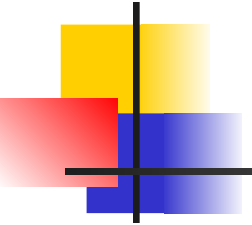
RoviSys Project 2

- Proposal Workflow
 - Overview
 - In today's business environment "time is of the essence" and our customers need fast turnaround on proposal documents that define scope of work, schedule, and pricing based on their needs.
 - We want to build an application that allows us to streamline the process of creating and collaborating on documents, so that we can instead focus on customers.
 - Implementation
 - Angular.JS
 - Angular Material
 - Microsoft Web API



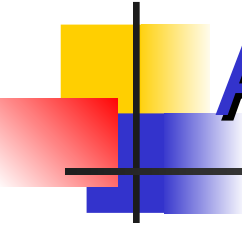
RoviSys Project 3

- Expense Android App
 - Overview
 - Every employees at some time or another will find themselves creating an expense report. Sitting down weeks after and building a reports can be hard to recall every single expense, not to mention sifting through all of the receipts.
 - The goal of this project should be to create an Android application that will allow users to create the expense reports as soon as possible. Taking pictures of receipts and attaching them to the line items.
 - Implementation
 - Android SDK
 - Microsoft Web API



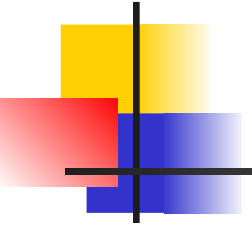
RoviSys Project 4

- Talent Scout
 - Overview
 - Sometimes Stack Overflow is not enough, instead you need to talk to an actual human being for assistance. Strange.
 - We use our Talent scout system to track our experience with different technologies and platforms. So anyone can find out who they need to talk to for assistance.
 - We are looking to expand the application, adding some new features to help increase user participation.
 - Implementation
 - MVC
 - Bootstrap



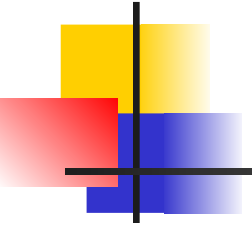
Agile Project Management

- **We will use the Agile Project Development Paradigm to develop our solutions.**
- **Characteristics**
 - Agile methods break tasks into small increments with minimal planning, and do not directly involve long-term planning. *Think top down design.*
 - Each iteration (tri-weekly in our case) involves a team working through a full software development cycle including planning, requirements analysis, design, coding, unit testing, and acceptance testing when a working product is demonstrated to stakeholders. *It is an important part of the Agile to produce a working extension of previous iteration in each iteration.* Weekly team meetings (scrums) at the start of the week will discuss progress and plans for the week.
 - No matter what development disciplines are required, each agile team will contain a customer representative, Chris Conry. He is stakeholders representative and is making a personal commitment to being available to us to answer mid-iteration problem-domain questions. Except when a Chris visits us, I am the stackholders representative. I am also the project director of software development.



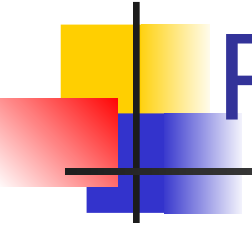
Agile Software Development Paradigm: weekly scrum requirement

- The team will meet(a scrum) during the class lab to jointly access the accomplishments from the prior week and to lay out tasks for the current week. **In particular, note any project stall due to incomplete or non-working code, assigned previously, and report that to the project supervisor (me or Mr. Conry).**
- Each assigned task will be submitted as ticket in TRAC. Completed code will be reviewed and added to the svn and to project mockup(described below).
- In the team scrum in the first lab of an interation, the team will set milestones for the iteration and adjust as necessary the milestones (goals) for the project.
- On the first lab of the project development, **and new team secretary will take over** and the team will set forth the milestones for the project and build the intial mockup.



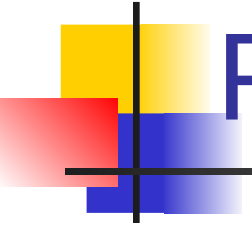
Working Code Requirement

- The team will have an up-to-date mockup of the project code with a web interface working at all times. Initially the mockup can be a single page with **stubs** representing future links or executeables. As the project develops more web pages will be added to the web tree and the stubs will be replace with actual links or executables.
- In each class lab the team will do a code review and add all acceptable code to the demo and to the svn. The demo should be modified only after the new code has been tested on a full copy of the.
- The mockup must be used for iteration demos!
- The complete code for each demo must be stored in a separate folder. That code should not be modified after the demo is given.



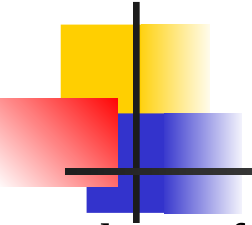
Project Development Tree

- The project development tree is essentially an outline of how the project was developed. The interior nodes of the tree are high level descriptions of what needs to be implemented, and the leaves of the tree represent the code to implement the parent.
- It begins with a node containing the project description as the root of the tree.
- The children (level 1) of the root are the goals set forth in the first iteration (and possibly revised at a later time.)



Project Development Tree

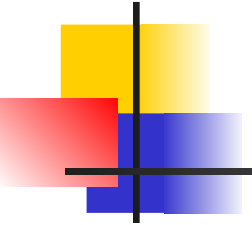
- The nodes at level 2 are the goals/code set forth (and possibly revised at a later time) to describe what must be accomplished to implement the goals given in level 1.
- Etc.



Project Development Tree

Start of Iteration 1

1. Library functions for array data
 - (a) ghost_obj class definition
 - (b) constructor functions
 - i. default constructor
 - ii. copy constructors
 - (c) point transfer functions
 - (d) copy data functions
 - (e) copy with resize functions
 - (f) io functions
 - (g) unit test system // A system to run all the unit tests developed subtree
2. Library functions for graph data
 - (a) constructor functions
 - (b) point transfer functions
 - (c) copy data functions
 - (d) io functions
 - (e) unit test functions



Project Development Tree

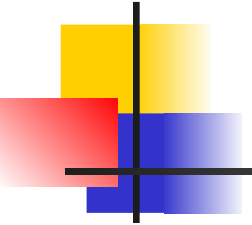
End of Iteration 1

1. Library functions for array data

- (a) [ghost_obj class definition](#) //link to class definition ticket(ticket gives link to code in mockup)
- (b) constructors
 - i. [default constructor functions](#) //link to default constructor ticket
 - ii. [data specific constructor](#)
 - iii. ghost_object copy constructor (requires 1ei)
 - iv. ghost_object with nonuniform data dimensions on cluster cluster nodes
- (c) ghost_obj destructor
- (d) point transfer functions
- (e) copy data functions
 - i. [ghost_obj copy function](#) //link to copy function tickets
 - ii. assignment operator
- (f) copy with resize functions
- (g) io functions
- (h) unit test functions
- (i) array data functions documentation

2. Library functions for graph data

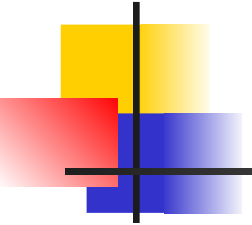
- (a) constructor functions
- (b) point transfer functions
- (c) copy data functions
- (d) io functions
- (e) unit test system
- (f) graph data functions documentation



Tickets

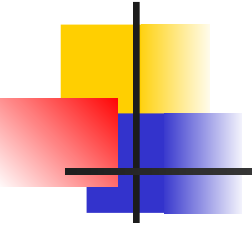
■ When opening:

- Describe what your deliverable will be. Be specific as to what you will produce: ie code or report. Say what the code will do or what the report will give details on.
- Assign it to an iteration goal.
- In the ticket description put a completion date. Note that iteration-long tickets should not be assigned unless they are accompanied by weekly tickets which detail subgoals of the the iteration long ticket.
- All development and documentation tickets must have an accompany verification ticket (assign to a second individual) whose task it is to verify that the code or documentation produced is
 - ➔ Understandable and useful in the case of documentation.
 - ➔ Works as required in the case of code. In this case unit tests for the code is the deliverable.
- It is the group secretaries duty to see that the above conditions are meet.



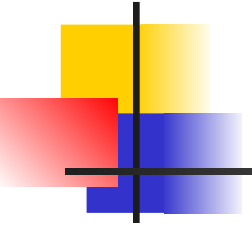
Tickets

- When closing:
 - Put the code or document you produced in the code repository.
 - Reference that code or document with a live link in the ticket and reference the accompanying verification ticket with a live link.
- Points will be deducted from the group secretary's iteration score if tickets do not conform to these requirements.



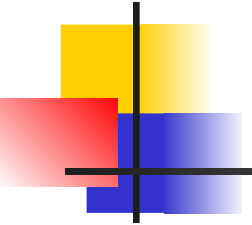
Teams iteration secretaries duties (35% of class grade)

- Weekly Group Planning/Progress Reports by current group secretary summarizing group progress.
 - A list of tickets assigned to each group member and a list of team members who were absent from a planning meeting during the week together with which meetings were missed. Also a list of the code (referenced by ticket) added to the project mockup. Due on by 9:00 am each Monday in the iteration except the last.
 - A final report on the progress made by the group due on the Monday after the end of a iteration by 9:00 am.



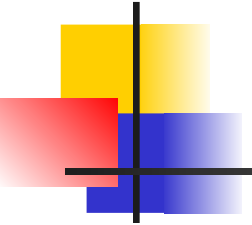
Teams iteration secretaries duties (35% of class grade)

- The written Final Group Progress Reports (due on the Monday after the iteration ends) by current group secretary summarizing group progress, must include:
 - ➔ The prior iteration's project development tree.
 - ➔ The initial iteration goals set by the team.
 - ➔ The tickets complete and added to the project mockup during the iteration.
 - ➔ The project development tree at the end of the iteration with newly added subtrees highlighted.
 - ➔ A summary of the teams accomplishments at the end of the iteration. The summary should highlight the functional features added and reference, when the feature is described, the corresponding, tickets



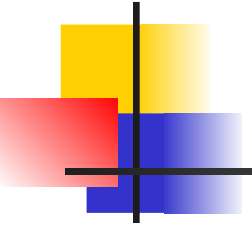
Teams iteration secretary's duties (35% of class grade)

- The Final Group Iteration Report by current group secretary summarizing group progress requirements:
 - In the verbal report: slides (for the stake holder) showing
 - ➔ An appropriate introduction to the project. Do not assume that the stake holder remembers the previous iteration results. Do not assume that everyone in the room knows what the project is.
 - ➔ The state of the project at the beginning of the iteration as shown by the project development tree (see below).
 - ➔ A list of goals for the iteration, and corresponding tickets or milestones with percentages of completion attached to each goal.
 - ➔ The project development tree at the end of the iteration with newly completed branches highlighted.
 - ➔ A demo showing the team's accomplishments using the mockup.
 - The written report (for documentation) detailed in the previous slide above, a list of the tickets assigned to each team member and an indication of its completion, and a narrative **summary of the functional progress** made during the iteration (referencing tickets) as demonstrated by the new code added to the working mockup.



Final Individual Report (45% of class grade)

- The Final Individual Written Report:
 - At the end of the semester, each member of the class will produce a report detailing her/his contributions to the project. The report must include
 - ➔ A list of the closed tickets, that were actually used in the final code or in a demo project or in the project documentation, completed by that individual during the semester. *The report should list separately the tickets for the project documentation the student wrote, and separately list the verification tickets produced.*
 - ➔ A discussion, referencing completed tickets list above, that describes the importance of his/her work to the final state of the project. This discussion is where you describe what you accomplished during the semester. I can list what you learned, providing you produced documentation about how and what your learned. It should also estimate what percent of the final code each of your teammates, including yourself, contributed. The total contributions must at up to 100%.
 - The due date of the individual final reports will be assigned later.



Grading

- Group reports: 35 % of grade.
- Final Individual Report and Final Group presentation (participation in final group demo is required) 45% of grade. A report is required for each team member that should summarize his/her contribution to project (referencing tickets to support your statements) and indicate the relevance of that contribution. This report will be used to assess your contribution to the project.
- Group Grade 10%. How well did your group perform. Regardless of how well your group does, if you are trying to make it work, you'll get a good grade. *Note that inadequate team project documentations will affect this mark.*
- Class participation and attendance 10% of the grade.
- It is possible (although difficult) to get extra credit for efforts significantly above and beyond the course expectations.