Testing and Beginning New Iterations

For this segment of the project you should continue to develop and test new modules within your projects. Be sure to continue to follow the instructions from worksheets 3 and 4 for new design and/or code for your project.

The items listed below are required for this segment of the project. They are not meant to be all encompassing for the work you are expected for this segment.

1. For at least one meeting during this segment of the project, one team member (preferably someone who has knowledge of most of the ongoing issues) should develop an agenda of the items to discuss during a team meeting.
	1. This agenda should be sent to the other team members for approval, review, and additions.
	2. A sample agenda is available at <http://www.projectmanagement-training.net/book/appendix9.html> , but this is not the only way to draft an agenda. Feel free to use your own style. We will cover agendas in class.
	3. Another team member should take minutes (a written record) from the meeting.
	4. Submit your agenda and minutes to the dropbox for Project 4 Milestone 1.
2. Meet with the instructor as a team during office hours or make an appointment to do so. This meeting will serve as both a progress assessment and a time to determine recommended future progress on your project.
3. All teams are near completion for basic functionality of their projects. Complete this functionality before adding new code. While it is fun to develop new functionality and features, it is important to ensure that existing functionality is complete.
	1. Ensure the functionality you provide is complete and not broken.
	2. Ensure via manual and automated testing that your project functions as it should.
	3. Addition of new features for the remainder of the project should be limited.
4. Comment all code for your project. Comments must be informative, but should not be overly verbose. That is, you should explain what each method does, and comment each file, but you should not comment self-descriptive variables or provide comments that would be necessary to teach a novice developer how to write code.
5. Draft a user manual for your code. It is recommended that you use screen shots and explanations and/or create a video showing how the system may be used.
6. Pretend that you are going to deploy your system. Draft deployment instructions if you have not already done so, and update your README file to match.
7. Analyze the amount of time spent on your project using one of the estimation models presented in class. Additionally, perform an “after the fact” assessment based upon your previous time estimates using an effort estimation model of your choice. Each team member must complete this assessment.

Information on Effort Estimation follows below:

<http://userfs.cec.wustl.edu/~cse528/Boehm-SE-Economics.pdf>

<http://en.wikipedia.org/wiki/Function_point>

<http://en.wikipedia.org/wiki/COCOMO>

<http://csse.usc.edu/csse/research/COCOMOII/cocomo_main.html>

<http://pioneer.netserv.chula.ac.th/~sperapho/pub/jack1.pdf>

<http://www.sciencedirect.com/science/article/pii/S0164121297100218>

A couple of fancy Effort Estimation methods follow below:

<http://www.mecs-press.org/ijisa/ijisa-v4-n6/IJISA-V4-N6-2.pdf>

<http://repository.tamu.edu/bitstream/handle/1969.1/ETD-TAMU-2329/ADEKILE-DISSERTATION.pdf?sequence=1>

1. Begin reading about process improvement and risk management models. Some information on risk assessment and process improvement is available at the following sites:

<http://www.pmhut.com/managing-project-risks>

<http://www.csee.wvu.edu/~cukic/CS430/Risk.pdf>

<http://www.sei.cmu.edu/reports/96tr012.pdf>

<http://en.wikipedia.org/wiki/Capability_Maturity_Model>

<http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration>

<http://www.sei.cmu.edu/reports/10tr033.pdf>

<http://en.wikipedia.org/wiki/ISO_9000>

<http://www.computer.org/portal/web/swebok>

<http://en.wikipedia.org/wiki/Verification_and_validation>

<http://en.wikipedia.org/wiki/Verification_and_validation_%28software%29>