Introduction

The purpose of this homework is to give you a chance to think about some of the topics presented in class related to project development and relate them to your own experiences on the project teams or elsewhere.

Assignment

Select at least one of the readings listed below. Write a short paper that accomplishes the following goals.

1. Summarize the main points of the paper as expressed by the author.
2. Identify two or more issues that you believe are important in the paper from your point of view. You can select issues where you agree or where you disagree with the author's opinion on the topic, either way is okay.
3. Express your opinion on these issues and show why you agree or disagree. Cite *specific statements* from the paper and use *specific examples* from your experience (in this class or during any other development experience) to illustrate the issues and justify your arguments.

The paper should be about 1500 to 2500 words long (3-5 pages). However, length is not the critical measure; the task is to do the reading, think about it, and write something interesting about it.

Turnin

This is an individual assignment, not part of the project development. Your paper is due Wednesday, May 19, at the start of class.

Readings

Rapid Development, Steve McConnell
1 - Introduction
2 - Rapid-development strategy
4 - Software Development Fundamentals
5 - Risk management
7 - Lifecycle planning
10 - Customer-Oriented Development
12 - Teamwork
13 - Team Structure
18 - Daily Build and Smoke Test
19 - Designing for Change

Anchoring the Software Process, Barry Boehm,
http://citeseer.nj.nec.com/boehm95anchoring.html

Implementing VisiCalc, Bob Frankston
http://www.frankston.com/public/writing.asp?name=ImplementingVisiCalc

Painless Functional Specifications, Joel Spolsky
http://www.joelonsoftware.com/articles/fog0000000036.html

The Joel Test: 12 Steps to Better Code, Joel Spolsky
http://www.joelonsoftware.com/printerFriendly/articles/fog0000000043.html

The Mythical Man-Month, Brooks
Chapter 3, The Surgical Team

The Pragmatic Programmer, Hunt & Thomas
Chapter 7, Before the Project

Structuring Use Cases with Goals , A. Cockburn
http://alistair.cockburn.us/crystal/articles/alistairsarticles.htm

Use cases in theory and practice, A. Cockburn
http://alistair.cockburn.us/crystal/articles/alistairsarticles.htm

Programming Considered as a Human Activity, EW Dijkstra, Proceedings of the IFIP Congress 65

On the Criteria To Be Used in Decomposing Systems into Modules, DL Parnas, Comm. of the ACM, V15, No 12, Dec 1972
http://www.acm.org/classics/

The Hundred-Year Language, Paul Graham
http://www.paulgraham.com/hundred.html

Perfection and Simplicity, *Taste and Aesthetics*, and *Designing Distributed Systems*, from A Conversation with Ken Arnold, by Bill Venners
http://www.artima.com/intv/perfect.html
http://www.artima.com/intv/taste.html
http://www.artima.com/intv/distrib.html

Software Architecture, David Garlan, CMU, 2001
http://www-2.cs.cmu.edu/~able/publications/encycSE2001/

A Practical Method for Documenting Software Architectures, Clements, et al, CMU
http://www-2.cs.cmu.edu/~able/publications/icse03-dsa/

If You Didn’t Test It, It Doesn’t Work, Bob Colwell, IEEE Computer
http://www.computer.org/computer/homepage/0502/Random/