Software Engineering  
CS5704: Class 5 - 2/16/01

Instructor: Shawn A. Bohner  
Voice: (703) 538-8374  
Email: bohner@nvc.cs.vt.edu

Teaching Assistant: Sepna Georges  
Voice: (703) 538-8381  
Email: sgeorge@vt.edu

Agenda

▲ Review Last Week’s Material
  ● Turn in Homework
  ● Discussion

▲ Software Project Feedback
  ● Break

▲ Chapter 9 – Software Configuration Management

▲ Class Project Discussion

▲ Homework and Project Assignment
Spring Semester Timeline

Class Begins
Product & Process

PM Metrics & Estimation

Analysis, Design, & Architecture
Mid-Term Exam

SW Metrics & Testing Strategies

Maintenance & Evolution
Final Exam

Jan — Feb — Mar — Apr — May

Project Management
Cross-Life-Cycle Process
Testing Techniques
Object-Oriented Development
Advanced SWE Topics

12 weeks, 9 sessions to go... Midterm Exam on March 2\textsuperscript{nd}! So much to do and so little time...

Problem 7.1

▲ Objective: Deal with Unrealistic Deadlines

▲ Document your reservations using quantitative arguments derived from past project metrics. See bottom of page 167.

▲ Then suggest an incremental approach (see Chapter 2) that offers to deliver partial functionality in the time allotted with complete functionality following.
Problem 7.2
▲ Objective: Difference between Macroscopic and Detailed Schedule. Is a macroscopic schedule enough?
▲ A macroscopic schedule defines major activities and tasks. It is the table of contents for the project and may be sufficient for small efforts. For larger projects, a detailed schedule, defining all work tasks, work products, and milestones should be developed. Otherwise, devil in the details syndrome sets in!

Problem 7.4
▲ Objective: Develop counter-example of communications overhead discussion in SEPA 7.2.1.
▲ The reduction of rework is the key to your argument. If good reviews reduce rework significantly, it can be argued that the time spent on communication (during reviews) results in time savings (due to less rework) overall. The crux of this argument is as follows: Typically, it costs more than 10 times as much to find and correct an error during testing as it does to find and correct the same error during, say, design. Therefore, if one hour of communication finds one error during design, it saves 10 hours of work later on. Big payoff!
Problem 7.12

▲ Objective: Detail Technology Risk Assessment

Task 1.3 Technology Risk Assessment

1.3.1 Identify Key Technology Risks
   1.3.1.1 Create Risk Item Checklist
   EndTask 1.3.1

1.3.2 For each risk; Determine Category, Probability of Occurrence, and Impact; endfor

1.3.3 Construct Initial Risk Table

1.3.4 FTR: Review Risks with Team

1.3.5 Develop RMMM for Identified Risks
   1.3.5.1 For each risk; Determine Mitigation, Monitoring, and Management opportunities; endfor
   EndTask 1.3.5

1.3.6 FTR: Review Risks with Customer

1.3.7 Finalize Technology Risk Assessment Numbers with a Risk Information Sheet.

EndTask 1.3

Problem 7.13

▲ Objective: Compute Earned Value Numbers

▲ Use the steps defined in Section 7.8 to compute earned value. First sum all planned effort through task 12.

▲ Budgeted Cost of Work Scheduled BCWS = 156.50
   Budget at Completion BAC = 582
   Budgeted Cost of Work Performed BCWP = 126.50
   Actual Cost of Work Performed ACWP = 127.50

   SPI = BCWP/BCWS = 126.5/156.5 = 0.81
   SV = BCWP - BCWS = 126.5 - 156.5 = -30.0 person-day
   CPI = BCWP/ACWP = 126.5/127.5 = 0.99
   CV = BCWP - ACWP = 126.5 - 127.5 = -1.0 person-day

   % scheduled for completion = BCWS/BAC = 26.9%
   % complete = BCWP/BAC = 21.9%
Problem 8.1

▲ Objective: Program variations and how to control them
▲ We assess the variation in the form and content of work products
▲ We look for variation in:
  ● Traceability from requirements to design and design to code
  ● The software process — repeatable process is a goal
  ● Expected and actual results derived from software testing

Problem 8.2

▲ Objective: Possibility to assess the quality if customer keeps changing function
▲ If we define quality as "conformance to requirements," and requirements are dynamic (keep changing), the definition of quality will be dynamic and an assessment of quality will be difficult.
▲ While it is possible to assess the quality, it can only be for some point in time and the likelihood is that the quality will be assessed poorly.
Problem 8.3

▲ Objective: Discuss quality and reliability

▲ Quality focuses on the software's conformance to explicit and implicit requirements.

▲ Reliability focuses on the ability of software to function correctly as a function of time or some other quantity. Safety considers the risks associated with failure of a computer-based system that is controlled by software.

▲ In most cases an assessment of quality considers many factors that are qualitative in nature (i.e., “ilities”).

▲ Assessment of reliability and to some extent safety is more quantitative, relying on statistical models of past events that are coupled with software characteristics in an attempt to predict future operation of a program.

Problem 8.4

▲ Objective: Can a program be correct but not reliable?

▲ Yes. It is possible for a program to conform to all explicit functional and performance requirements at a given instant, yet have errors that cause degradation that ultimately causes the program to fail.
Problem 8.8

▲ Objective: Countable characteristics of software that imply quality (beside errors)

▲ Any countable measure that indicates the factors noted in Chapter 8 are candidates. For example, maintainability as measured by mean-time-to-change; portability as measured by an index that indicates conformance to language standard; complexity as measured by McCabe's metric, availability, reliability, and so on.

Problem 8.12

▲ Objective: Calculate Error Index

- Total = 942 (128 Serious, 379 Moderate, 435 Trivial)
- P1 20% = 188 (26 Serious, 76 Moderate, 87 Trivial)
- P2 15% = 141 (19 Serious, 57 Moderate, 65 Trivial)
- P3 15% = 141 (19 Serious, 57 Moderate, 65 Trivial)
- P4 40% = 377 (51 Serious, 152 Moderate, 174 Trivial)
- P5 10% = 95 (13 Serious, 38 Moderate, 44 Trivial)

Ws=10, Wm=3, Wt=1 ➔ Pli = 10(Si/Ei) + 3(Mi/Ei) + 1(Ti/Ei)

- PI1 = 10*(26/188) + 3*(76/188) + (87/188) = 3.06
- PI2 = 10*(19/141) + 3*(57/141) + (65/141) = 3.02
- PI3 = 10*(19/141) + 3*(57/141) + (65/141) = 3.02
- PI4 = 10*(51/377) + 3*(152/377) + (174/377) = 3.02
- PI5 = 10*(13/95) + 3*(38/95) + (44/95) = 3.03

EI = (PI1 + 2*PI2 + 3*PI3 + 4*PI4 + 5*PI5)/PS
- EI = (3.06 + 3.02 + 3.02 + 3.02 + 3.03)/100000 = 0.000454
Why Are Projects Late?

▲ Unrealistic deadline
▲ Changing customer requirements
▲ Honest underestimates
▲ Predictable and/or unpredictable risks
▲ Technical difficulties
▲ Human difficulties
▲ Miscommunication
▲ Failure by project management

What are some examples of the above?

Project Delivery Success

<table>
<thead>
<tr>
<th></th>
<th>Early</th>
<th>On-Time</th>
<th>Delayed</th>
<th>Canceled</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 FP</td>
<td>14.68%</td>
<td>83.16%</td>
<td>1.92%</td>
<td>0.25%</td>
<td>100.00%</td>
</tr>
<tr>
<td>10FP</td>
<td>11.08%</td>
<td>81.25%</td>
<td>5.67%</td>
<td>2.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>100FP</td>
<td>6.06%</td>
<td>74.77%</td>
<td>11.83%</td>
<td>7.33%</td>
<td>100.00%</td>
</tr>
<tr>
<td>1000FP</td>
<td>1.24%</td>
<td>60.76%</td>
<td>17.67%</td>
<td>20.33%</td>
<td>100.00%</td>
</tr>
<tr>
<td>10,000FP</td>
<td>0.14%</td>
<td>28.03%</td>
<td>23.83%</td>
<td>48.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>100,000FP</td>
<td>0.00%</td>
<td>13.67%</td>
<td>21.33%</td>
<td>65.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>Average</td>
<td>5.53%</td>
<td>56.94%</td>
<td>13.71%</td>
<td>23.82%</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

From Capers Jones, Patterns of Software Systems Failure and Success (International Thomson Computer Press, 1996)

How does size impact delivery success? Why?
How does uncertainty and risks interact with size for delivery?
Define a Task Network

Three I.3 tasks are applied in parallel to 3 different concept functions

How does process and functional decomposition results get realized in the Task Network?
What type of project is this network reflecting?

Derive a Timeline Chart (AKA Gantt)

<table>
<thead>
<tr>
<th>Work tasks</th>
<th>week 1</th>
<th>week 2</th>
<th>week 3</th>
<th>week 4</th>
<th>week 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.1.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I.1.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I.1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I.1.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I.1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I.1.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I.1.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I.1.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How does a milestone differ from a task?
What is a work breakdown structure (WBS)?
Earned Value

▲ What is Earned Value used for?
▲ What is CPI?
▲ What is SPI?
▲ What is BAC
▲ How does BCWP relate to BCWS?

Why SQA Activities Pay Off?

Cost to find and fix a defect

log scale

0.75 1.00 1.50 3.00 10.00

60.00-100.00

0.75 1.00 1.50 3.00 10.00

Field Use

Test System

Test

Code

Design

Reqts.

How does rework result in the costs reflected here?

Explain how quality improves productivity.
Quality Concepts

▲ What is Quality Control?
▲ How is Quality Assurance different?
▲ What are the primary drivers in Cost of Quality?

Software Testing for Quality Control

Acceptance
- Plan
- Develop
- Execute

System
- Plan
- Develop
- Execute

Integration Testing
- Plan
- Develop
- Execute

Unit Testing
- Plan
- Develop
- Execute

What is the difference between Acceptance Test and System Test?
Why is unit test associated with Program or Detailed Design?
Describe Integration Testing in the context of Architecture?
What Are Reviews?

▲ Meeting conducted by who for who
▲ Technical assessment of ???
▲ Software quality ??? mechanism
▲ Bottom Line?!?

Review Options Matrix

<table>
<thead>
<tr>
<th></th>
<th>IPR</th>
<th>WT</th>
<th>IN</th>
<th>RRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trained leader</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Agenda established</td>
<td>maybe</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Reviewers prepare in advance</td>
<td>maybe</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Producer presents product</td>
<td>maybe</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>“Reader” presents product</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Recorder takes notes</td>
<td>maybe</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Checklists used to find errors</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Errors categorized as found</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Issues list created</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Team must sign-off on result</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>maybe</td>
</tr>
</tbody>
</table>

IPR—informal peer review  WT—Walkthrough  IN—Inspection  RRR—round robin review

How does rigor play into Reviews?
Describe an inspection and its purpose. Example?
Chapter 9
Software Configuration Management

Purpose

▲ The purpose of this chapter is to provide an overview of issues associated with software configuration management, version control, and change control
  ● Software change impacts in the context of demand and release management
  ● Automation to manage details

▲ Objectives
  ● Outline key software configuration management principles
  ● Examine key aspects of software change
The “First Law”

No matter where you are in the system life cycle, the system will change, and the desire to change it will persist throughout the life cycle. Bersoff, et al, 1980
Software is Suppose to Change

▲ Software is part of a computer system that is intended to change
▲ Business changes and its computer systems must respond
▲ Software is engineered, not manufactured
▲ Software is intangible
▲ Software is complex

Software is suppose to change… otherwise it would in the hardware!

Software Doesn’t Wear Out

▲ Software doesn’t change with age or “wear out” with use! However, ...
  ● Software “ages” or becomes “obsolete” with a changing environment
  ● Software deteriorates or “degrades” with continued changes
Software Design Degradation

*The Original Software Design...*  
*...Plus a few “Changes”*

- Easy to Understand
- Components well isolated to facilitate change
- Isolation supports change validation

- Increased size and complexity
- ...but it works (for awhile)
- Reliability of system degrades, errors creep in
- At some point, it’s unmaintainable
- ...effort to make the next change becomes prohibitive

Information Lose Due to Relentless Change

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Change Set 1</th>
<th>Change Set 2</th>
<th>Change Set N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spec's</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Change Management and SCM

▲ Changes starts somewhere
  ● Change Requests
  ● Trouble calls ...
▲ Must be associated with the software configuration and managed through the change cycle

What are the Changes?

- Changes in Business Requirements
- Changes in Technical Requirements
- Changes in User Requirements

Software Models

Project Plan

Test

Code

Data

Other Documents
The Software Configuration

Flow of SCM: Definition, Use, Archive

SCM Staff
- Defines SCM Structures, Standards and Procedures

Software Project Staff
- Uses SCM Library Structures, Standards and Procedures

SCM Staff
- Archives/Controls Baseline Software

Key Software Baseline Versions
- Archives of System Software
Change & SCM

Software Engineering

- tools
- methods
- procedures
- a TQM foundation

SCM

- Identification
- Version Control
- Change Control
- Auditing
- Reporting
- Construction

Change Control

STOP
Change Control Process—I

Need for Change is Recognized
   ↓
Change from User
   ↓
Developer Evaluates
   ↓
Change Report is Generated
   ↓
Change Control Authority Decides
   ↓
Request Is Queued for Action
   ↓
Change Request is Denied
   ↓
User is Informed

Change Control Process—II

Change Control Process—III
Change Control Process-III

- Perform SQA and Testing Activities
- Check-in the Changed SCIs
- Promote SCI for Inclusion in Next Release
- Rebuild Appropriate Version
- Review/Audit the Change
- Include All Changes in Release

Auditing
Status Accounting

- Change Requests
- Change Reports
- SCIs
- ECOs

Reporting

Context: Networking Dog Breeders and Owners who want to find eligible dogs for breeding (dates).

Purpose: To facilitate the connection between potential dogs (between owners/breeders) with effective information that enables selection and monitoring of dog breeding opportunities.

Doggie Dating System

- Doggie Data, Inquiries, Customers
- Breeds Guidelines
- Mating Criteria
- Client Guidelines
- Doggie Dates, Potential Matches
- Breeds Database

Dog E-Dating System
Doggie Dating Statement of Scope

▲ The Dog E-Dating System (DDS) will accept information from clients and potential clients that supports matching of eligible dogs for breeding. DDS will interact with clients (people) through a web interface both for data entry and business transactions. There must be a database maintained on a web connected personal computer for dog information and client information to be entered, updated, and removed.

▲ The DDS will accept dog information, perform initial classification of dogs, analyze potential matches, and report on potential matches according to breed and client criteria. The information will be accurate according to data entered. The reliability, availability, and maintainability performance of the software will be agreed upon once the price has been established. The system will interface with people via the workstation and via the web.

Homework Assignment for 2/23/01

▲ Read Pressman Chapters
   ● 10.1-10.7
     – Do problems 10.1, 10.2, 10.5, 10.6, 10.9
   ● 11.1-11.7
     – Do problems 11.4, 11.7, 11.9

▲ Continue Refining Project Plan for “Dog E-Dating System” New Name…
   ● Based on my basic Statement of Scope
   ● Derive additional requirements
   ● Provide initial estimates using COCOMOII from Macroscopic perspective

▲ Have a great week!