Software Engineering

CS5704: Class 4 - 2/9/01

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
▲ Chapter 7 – Project Scheduling & Tracking
  ● Break
▲ Chapter 8 – Software Quality Assurance
▲ Class Project Discussion
▲ Homework and Project Assignment
Spring Semester Timeline

Class Begins
Product & Process

PM Metrics & Estimation
Mid-Term Exam

Analysis, Design, & Architecture

SW Metrics & Testing Strategies

Maintenance & Evolution

Jan — Feb — Mar — Apr — May

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

13 weeks, 10 sessions to go… Midterm Exam on March 2nd! So much to do and so little time…

Problem 5.1

▲ Objective: to complete a simple statement of scope for home security system
   ● Bounded software scope – data/control, primary functions, performance, and interfaces
▲ Process data/information from user, alarms, and sensors
▲ Interact with user from control panel (for status and password), configure system, store configurations, display functions and status, activate / reset alarm, and monitor sensors
▲ Sensors monitored every 2 seconds, alarm activated within 1 second of event detection
▲ Interface with control panel, sensors, and alarm
Problem 5.2

▲ Sometimes, complexity arises from a poorly established interface between the customer and the software developer. Discounting that, the following technical characteristics should be considered:
  ● real-time attributes
  ● multiprocessing requirement
  ● (concurrency)
  ● nature of the algorithm
  ● requirement for recursion
  ● nature of input
  ● determinacy of input
  ● nature of output
  ● language characteristics
  ● . . . and knowledge/experience of staff on application.

Problem 5.3

▲ Objective: to outline performance in different application domains
  ● Real time application—raw processing of data measured by CPU time and possibly interrupt servicing efficiency
  ● Engineering/scientific applications—numerical accuracy and for large systems, CPU time.
  ● Commercial applications—I/O efficiency
  ● Interactive applications—user "wait time" and application availability
  ● Microprocessor applications—CPU time and memory requirement
Problem 5.4

▲ Objective: do a simple functional decomposition of SafeHome

- user interaction (2400)
- sensor monitoring (1100)
- message display (850)
- system configuration (1200)
- system control [activation/deactivation] (400)
- password processing (500)
- LOC estimates (in the C language) for each function are noted in parentheses. A total of 6450 LOC are estimated. Using the data noted in the problem:
  6450 LOC / 450 LOC/pm = 14.3 pm
- Cost: 14.3 pm * $7,000/pm = $100,000 (approx)

Problem 5.6

▲ Objective: Estimate effort for simple ATM using application composition model

- 12 screens X ((1+2+3)/3) = 24 NOPs
- 10 reports X ((2+5+8)/3) = 50 NOPs
- 80 components X ((0+0+10)/3) = 264 NOPs
- NOP = 338 total NOPs
- PROD = NOP/PM = (4+7+13+25+50)/5
  = 19.8 NOPs / person month
- Estimated Effort = NOP/PROD
  = 338 NOPs / 19.8 NOPs / person month
  = 17.1 person months
Problem 6.1

▲ Objective: 5 examples of reactive risk strategy
- Starting with coding since there will be so much testing to do
- Testing components after a defects are reported
- Putting a stop sign at a dangerous corner only after a fatal accident has occurred
- Fixing a pothole only after the city has been sued by an angry motorist
- Only good thing about not planning is that failure is a complete surprise – not preceded by extended times of angst!

Problem 6.2

▲ Objective: Difference between known and predictable risks
▲ Known risks are those that are determine through careful evaluation of the project and technology.
▲ Predictable risks are extrapolated from past experience.
Problem 6.3

▲ Objective: 3 additional questions for each risk area – a think about it question…
▲ Product Size
▲ Business Impact
▲ Customer
▲ Process Maturity
▲ Technology
▲ Staffing/People

Problem 6.4

▲ Objective: Low cost video editing system technical risks
  ● Rapid changes in digital format for video data
  ● Changing compression algorithms and format
  ● Rapid changes in processing power and bus architectures
  ● Rapid changes in video input modes (e.g., via internet, direct from camera, across LAN, from analog tape, from DAT)
Problem 6.5

▲ Business risks here are probably the most relevant, although you may be tempted to select technical and project risks.
▲ Large market share of existing WPs with significant penetration problems
  ● Category – BU Probability – 35% Impact - 1
▲ Industry standard WPs will cause reticence to buy
  ● Category – BU Probability – 15% Impact - 2
▲ Timeline tight due to high staff turnover and limited staffing profile
  ● Category – PS Probability – 55% Impact - 2

Problem 6.12

▲ Objective: Recalculate Risk Exposure
  ● Risk Exposure = Probability X Impact
  ● Probability = 60%
  ● Impact =
    18 comp. X 100 LOC/comp. X $16/LOC
    = $28,800
▲ Risk Exposure = .60 X $28,800 = $17,280
Cost Estimation
▲ Project scope must be explicitly defined
▲ Task and/or functional decomposition is necessary
▲ Historical measures (metrics) are very helpful
▲ To assure fit, use two or more estimation techniques
▲ Uncertainty is inherent in most estimation endeavors – plan on it!

How does process and functional decomposition impact estimation?
Explain uncertainty with respect to software estimates.

Estimation Techniques
▲ Based on past (similar) project experience
▲ Conventional estimation techniques
  ● Task breakdown and effort estimates (e.g., WBS)
  ● Size (e.g., FP or LOC) estimates
▲ Tools (e.g., COCOMOII, SLIM, Checkpoint)

When do you use an estimation method based on project experience?
Is “size” important for conventional estimation techniques? Why?
How does a functional decomposition come from a scope statement?
What is meant by grammatical parse?

Example: FP Approach

<table>
<thead>
<tr>
<th>measurement parameter</th>
<th>count</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of user inputs</td>
<td>40</td>
<td>4</td>
</tr>
<tr>
<td>number of user outputs</td>
<td>25</td>
<td>5</td>
</tr>
<tr>
<td>number of user inquiries</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>number of files</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>number of ext.interfaces</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>algorithms</td>
<td>60</td>
<td>3</td>
</tr>
<tr>
<td>count-total</td>
<td></td>
<td></td>
</tr>
<tr>
<td>complexity multiplier</td>
<td></td>
<td></td>
</tr>
<tr>
<td>feature points</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Productivity Factor

\[
0.25 \text{ p-m / FP} = 120 \text{ p-m}
\]

What is a function point? A feature point? How does a function point relate to a software estimate?
Parametric Estimates

▲ A parametric cost estimating relates the cost of a system to one or more parameters (variables) of the system such as physical or performance characteristics

▲ Attributes
  ● Moderate to high level of detail
  ● Based on actual costs of many systems
  ● Uses system parameters to estimate costs

▲ What are some of the advantages?
▲ Disadvantages?

Analogy Estimating

▲ Estimating by Analogy relates the cost of a system to the cost of a known similar system through comparisons of such items as complexity, technical characteristics, and producibility

▲ Attributes
  ● Direct relationship to costs for a similar item
  ● May be adjusted by subjective factors
  ● Requires engineering judgment to quantify differences between proposed system and the analogue system

▲ What are some of the advantages?
▲ Disadvantages?
Bottom-Up Estimating

▲ Bottom-up (grass roots) estimating builds up an estimate of a system using detailed information
▲ Attributes
  ● Low level of detail
  ● Based on detailed specifications
▲ What are some of the advantages?
▲ Disadvantages?

Projects are about Risk

▲ Risk translates to:
  ● Lack of ?
  ● Lack of ?, and/or
  ● Lack of ?
▲ Risk = F(? X ?)

Must Effectively Balance Technology Risk and Opportunity

Banker’s Risk
Investor’s Risk
Gambler’s Risk
How does one identify risks?
What is necessary to do for risk planning?
How does that effect tracking?

Risk Mitigation, Monitoring, and Management (RMMM)

▲ Mitigation—what is it?
▲ Monitoring—what is it?
▲ Management—what is it?
Barry Boehm’s Top 3 Software Risks?

- Shortfalls in externally-furnished components
- Developing the wrong user interface
- Developing the wrong software functions
- Gold-plating. Requirements scrubbing
- Straining computer science capabilities
- Personnel Shortfalls
- Continuing stream of requirements changes
- Unrealistic Schedules and Budgets
- Shortfalls in externally-performed tasks
- Real-time performance shortfalls.

Chapter 7
Project Scheduling and Tracking
Purpose

▲ The purpose of this chapter is to describe many of the issues associated with building and monitoring schedules for software projects

● Proactively deal with unrealistic customer deadlines (based on detailed estimates)
● To answer the question “So what?”

▲ Objectives

● Outline key software project schedule and monitoring principles
● Understand work breakdown and tasking
● Examine software work tracking techniques

Why Are Projects Late?

▲ Unrealistic deadline established by someone outside the software development group
▲ Changing customer requirements that are not reflected in schedule changes
▲ Honest underestimate of the amount of effort and/or the number of resources that will be required to do the job
▲ Predictable and/or unpredictable risks that were not considered when the project commenced
▲ Technical difficulties that could not have been foreseen in advance
▲ Human difficulties that could not have been foreseen in advance
▲ Miscommunication among project staff that results in delays
▲ Failure by project management to recognize that the project is falling behind schedule and a lack of action to correct the problem
### 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%</td>
</tr>
<tr>
<td>10FP</td>
<td>11.08%</td>
<td>81.25%</td>
<td>5.67%</td>
<td>2.00%</td>
<td>100%</td>
</tr>
<tr>
<td>100FP</td>
<td>6.06%</td>
<td>74.77%</td>
<td>11.83%</td>
<td>7.33%</td>
<td>100%</td>
</tr>
<tr>
<td>1000FP</td>
<td>1.24%</td>
<td>60.76%</td>
<td>17.67%</td>
<td>20.33%</td>
<td>100%</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%</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%</td>
</tr>
<tr>
<td>Average</td>
<td>5.53%</td>
<td>56.94%</td>
<td>13.71%</td>
<td>23.82%</td>
<td>100%</td>
</tr>
</tbody>
</table>


### Relationship Between People and Effort

▲ Arbitrarily adding people to a project does not reduce project completion time

● Could lengthen the completion time

▲ Sometimes a project schedule has slipped so badly that the only option is to renegotiate the completion date with the customer
Key Scheduling Principles

▲ Compartmentalization—define distinct tasks
▲ Interdependency—indicate task interrelationships\textit{force validation—be sure resources are available}
▲ Defined responsibilities—people must be assigned
▲ Defined outcomes—each task must have an output
▲ Defined milestones—review for quality

Defining Task Sets

▲ Task set - a collection of engineering tasks, milestones, and deliverables
  ● Process model provides guidance in determining the task set
▲ Determine type of project
▲ Assess degree of rigor required
  ● Identify adaptation criteria
  ● Compute task set selector value
  ● Interpret TSS to determine degree of rigor
▲ Select appropriate software engineering tasks
Example from Book

Figure 7.2 Concept development tasks using an evolutionary model

Define a Task Network

Three L3 tasks are applied in parallel to 3 different concept functions

Three L3 tasks are applied in parallel to 3 different concept functions
But what about big projects?

▲ Automated scheduling tools are essential for software projects of any real size
▲ PERT (program evaluation and review technique) and CPM (critical path method) determine the critical path and compute minimum project completion time
▲ Timeline (Gantt) charts – mapping the work breakdown structure (WBS) to a timeline
▲ Time-boxing – working in reverse from deadline

<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>Identify need and benefits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meet with customers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identify needs and project constraints</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Establish product statement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Milestone: product statement defined</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>Define desired output/control/input (OCI)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scope keyboard functions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scope voice input functions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scope modes of interaction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scope document diagnostics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scope other WP functions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document (OCI)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTR: Review OCI with customer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revise OCI as required</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Milestone: OCI defined</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>Define the functionality/behavior</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Define keyboard functions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Define voice input functions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Describe modes of interaction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Describe spell/grammar check</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Describe other WP functions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTR: Review OCI definition with customer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revise as required</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Milestone: OCI definition complete</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>Isolate software elements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Milestone: Software elements defined</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>Research availability of existing software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Research text editing components</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Research voice input components</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Research file management components</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Research Spell/ Grammar check components</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Milestone: Reusable components identified</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>Define technical feasibility</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluate voice input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluate grammar checking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Milestone: Technical feasibility assessed</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>Make quick estimate of size</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>
<tr>
<td>Create a Scope Definition</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Review scope document with customer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revise document as required</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Milestone: Scope document complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Software Development Effort Allocation

40-50%

▲ “Front end” activities
- customer communication
- analysis
- design
- review and modification

15-20%

▲ Construction activities
- coding or code generation

30-40%

▲ Testing and installation
- unit, integration
- white-box, black box
- regression

Earned Value

▲ Quantitative technique for monitoring project completion to date and assessing progress from a value perspective

▲ Estimate total project completion time then compute the percentage of the total project time associated with each project task

▲ www.acq.osd.mil/pm/
Chapter 8
Software Quality Assurance

Purpose

▲ The purpose of this chapter is to introduce software quality assurance (SQA) as a software discipline
  ● Software quality work begins before testing and continues after the software is delivered
  ● Metrics in software management is reinforced

▲ Objectives
  ● Outline software quality assurance principles
  ● Examine key aspects of software quality
Why SQA Activities Pay Off?

Cost to find and fix a defect

log scale

1
10
100

0.75
1.00
1.50
3.00
10.00
60.00-100.00

Reqs.
Design
Code
Test
System
Test
Field
Use

Quality Concepts

▲ between samples”
...But how does it apply to software?
● Maintainability
● Low Defect/Failure Rates
▲ Quality Control: a series of inspections, reviews, tests
▲ Quality Assurance: analysis, auditing and reporting activities
▲ Cost of Quality
● Appraisal costs
● Failure costs
● External failure costs
Software Testing for Quality Control

- Acceptance
  - Plan
  - Develop
  - Execute
- System
  - Plan
  - Develop
  - Execute
- Integration Testing
  - Plan
  - Develop
  - Execute
- Unit Testing
  - Plan
  - Develop
  - Execute

Software Quality Assurance

- Process Definition & Standards
- Formal Technical Reviews
- Analysis & Reporting
- Test Planning & Review
- Measurement
What Are Reviews?

▲ Meeting conducted by technical people for technical people
▲ Technical assessment of a work product created during the software engineering process
▲ Software quality assurance mechanism
▲ A training ground
▲ Communication!

What Reviews are Not!

They are not:
- A project budget summary
- A scheduling assessment
- An overall progress report
- A mechanism for reprisal or political intrigue!!
Potential Players

- Review Leader
- Standards Bearer (SQA)
- Producer
- Maintenance Oracle
- Recorder
- User Rep
- Reviewer

Conducting the Review

1. Be prepared—evaluate product before the review
2. Review the product, not the producer
3. Keep your tone mild, ask questions instead of making accusations
4. Stick to the review agenda
5. Raise issues, don’t resolve them
6. Avoid discussions of style—stick to technical correctness
7. Schedule reviews as project tasks
8. Record and report all review results
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>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>&quot;Reader&quot; presents product</td>
<td>maybe</td>
<td>no</td>
<td>no</td>
<td>yes</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

Metrics Derived from Reviews

- Inspection time per page of documentation
- Inspection time per KLOC or FP
- Inspection effort per KLOC or FP
- Errors uncovered per reviewer hour
- Errors uncovered per preparation hour
- Errors uncovered per SE task (e.g., design)
- Number of minor errors (e.g., typos)
- Number of major errors (e.g., nonconformance to req.)
- Number of errors found during preparation
Statistical SQA

• Collect information on all defects
• Find the causes of the defects
• Move to provide fixes for the process

... an understanding of how to improve quality ...

Doggie Dating System

▲ 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 Statement of Scope

▲ The Doggie 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 2/16/01

▲ Read Pressman Chapters
  ● 9.1-9.9

▲ Develop Project Plan for Doggie Dating System Project
  ● Use SEPA Project Plan Template (will be on both CS5704 and SEPA websites)
  ● Based on basic Statement of Scope
  ● Derive additional requirements
  ● Provide initial estimates using COCOMOII

▲ Have a great week!