

# CS577b Quality Mgt. 1

## Overview & CM/MDM



# Announcement

**Team formation form is available from class main page.**

**Wishes for projects by teams to be completed by THIS  
FRIDAY, 1/19, NOON**

# Goals of Presentation

You should get

- An introduction to Quality Management in MBASE/CS577b
- Some immediate remediation
  - Quality Assessment actions
  - Quality Management Reports
  - Rose modeling
- Relevance of CM
- Principles behind CM
- Preview of coming attractions

You should learn about ClearCase

- How it supports CM and source code control
- How it will be used in CS577b

# Quality Management Data

## Why

- Large, well managed systems development norm
- For our CeBASE research
- For our COQUALMO tool research

## What

- Defect history: readiness to deploy
- Defect types and descriptions
- Defect densities
  - injection and removal data
  - for trial calibrations of tool

# Immediate Remediations

## Capture the information correctly: many Quality Reports

- Ways of "finding" defects: Internal Reviews for LCP & FRD
- Extent of reports: **ALL** defect finding techniques including
  - Inspections, including requirements reading techniques
  - Internal Reviews (for non-critical [parts of] documents)
  - Formal Technical Reviews (ARR's)
  - Testing
  - Field reports (including trial use)

## Capture the right information: Defect descriptions from inspections or internal reviews should help researchers identify what "triggered" the identification

- **BAD:** "Purpose redefine" [Major; Extra]
- **BETTER:** "~~removed~~ purpose XYZ ~~which~~ was never mentioned by client" [Major; Extra]

# Immediate Remediations (cont.)

## LCP descriptions of Quality Management

- Many of last semesters LCP's went on and on about inspections which should have been done by reference.

## Quality Assessment Reporting Plan/Schedule

- Few of last semester LCP's said what was to be inspected (or internally reviewed) and when, even in terms relative to "milestones"

# Immediate Remediations (cont.)

## Rose modeling

- Guidelines have been made more explicit and named more concretely
- Template (mapping diagrams to document sections)
  - should be followed to allow easy assessment of the model's conceptual integrity
  - together with packages like UCVtoLCVMapping, provides the basis for architecture to implementation verification tools
- RUP's diagram "Artifact Guidelines" may helpful

# Future Quality Management Lectures

## January 31: QM II

- Design and code defect identification techniques:  
Inspections vs. testing
- Development approaches and types of testing
  - Daily [bi-weekly for 577b] builds
  - Units and integration

## February 26: Code Control and Testing Techniques

- Branching and merging [before "unit" integration]
- Integration vs. System/Acceptance Tests
- Clear Box vs. Black Box testing

# Configuration Management

**Can't make sense out of quality data if configuration is not "controlled"**

**Loose control of changes if configuration isn't "managed"**

## **Source code control**

- version check out and check in with comments and meta-data
- tools include emacs, PVCS, RCS, etc.

## **Problems compound when**

- there are teams doing the same tasks (design and development; system test development)
- multiple versions exist simultaneously (integration and systems test)

# Product Data Management

**Baselines** (client & developer & development mgt. agreement)

**Change control:** large project contract specified deliverables

- Baselines plus
- "Engineering change requests"
- "Engineering change proposals"

**Artifact control: Documents and code and design and requirements**

- Clear development exit conditions
- Permits clients "buy-off"

# Configuration Management in CS577b

**ClearCase – a Rational Software large system development support tool**

- Two interfaces: Web and GUI
- Six Functions

**Introduced and demonstrated by former CS577a classmate: Greg Flesch**