



Fall 2009

Dr. Kevin Hamlen



# CS 4485: CS PROJECT

# Stage 4 Demo

- Stage 4 demo NEXT WEEK
  - Fully working Stage 1+2+3
  - Your stage 4 option mostly completed
    - Option 1: k-replication
    - Option 2: message-dropping defense
    - Option 3: boolean search + voting
    - Option 4: interoperate w/another team's project
  - Come to my office (not here): ECSS 3.704

# Maximizing your Grade

- Stage 3 minor catastrophes
  - several teams unable to effectively demo
  - some complete rewrites in progress (yikes!)
- Demo grades
  - based on what a customer would think based on in-demo presentation of completed features
  - “Nothing works” can sink your company even if you later you get it to work! (lowers customer confidence)
- Maximizing your final grade
  - A fully working Stage 1+2+3 with no Stage 4 at all is WORTH MUCH MORE than a complete Stage 4 atop a malfunctioning Stage 1+2+3!
  - Prioritize getting Stage 1+2+3 to work flawlessly even if it means sacrificing Stage 4.

# Stage 1+2+3

- Only one form of acceptable evidence for a fully working Stage 1+2+3: TEST SUITE
- Test suite must...
  - launch at least 20 clients (on same machine)
    - clients must have RANDOMLY chosen port numbers (within some large range of free ports on the machine)
  - then randomly do the following in a loop:
    - join a new client
    - gracefully leave a client
    - ungracefully kill a client, followed by sending lots of messages (to re-stabilize the network)
    - publish/download/retract a file
    - associate/unassociate a keyword
  - do not wait for previous operation to complete before beginning next one (to test for concurrency bugs)
    - exception: possibly wait after ungraceful kills for network to re-stabilize

# Final Submission


- Tuesday, 12/15, 11:00am-1pm
  - Bring TWO copies of all deliverables to my office (ECSS 3.704)
- Deliverables
  - CD/DVD with all sources and project files
  - include binaries if targeting non-Unix
  - README with complete installation instructions
    - required third-party software (e.g., Java runtime)
    - any required PATH variable settings, etc.
  - User manual
    - explains basic operation for non-scientists (e.g., configuration instructions, explanation of features, limitations, trouble-shooting steps)
    - electronic form (ideally a LOCAL webpage)
  - Technical manual
    - explains how to launch the test suite and interpret the results
    - documents the API so that third-parties can create new front-ends
    - can be electronic or written
  - Final progress report
    - discloses final project state and any known bugs or deficiencies

# Instructor/TA availability

- Office hours today
- Available next week by email/appointment
- Questions about testing architecture
  - contact Ajay
- Questions about requirements/deliverables
  - contact me

# Taking the next step..

- Release your product
  - Put it on SourceForge, CodeProject, etc.
  - Make a nice webpage
  - Point interviewers to it during interviews or in resumes
- Clean it up
  - USE your product for a while
  - Change GUI features that annoy you after the first 30 minutes of regular use
  - Fix bugs that arise during first week of use
- Apply it to an interesting domain
  - p2p is not just for music-sharing or file-sharing
  - examples: interlibrary loan of ebooks, sharing of DNA analysis data, distributed computations (e.g., seti@home)



Hope you enjoyed  
the course!