COSC 499: Capstone Software Engineering Project

Dr. Bowen Hui
Computer Science
University of British Columbia Okanagan

Term 1 Where We're At

- By now (Milestone 2): MVP for your project
  - Review machine learning and gamification feedback
- **Milestone 3:**
  - UBC server deployment with working core features
- Team dynamics:
  - Everyone should know their teams well and plan accordingly
  - Most teams show good coding collaboration process
  - Continue discussions on testing, code reviews, work distributions
  - **Peer evaluations** used to weigh individual contributions, especially important for teams with 3 members

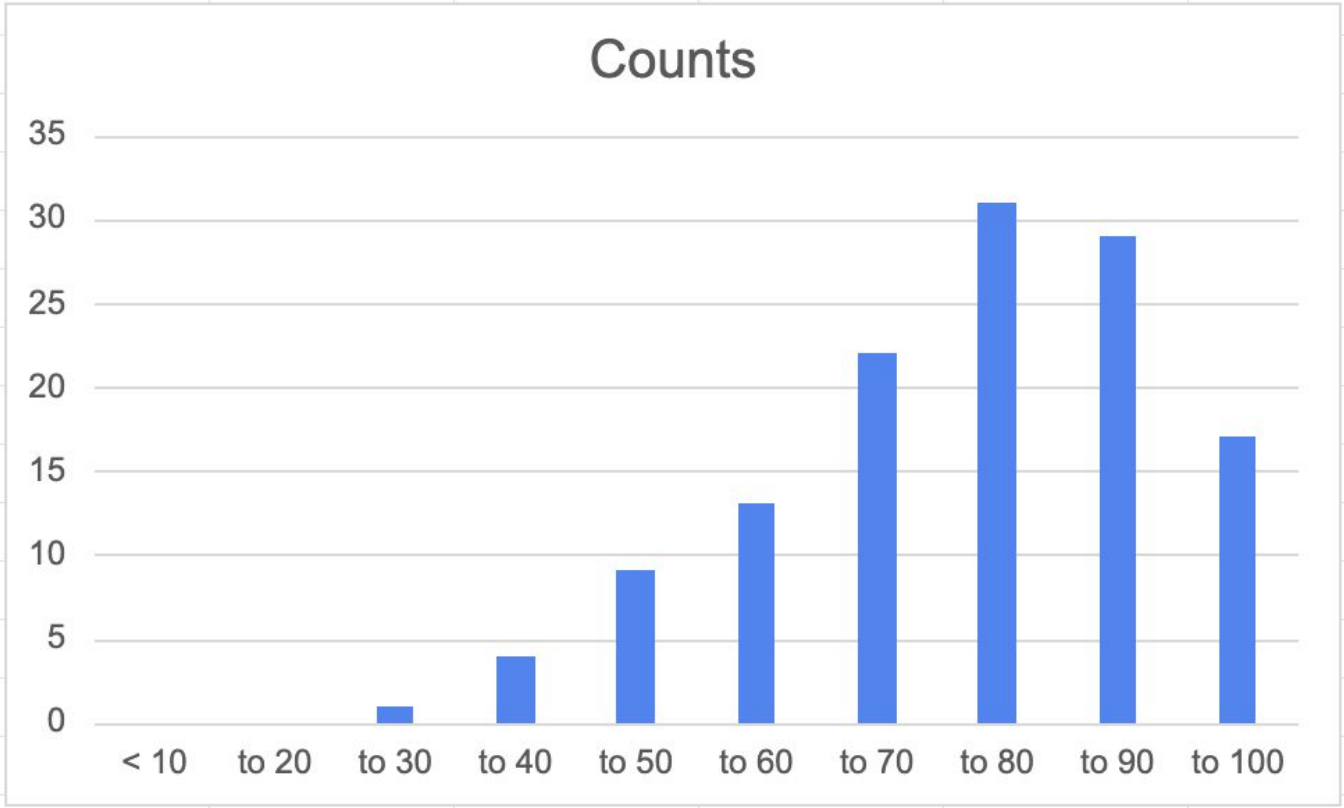
Term 2 Timeline

- Today: resume classes
- Week of Jan 27: **Peer testing #1**
  - Focus on workable core requirements on production server
  - Prioritize usability feedback
- Week of Feb 10: **Milestone 3**
  - Complete optional requirements
  - Work on polishing and refactoring
- Week of Mar 10: **Peer testing #2**
  - Prioritize usability feedback
- Week of Mar 31: **Milestone 4 (final project)**
  - No additional project work once submitted
- Final exam period: **Final exam**

Midterm Review

Tentative Results - PR verification pending

- Histogram for both classes N=126 students
- Average: 72.7% Max: 100%
- % passed: 89%
- % A's: 37%



Question 1: System Architecture

Avg 5.1 / 8 (64%)

- Common mistakes for Part (a):
  - Using generic labels/diagram without applying to your project
  - Missing or poor diagram description
  - Missing significant components in diagram
  - Using incorrect system architecture than requested

- Common mistakes for Part (b) and Part (c):
  - Advantage not identified, simply explained architecture organization

Question 2: DFD

Avg 3.9 / 8 (49%)

- Common mistakes for Part (a):
  - Missing data labels, and simply confused about processes and data flows
  - Confusion with data stores - missing or represented as a process
- Common mistakes for Part (b):
  - Mentioned use cases not included (e.g. login, user agent) and key processes that are needed to complete the use case (e.g. text extraction, error display)
  - Missing or poor description
  - Unnecessarily representing external use cases

Avg 7.1 / 8 (88%)

Question 3 and 4: Refactoring Your/Teammate's PR

- Not a code refactor
  - New feature, not an improvement
  - Others: Design in Figma, team communication, …
- Improvement too generic
  - "some code" with no details of what was in the PR
  - Better user interface - how exactly?
  - More test cases - which ones?
- Poor context
- Shortcoming not explained
- Rationale or impact of refactor not provided or does not match refactor

Next Steps

- Next week:
  - How peer testing works

- Rest of this class:
  - Record PR number for Questions 3 and 4
  - Address questions regarding grading
    - Focus on mistakes and clarifying misconceptions to improve your final exam performance