Today ...

- Course overview
- Quiz 1
- Soft. Eng. Topics
- Teams

Readings

- Classic mistakes handout

Homework

- Proj part 1 (due Tues.)
Course overview

WARNING: This is a very different course than your other CS classes!

Largely non-technical

... no new programming languages, algorithms, theory, etc.
... some new soft. dev. tools and techniques

More reading, contemplating, writing (than other CS courses)

... so readings are required!
... you’ll be tested on readings

More group/teamwork activites (than other CS courses)

... one large semester team project
... smaller individual assignments

But probably one of the more relevant courses for life after graduation

... as a computer developer, project manager, etc.
Course Logistics

Me:

- email: bowers@gonzaga.edu
- office: Herak 309C
- office hours: 2-3pm or by appointment

Course webpage:

- [http://www.cs.gonzaga.edu/~bowers/courses/cpsc330](http://www.cs.gonzaga.edu/~bowers/courses/cpsc330)

Email list:

- Google group
- I'll sign you up (see handout)


- other readings and handouts
- again, readings are **required**
Grades:

- 15% Assignments
- 15% Group project
- 20% Quizzes
- 30% 3 mid-semester exams
- 20% Comprehensive final

Standard letter grades: A (90-100), B (80-89), C (70-79), D (60-69), F (< 60)

Attendance

- 4 absences may result in a V letter grade (GU policy)
- Emergencies
  - notify me asap if you are going to miss a class (prior to class if possible)
  - may be able to arrange special schedule in certain cases
Group project

Grading

- a series of project assignments
- some short presentations over the semester
- a final project report & presentation
- self evaluation
- evaluation by your team members

Quizzes

Short (5-10 min), weekly or more frequent

Why quizzes?

1. examples of questions I tend to ask on exams
2. gives me feedback on how folks are doing
3. gives you feedback on how you are doing
Two major themes in the course:

1. **Software development** tools & techniques
   - testing
   - build management
   - source code control
   - design approaches
   ... look mainly at Java tools
   ... don’t be scared if you don’t know Java!

2. **Software project management**
   - why it is hard
   - different approaches
   - chartering, requirements, estimation, planning, etc.
   ... focus on Scrum (an Agile process)
   ... Scrum used in the textbook

The main goals of the course:

- An introduction to the field of software engineering
- Approaches to apply in real projects
- Help you “hit the ground running” in senior design
Intro to Software engineering ...

(Good) Software Engineering is hard

Q: Why is software developed?

To achieve ("business") objectives

- make a profit direct profit
- save time/money (reduce cost of other tasks) indirect profit
- attract investors (eg, proof of concept) indirect profit
- others: gain experience, contribute/join a community (open-source)

Q: How do we know if the software developed is successful?

1. if it meets the objectives! ... but this can be tricky:
   - developed correctly (specifications), but not meet business objectives
   - for some objectives, assume that it will meet objectives if built correctly
     - eg, number of purchases, customer satisfaction, etc.

2. if it is not too costly to develop
   - profits directly tied to cost of building the product
   - “cost” is an objective we need to worry about

Unfortunately, software development is notorious for cost overruns!
Some definitions

- **Failed project**: cancelled or never used
- **Challenged project**: late, over-budget
- **Successful project**: on time & budget, met customer needs

Q: What % of projects are in each category?

The [Chaos Report](https://www.cacm.acm.org/interviews/2006/062306.cfm) (Standish group)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Successful</td>
<td>16%</td>
<td>27%</td>
<td>26%</td>
<td>28%</td>
<td>34%</td>
<td>29%</td>
<td>35%</td>
<td>32%</td>
</tr>
<tr>
<td>Challenged</td>
<td>53%</td>
<td>33%</td>
<td>46%</td>
<td>49%</td>
<td>51%</td>
<td>53%</td>
<td>46%</td>
<td>44%</td>
</tr>
<tr>
<td>Failed</td>
<td>31%</td>
<td>40%</td>
<td>28%</td>
<td>23%</td>
<td>18%</td>
<td>19%</td>
<td>19%</td>
<td>24%</td>
</tr>
</tbody>
</table>

This is only one survey, but many others have found similar results

McConnell, 2006:

- $\approx 25\%$ delivered on-time
- $\approx 25\%$ cancelled
- $\approx 50\%$ delivered late, over budget, or both