Miscellaneous Stuff

- TA hours are posted on the web:
  - Erica Rice. Thurs 2:30-3:30, CSE Lab 002
  - Christian Bell. Mon 2:30-3:30, CSE Lab 002

- I’m not at UW on Tues, Thurs. Always reachable by email.

- Grading description on web has been updated with our Wed decision

- Signup LCO group by 11:59pm tonight
  - Link on the web

- Questions about the assignment?
  - Open source
  - Skill set of group
Software Development Lifecycle – Part II

The Power of Process
Readings

• “Rapid Development”, Steve McConnell
  o Chapters 7, 10, 21, 25, 35, 36
• “Anchoring the Software Process”, Barry Boehm
  o Pages 1-10 in particular
Review

Take 5 minutes

Work with your row to describe:

- Code and fix (row 1)
- Waterfall (row 2)
- Spiral (row 3)
  - What it is, one advantage, one limitation
- One example project suited to each and why (row 4)
Staged Delivery Model (Agile)

Waterfall-like beginnings, then develop in short release cycles: plan, design, execute, test, release with delivery possible at the end of any cycle.
Staged Delivery Model

Very practical in practice, widely used and successful

Advantages

- Can ship at the end of any release-cycle
- While not feature complete, intermediate deliveries show progress, satisfy customers, and provide opportunity for feedback
- Problems are visible early (i.e., integration)
- Facilitates shorter, more predictable release cycles

Limitations

- Requires tight coordination with documentation, management, and marketing (this is a good thing!)
- Must be decomposable
Evolutionary Prototyping Model

Develop a skeleton system and evolve it for delivery

What class of problems lends itself to this model?
Evolutionary Prototyping Model

Another popular and successful model, especially for custom products

Advantages

- Addresses risks early
- Produces steady signs of progress
- Useful when requirements are changing rapidly or customer is non-committal

Limitations

- Requires close customer involvement
- May spell trouble if the developers are inexperienced
  - Feature creep, major design decisions, use of prototyping time, etc.
- Hard to estimate completion schedule or feature set
Why are there so many models?

You tell me!

Choices are good!

- The choice of a model depends on the project circumstances and requirements
- A good choice of a model can result in a vastly more productive environment than a bad choice
- A cocktail of models is frequently used in practice to get the best of all worlds. Models are often combined or tailored to environment
Some themes occur in all the models

What do you see as common?

- Risk reduction
- Prioritization
- Customer involvement and feedback
- Visibility of progress
Which Model to Use?

Which model or mix of models would you use for your quarter-long project where you work as part of a team? Why? What else do you need to know to answer this?