The Home Stretch
The Plan

- Review of the Design
- The Last 10%
- Freezing the Project
- Ensuring Success
- Cutting your losses
- Testing
- Documentation
- Enhancements
Review of the Design

• Late for Serious Changes
  – May be some course correction possible.

• Often: Simplification rather than enhancement
  – Shed some frills.
  – Narrow down some of the goals.

• When Adding Things: Must be Orthogonal!
  – Orthogonal: Will not adversely affect other parts.
  – Don’t risk project for that “just one more”.
The Last 10%

• You may think you’re almost done
• Common wisdom is:
  \[ \text{The last 10\% of a project takes up } \frac{90}{90} \text{ of the time.} \]
• Allow for the unexpected
• Prevent serious problems by testing as you go
• Don’t put off fixing a problem to go to another enhancement
Freezing the Project

- Earlier rather than later, quit adding features
- Make sure what you have is done right
- Allow for the unexpected
- Resist the lure of enhancements
- Spend any extra time to make product more professional
Ensuring Success

• *Always* have a working project
  – After some natural increment (with everything so far working) *save a complete copy (project)*.
  – Think of it as a simpler version.
  – Don’t lose or change this copy.

• Start the next version
  – After significant progress, *repeat* this process.

• Too often, a formerly working project is “destroyed” by improvements
  – It may never work again.
  – You must have a previous version to *fall back on*.
Cutting your Losses

• Sometimes: Nothing Seems to Work
• *Before* it’s too late:
  – Severely *Prune* your goals.
  – Get help from instructors for this.

*A Simple Working Project*

*ALWAYS RATES HIGHER*

*than a Fancy Non-Working Project*
Testing

“Testing can prove the presence of bugs, but not the absence of bugs.”

- Games: different standard from most software
  - Bugs sometimes become features.
- Errors can be costly, however
  - When demo-ing for your grade!
  - When showing off for your date.
- Test Plan
- Testing by a naïve user
Documentation

• Javadocs provide imbedded documentation
• Traditional
  – Technical/Program Logic Manual (for heirs).
• On-line documentation
• Legal / for Grade
  – Make clear who did what work and when.
Enhancements

• *If* you have time …
  – Make sure you have polished main parts.
  – Make sure you do not risk previous work.

• Enhancements can be a lot of fun
  – Extra: can add that sparkle.

• The pressure is off
  – Little cost: if it doesn’t work (back it out).