Estimation Wrap-up
Why do we need estimates?

- To know
  - If a customer request for a delivery date is reasonable
  - If we can provide more features in the time we have
  - If we need more people to hit a deadline
  - How much to bid for a job
  - When we can provide our deliverable to another group, possibly internal, who is dependent on it
  - When we’ll be available to move to the next project
  - Which path is more cost effective to pursue (ie. build vs buy, design A or design B)
  - ....
Estimation in a nutshell

- There are multiple ways to come up with an estimate

- Make sure you take into account the broad spectrum of factors, including
  - Size of project
  - Type of project, type of language
  - Vacations, meetings, non-technical work
  - Size and talent in the team
  - Scope of the project
  - Input from past experience (yours and/or others)

- Try to avoid the deadly sins …
Testing Wrap-up

Coding Ain't Done 'Til All the Tests Run

Pragmatic Programmer
Take it from me, Microsoft begins every project with the certain knowledge that they will choose to ship with known bugs.
Integration/system testing

Shows that the major subsystems that make up the project work well together.

- **big bang**: no stubs; do unit testing, then throw all parts together

- **bottom-up**: integrate upward into double, triple, quadruple module test

- **top-down**: test top layer (UI) first, then add layers to replace underlying stubs
GUI testing

Testing a product that uses a GUI, to ensure it meets its written specifications

• Difficulties:
  o Many operations to test (ie. MS Wordpad has 325)
  o Order of testing matters
  o Regression testing is hard given GUI evolution
  o Need to test on an array of browsers (web apps)
Approaches to GUI testing

- **Automated UI testing**
  - Scripts that use your app and look for failures
  - A black-box system test

- **Manual tests**
  - Human beings click through predetermined paths
  - Need to write down the specific tests each time
  - [http://members.tripod.com/~bazman/checklist.html](http://members.tripod.com/~bazman/checklist.html)

- **Exploratory tests**
  - Human beings are "turned loose" on the app to see if they can break it
Web app compatibility testing

- Motivation
  - ensure that your web app is compatible with various browsers, platforms, etc.
  - ensure that your app's HTML code complies with web standards
  - ensure that you have no broken links or other HTML errors

<a href="http://validator.w3.org/check/referer"><img style="border:0;width:88px;height:31px" src="http://www.w3.org/Icons/valid-xhtml10-blue" alt="Valid XHTML 1.0 Strict" /></a>

<a href="http://jigsaw.w3.org/css-validator/check/referer"><img style="border:0;width:88px;height:31px" src="http://jigsaw.w3.org/css-validator/images/vcss-blue" alt="Valid CSS!" /></a>
TIME BREAKDOWN OF MODERN WEB DESIGN

- Swearing
- Time spent making the site W3C compliant.
- Time spent trying to get the bastard to work in Internet Explorer.
- Time spent wishing a slow, painful death on Bill Gates and more swearing.
- Time spent looking for that one extra space character in the JavaScript that Fire Fox is throwing a wobbly over.
- Time spent actually designing anything.
- Time spent trying to get the layout to work using only CSS before giving up and using tables.

Pragmatic Programmer Tips

Test early, test often, test automatically

Find bugs once

Test your software, or your users will

Design to test
BUGS!

Managing Bugs
What is a bug?

- Formally, a “software defect”
- System fails to perform to spec
- System causes something else to fail
- System functions, but does not satisfy usability criteria

If the system works to spec and someone wants it changed, that’s a **feature request**
What makes a good bug report?

Include:

- **Reproducible steps** – how did you cause the failure?
- **Observed result** – what did it do?
- **Expected result** – what should it have done?
- **Any collateral information**: return values/output,…
- **Environment**
  - OS version, env variables, compiler flags, …
  - Test platforms must be reproducible
  - “It doesn’t do it on my machine”
Why is a bug tracking tool valuable?

Vs, say, a spreadsheet?
Metrics can show quality

**Figure 16-1** Example of an “open defects” graph. Making this graph public emphasizes that controlling defects is a high priority and helps to keep potential quality problems under control.
Another example

Open defects by criticality (p1=high, p5=low)

Is this system ready to release?
Example 3

Open defects by module

How is this view helpful?
A bug’s life (simplistic)

1. Bug activated
2. Triage Fix?
   - YES: Defect fixed - bug resolved
   - NO: Won’t Fix, Not Repro, By Design, Postponed

   - Regression testing
     - Fixed?:
       - YES: Bug closed
       - NO: Regression testing (loop)

CSE 403, Spring 2008, Alverson
Team Member Contribution Assessments

Scores will be in database by end of day

You’ll hear from me only if there are exceptional comments

Email me if you want a summary
Beta and Final Releases

Excellent job on the Beta!
(We spent a huge amount of time grading it together – your products are looking really nice)

Let’s walk through Final Assignment so you can make sure you understand all the components