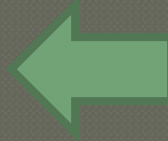


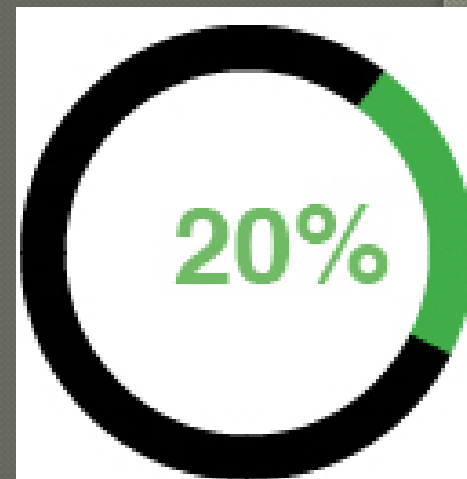
Design Patterns

- ◉ Automated Testing
- ◉ JUnit
- ◉ **Test-Driven Development**
- ◉ Test Coverage
- ◉ Integration Tests



Test Driven Development

- Add requirements/stories
- Write automated tests for each
- Write code until it passes
- Test pass % acts as progress bar to next release



Zero-Defects Philosophy

Introduced by Philip Crosby at the Pershing Missile Program

The philosophy was introduced to reduce the failure rate of the Pershing Missile

Major failures happen when engineers tolerate mistakes because they know inspectors will detect them later

Getting it right the first time dramatically reduces errors

Defects in Nuclear Fabrication



(agile) Zero-defects Software Development

- Don't write code you know to be bad, because your friends will catch the error in testing
- Don't build new code on defective code
- Don't add features if your tests aren't passing

Let's Be Real

- You will not ship with zero defects
- Perfect testing and defect control is not worthwhile
- Quality checkpoints will waste time on your small projects
- Buggy something is better than perfect nothing
- Just write code and fix bugs as they arise before continuing
 - This is the most important aspect of ZD
 - Don't build on top of bad code
- Good post: <http://www.agiledevelopment.org/agile-talk/134-zero-defects-in-software-development>