CLEMSON UNIVERSITY

CPSC 3720
Lesson 34

Connie Taylor
Professor of Practice

School of COMPUTING

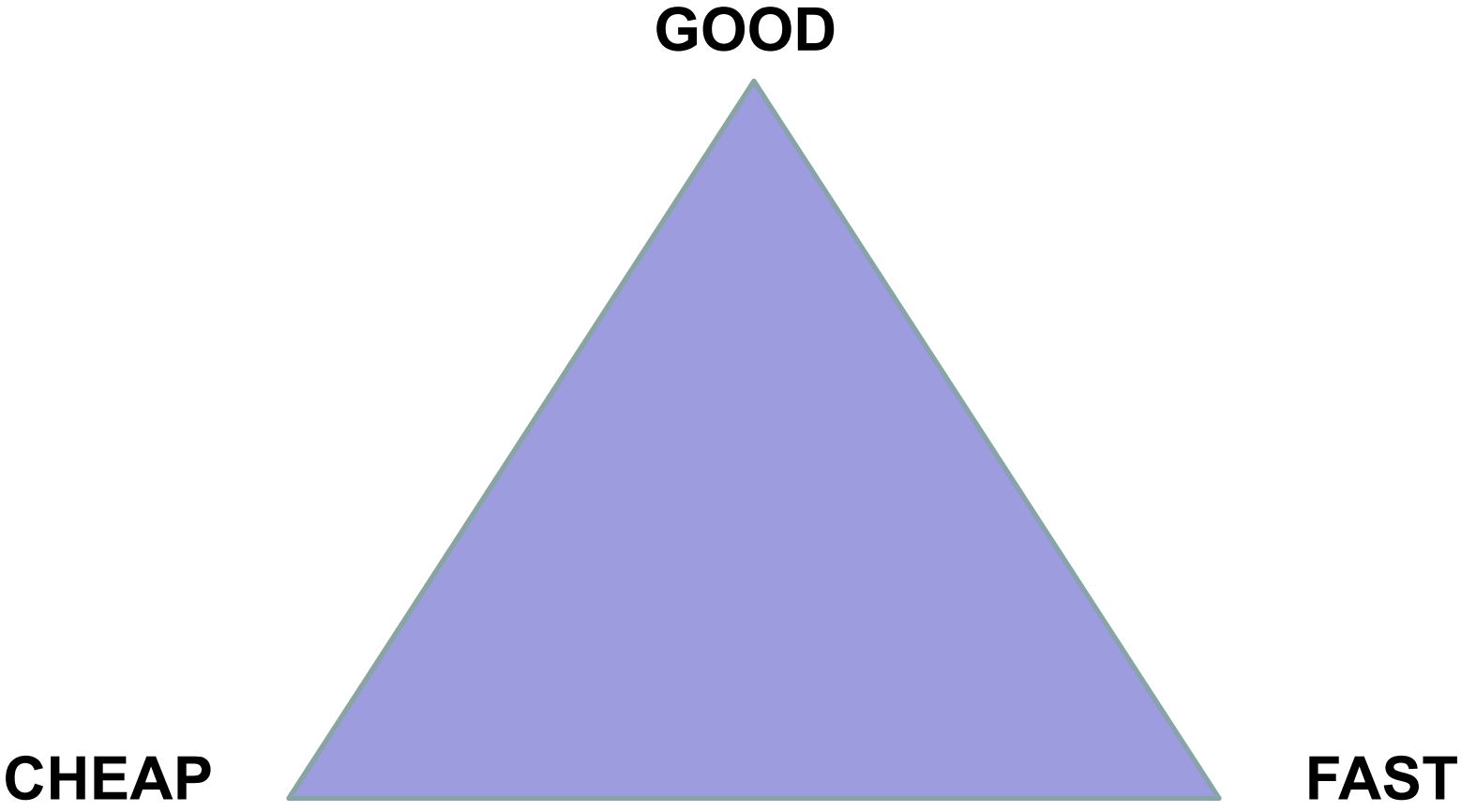
Don’t Go Dark

What does it mean to “Go Dark” in software and how do you avoid this?

How does Agile prevent going dark?

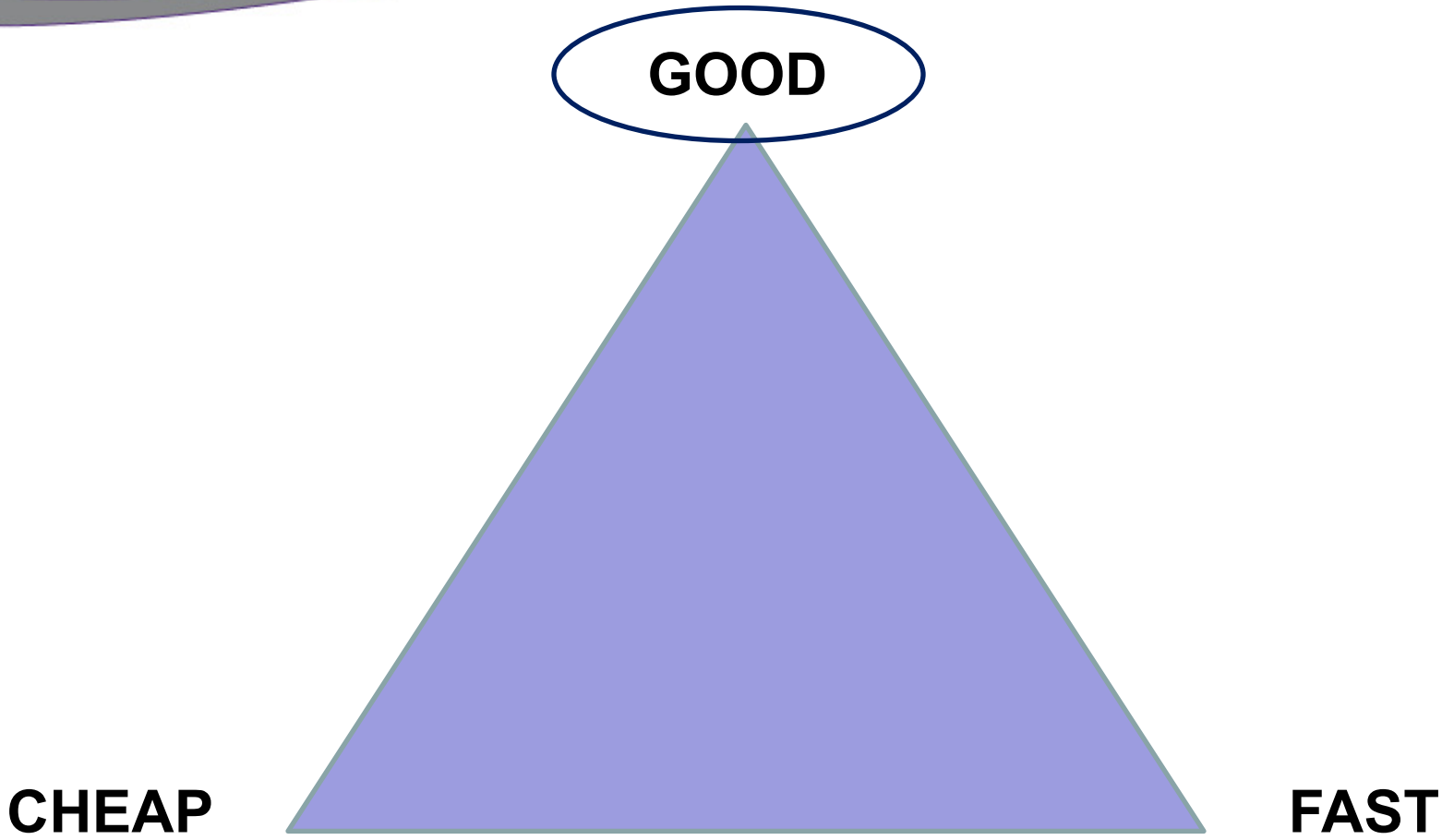
- Agile focuses on creating plans that:
  - **Are highly honest and visible**
  - Focus on customer value in the shortest amount of time
  - Allow you to change course
  - **Enable frequent progress reviews with each sprint**

Software Planning Triangle



Pick any two…

Software Planning Triangle



Pick any two…

Good Software

DISCUSSION:

What is "good" software?

and

How do you deliver "good" software?

Good Software means focusing on the ”ilities”

- Usability
- Maintainability
- Performance (or Perform-ability)
- Scalability
- Extensibility
- Security
- Portability (if delivered on-prem)
- Reliability and/or Availability
- Internationalize-ability
- Interoperability
- Audit-ability
- Administrability
- Configurability

TCO total cost of ownership

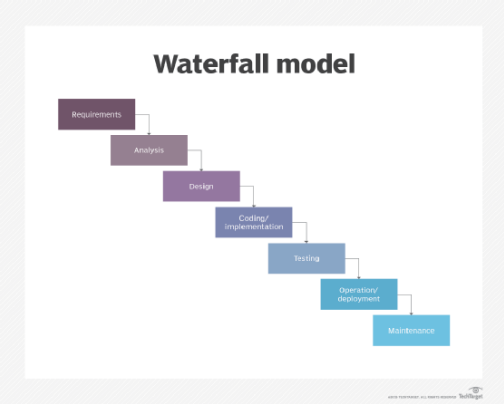
Good Software through Testing

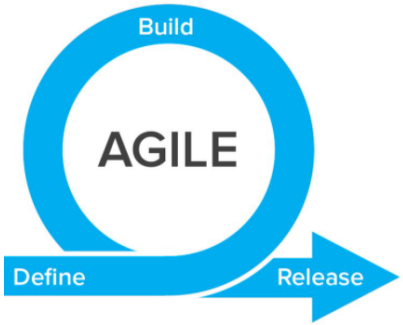
- **Functional Testing: (alpha beta functional)**
  - **Unit Testing:** code working properly
  - **Functional testing:** doing what you said it would do
  - **System and Integration testing:** does it all work together
  - **Exploratory Testing:** testing on the fly
  - **Regression Testing:** Tests to ensure that new functionality did not break existing functionality (can also be used for non-functional testing)

- **Usability Testing:**
  - **Alpha/Beta testing:** usability tests for customers
  - **Usability Studies:** observe customers using the software software or mocks

- **Non-functional Testing:**
  - **Recovery testing:** forces software to fail and verify that recovery is (ala Chaos Monkey) properly performed
  - **Security testing:** ensure protection mechanisms built into a system will, in fact, protect it from improper penetration
  - **Stress or soak testing:** executes system in a manner that demands resources in abnormal quantity, frequency, or volume
  - **Performance Testing:** test the run-time performance of software within the context of an integrated system

Testing – Waterfall vs. Agile

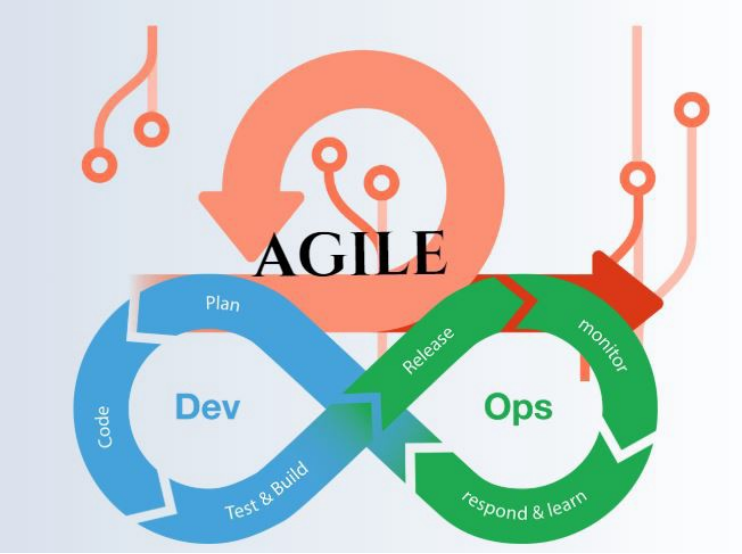




- Functional Testing at the end of development cycles
- Testing Teams usually separate organizations from development
- Regression testing for each release
- More manual testing

- Testing during each sprint and in parallel
- Scrum teams own responsibility for quality
- Regression testing ongoing
- Increased need for automation

Testing – DevOps



- "Shift Left" and Test-Driven Development (TDD)
- More blur between Dev and Test roles
- Automation is ESSENTIAL

Class Updates

- Sprint 2 due today
- Reading for Monday in Canvas
- TDD Deeper Dive and Sprint 3 Kickoff Monday
- Quiz 4 due on Tuesday night (opens Sunday)
- Sprint 2 Reviews will be Wed/Fri next week (will send out schedule)