

CSSE 220 Day 18

Object-Oriented Design

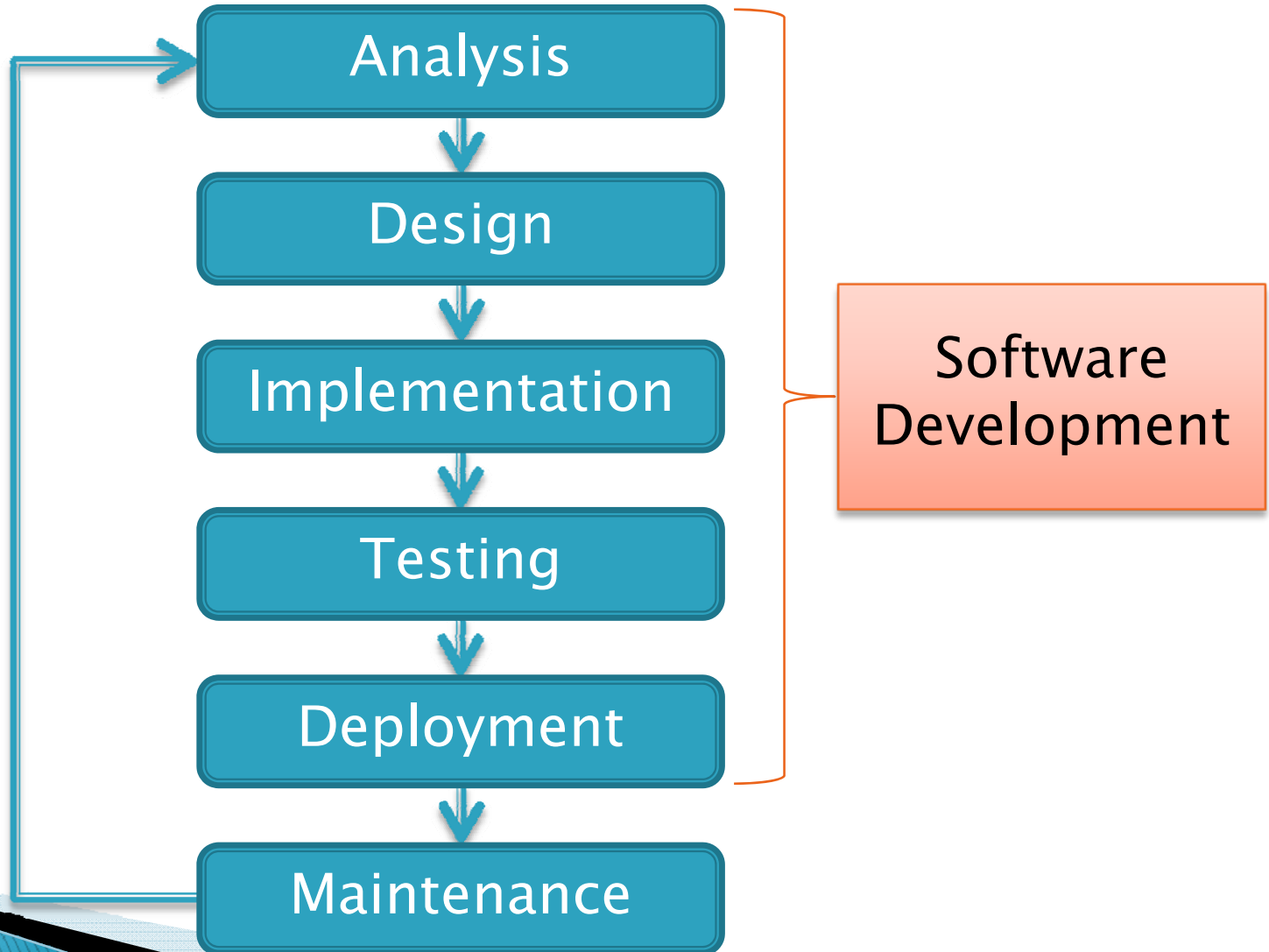
No SVN checkout today

Questions?

Software Development Methods



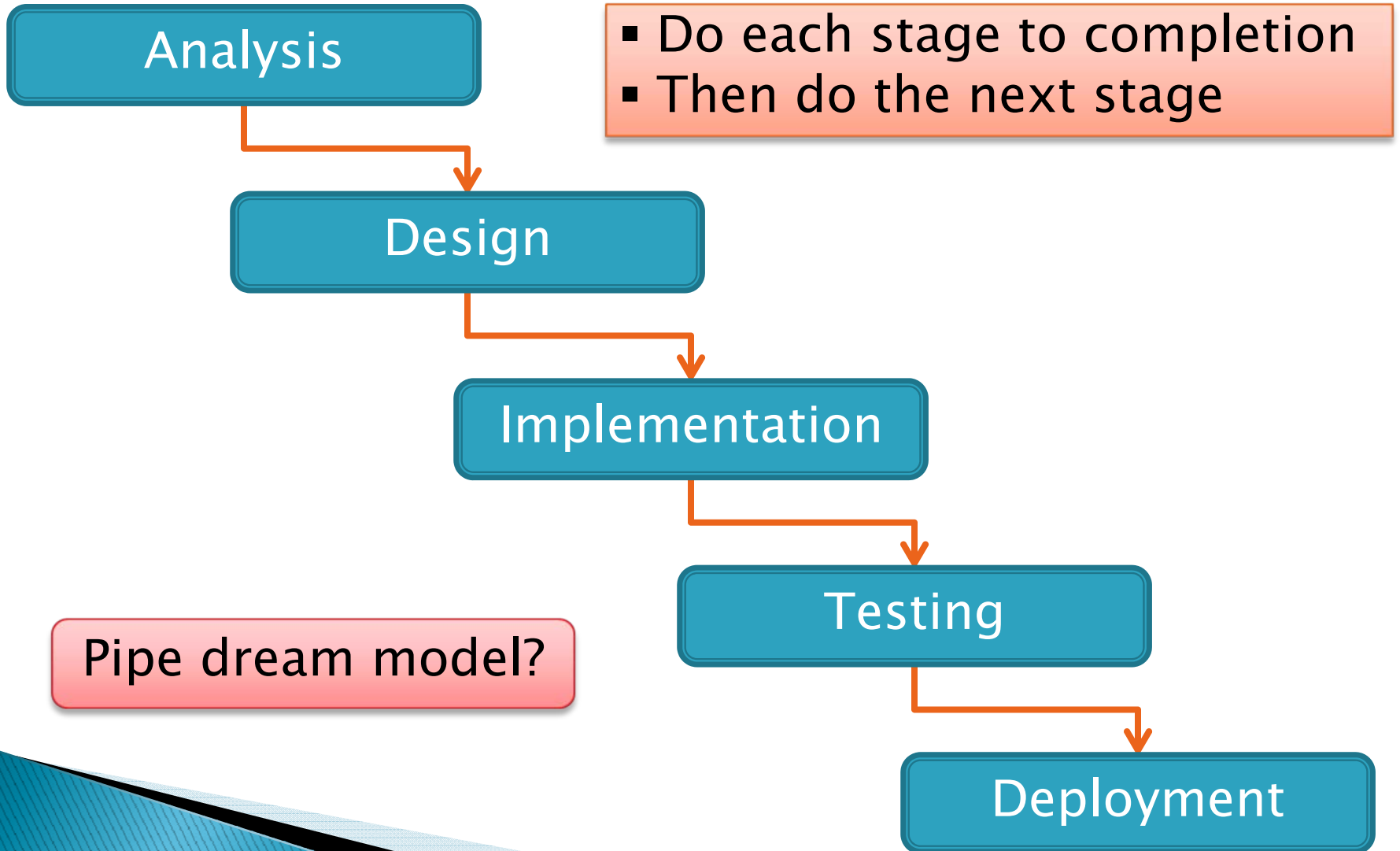
Software Life Cycle



Formal Development Processes

- ▶ Standardized approaches intended to:
 - Reduce costs
 - Increase predictability of results
- ▶ Examples:
 - Waterfall model
 - Spiral model
 - “Rational Unified Process”

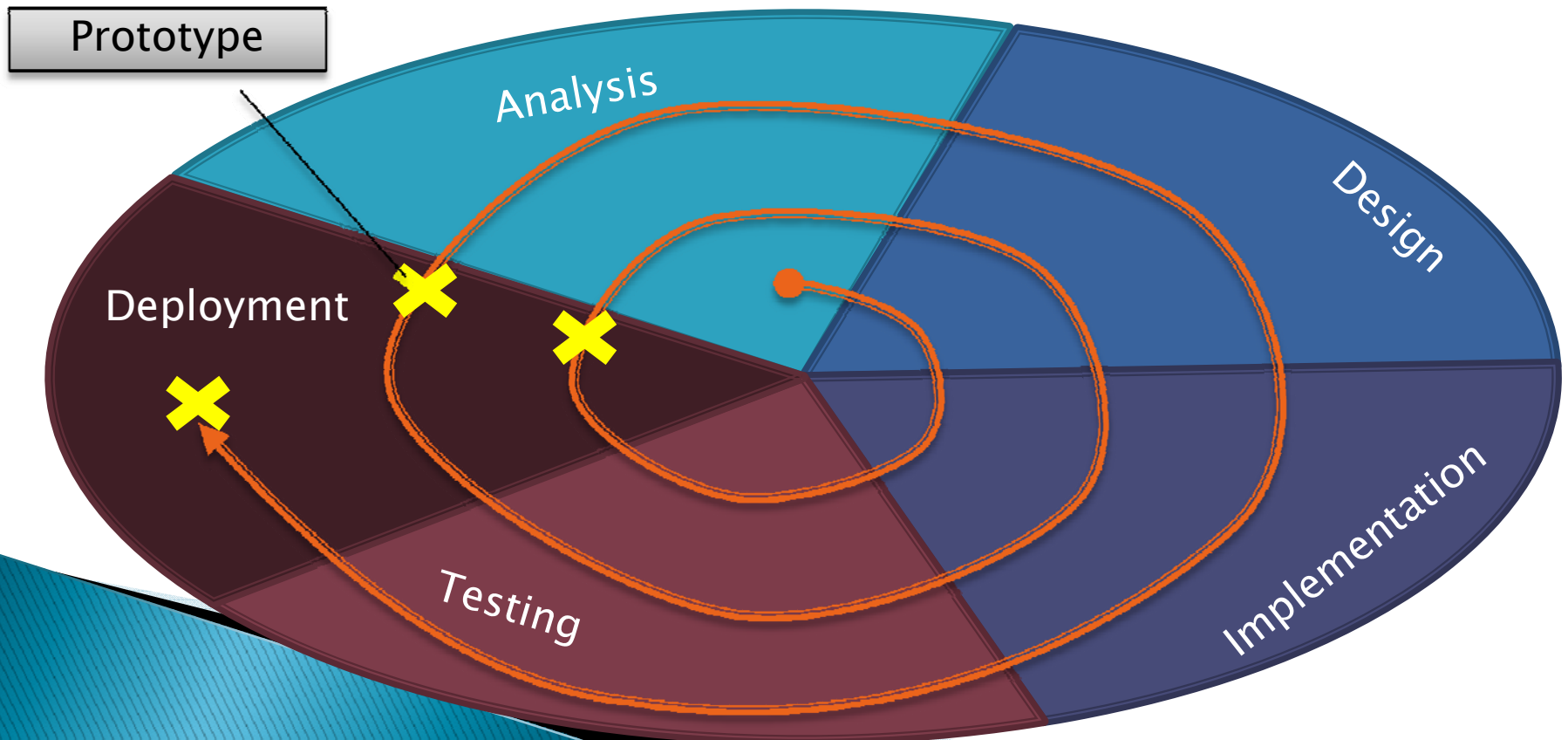
Waterfall Model



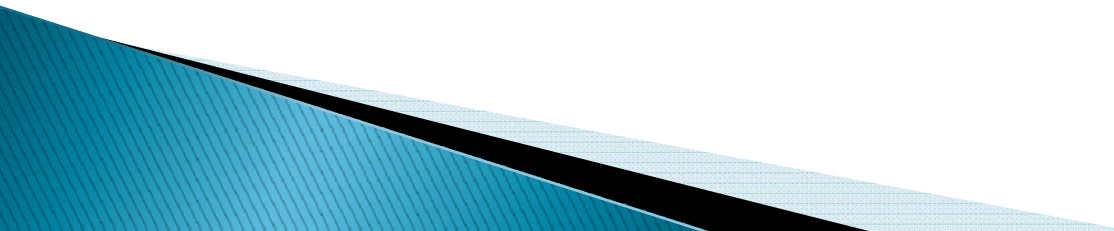
Spiral Model

- Schedule overruns
- Scope creep

- ▶ Repeat phases in a cycle
- ▶ Produce a prototype at end of each cycle
- ▶ Get early feedback, incorporate changes

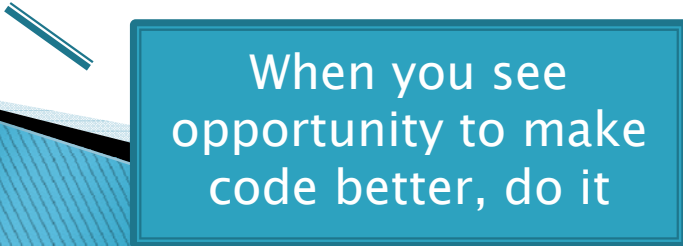


Extreme Programming—XP

- ▶ Like the spiral model with **very** short cycles
 - ▶ Pioneered by Kent Beck
 - ▶ One of several “agile” methodologies, focused on building high quality software quickly
 - ▶ Rather than focus on rigid process, XP espouses 12 key practices...
- 

The XP Practices

- Realistic planning
- Small releases
- Shared metaphors
- Simplicity
- **Testing**
- **Refactoring**
- **Pair programming**
- Collective ownership
- Continuous integration
- 40-hour week
- On-site customer
- **Coding standards**



When you see opportunity to make code better, do it



Use descriptive names

Object-Oriented Design

»» A practical technique

Object-Oriented Design

- ▶ We won't use full-scale, formal methodologies
 - Those are in later SE courses
- ▶ We will practice a common object-oriented design technique using **CRC Cards**
- ▶ Like any design technique, the key to success is practice

Key Steps in Our Design Process

1. **Discover classes** based on requirements
2. **Determine responsibilities** of each class
3. **Describe relationships** between classes

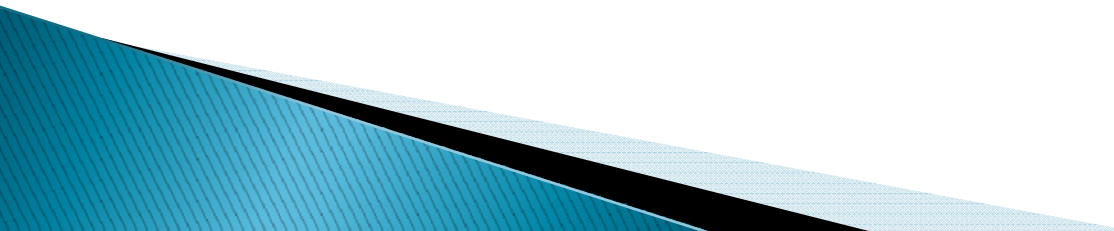
Discover Classes Based on Requirements

- ▶ Brainstorm a list of possible classes
 - Anything that might work
 - No squashing
- ▶ Prompts:
 - Look for **nouns**
 - Multiple objects are often created from each class
→ so look for **plural concepts**
 - Consider how much detail a concept requires:
 - A lot? Probably a class
 - Not much? Perhaps a primitive type
- ▶ Don't expect to find them all → add as needed



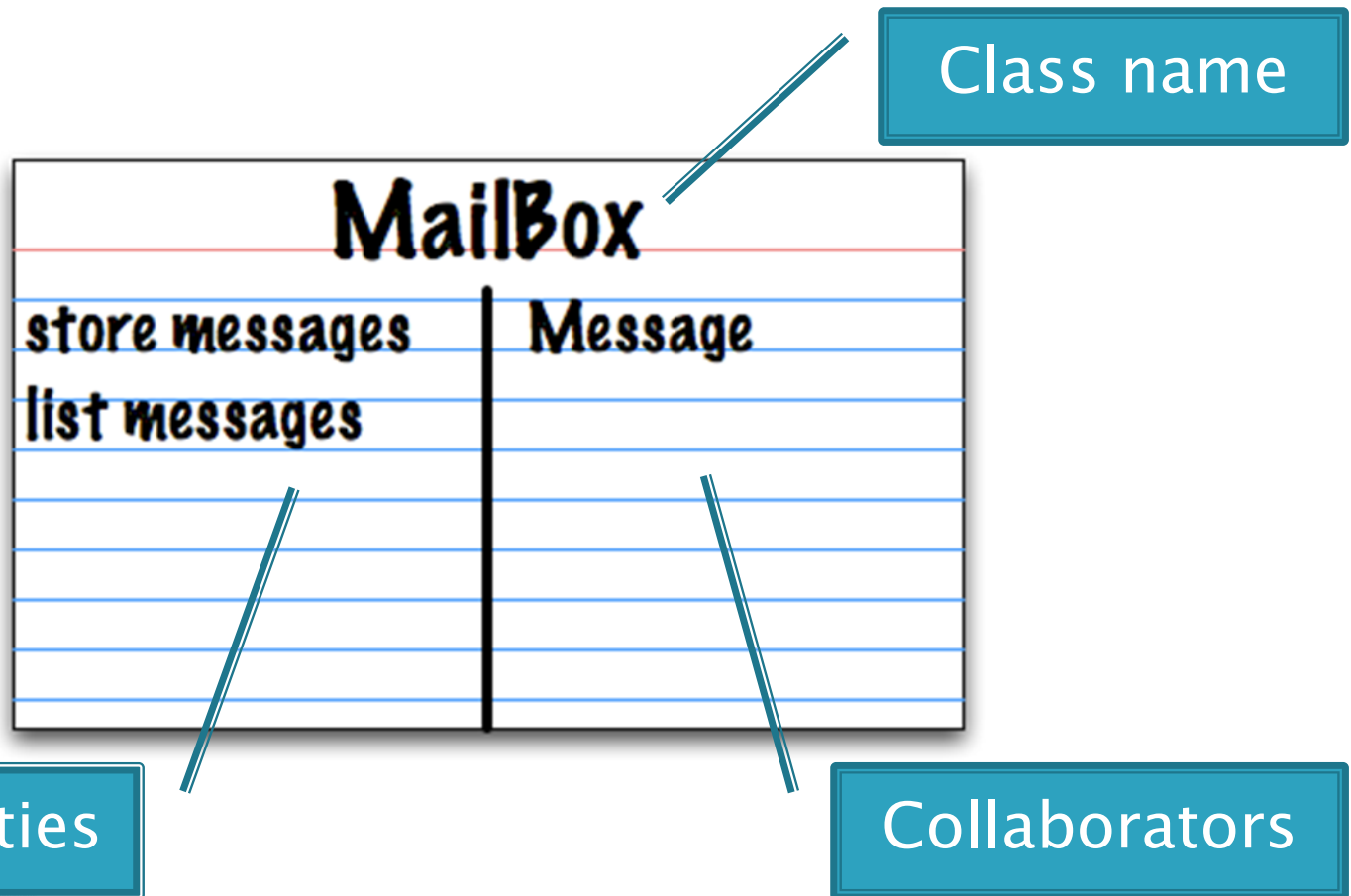
Tired of hearing this yet?

Determine Responsibilities

- ▶ Look for **verbs** in the requirements to identify **responsibilities** of your system
 - ▶ Which class handles the responsibility?
 - ▶ Can use **CRC Cards** to discover this:
 - **C**lasses
 - **R**esponsibilities
 - **C**ollaborators
- 

CRC Cards

- ▶ Use one index card per class



CRC Card Technique

1. Pick a responsibility of the program
2. Pick a class to carry out that responsibility
 - Add that responsibility to the class's card
3. Can that class carry out the responsibility by itself?
 - Yes → Return to step 1
 - No →
 - Decide which classes should help
 - List them as collaborators on the first card
 - Add additional responsibilities to the collaborators' cards

CRC Card Tips

- ▶ **Spread the cards out** on a table
 - Or sticky notes on a whiteboard instead of cards
- ▶ **Use a “token”** to keep your place
 - A quarter or a magnet
- ▶ **Focus on high-level responsibilities**
 - Some say < 3 per card
- ▶ **Keep it informal**
 - Rewrite cards if they get too sloppy
 - Tear up mistakes
 - Shuffle cards around to keep “friends” together

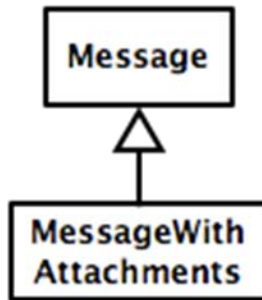
Describe the Relationships

- ▶ Classes usually are related to their collaborators
- ▶ Draw a UML class diagram showing how
- ▶ Common relationships:
 - **Inheritance**: only when subclass **is a** special case
 - **Aggregation**: when one class **has a** field that references another class
 - **Dependency**: like aggregation but transient, usually for method parameters, **“has a” temporarily**
 - **Association**: any other relationship, can label the arrow, e.g., **constructs**

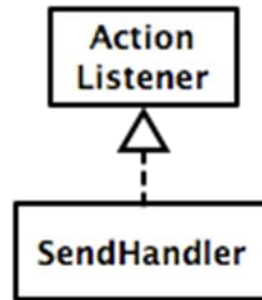
NEW!

Summary of UML Class Diagram Arrows

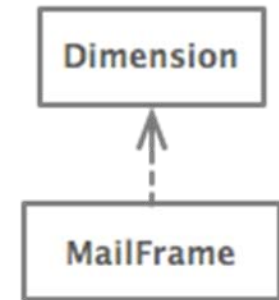
Inheritance
(is a)



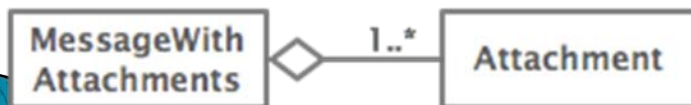
Interface
Implementation
(is a)



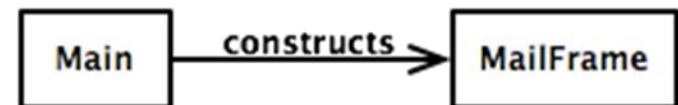
Dependency
(depends on)



Aggregation
(has a)



Association



Homework 18

- ▶ Finish BallWorlds by Friday at 2 PM.
- ▶ Do Appointment Calendar design exercise before Monday's class
 - Work with one, two, or three people (from either CSSE 220 section) if you wish
 - You will need a few index cards.
 - If you need to buy them from the Rose–Hulman bookstore, you will need to do so by Friday at 4:30 PM.
 - If you forget, find someone else who remembered!
 - Pay them for the cards. (Cards should be going at about \$1.00 each by Sunday at midnight!)
 - Use Violet for drawing your diagrams
 - Instructions for installing it are at <http://www.rose-hulman.edu/class/csse/resources/>

BallWorlds Work Time

»» Ask questions if you're stuck!