Building a System

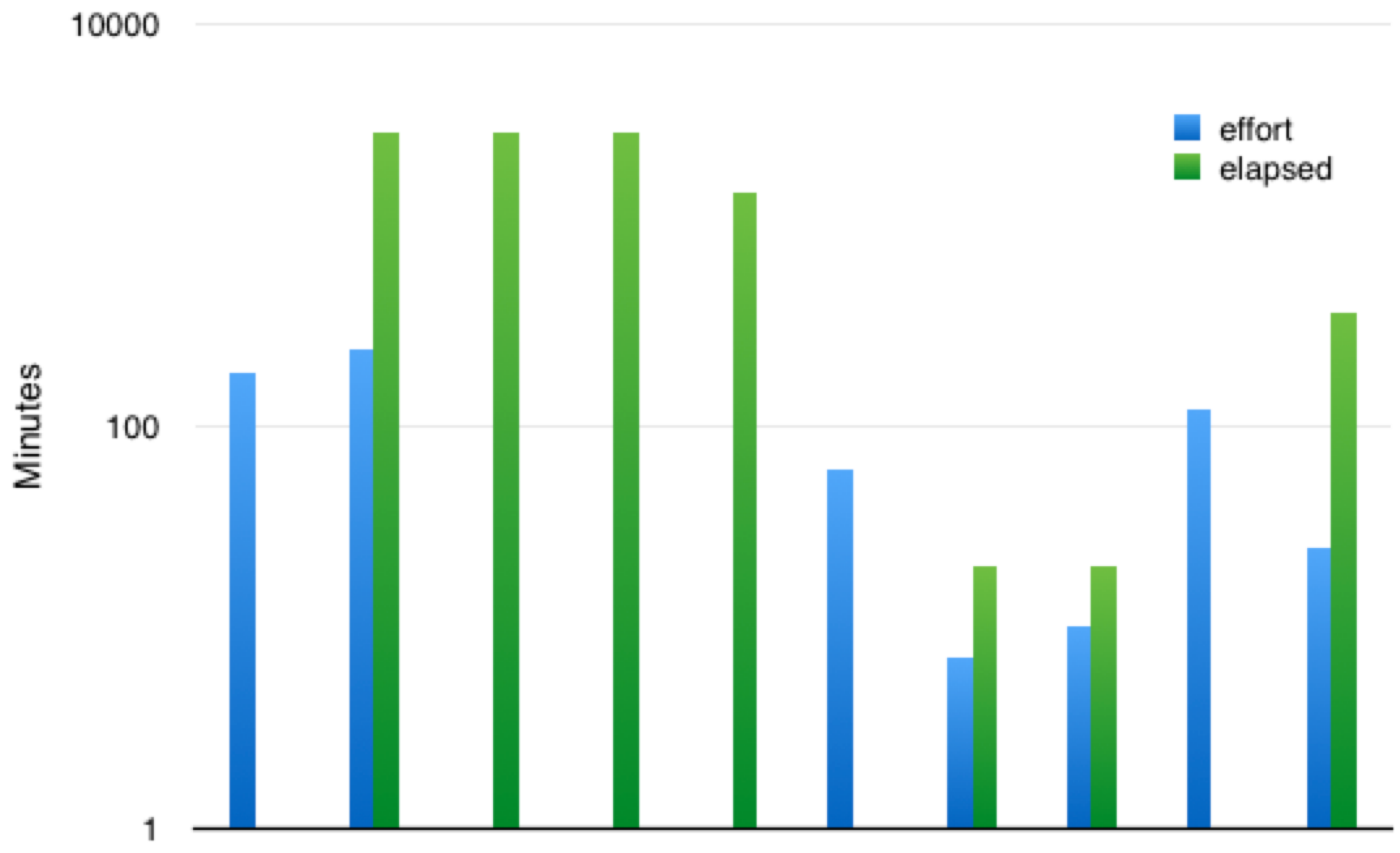
Software Engineering
CS 130
Donald J. Patterson

Content adapted from Essentials of Software Engineering 3rd edition by Tsui, Karam, Bernal Jones and Bartlett Learning

Prof. Patterson Experiment

$\boxed{\text{I estimated 30 minutes}}$ to do the task

- 10:00 - started
- 10:15 - Eclipse crashed
- 10:30 - Decide initial design was bad
- 11:00 - laptop battery died - no charger - stop
- 13:00 - restart
- 13:21 - done debugging
- 13:36 - done writing tests
    - effort ~ 1.6 person hours
    - $\boxed{\text{elapsed time ~ 3.5 hours}}$



- ideal time or effort: straight through with no interruptions
  - units: e.g., person-hours, person-days, etc.
- elapsed time or duration: actual calendar time including everything
  - units: e.g., days, weeks, etc.

Building a System

Moving from
writing a program to building a system

https://flic.kr/p/a8724x

Building a System

Moving from
writing a program to building a system

https://flic.kr/p/r3meit

Building a System

Moving from
  writing a program to building a system

- What's the difference?

https://flic.kr/p/a8724x

https://flic.kr/p/r3meit

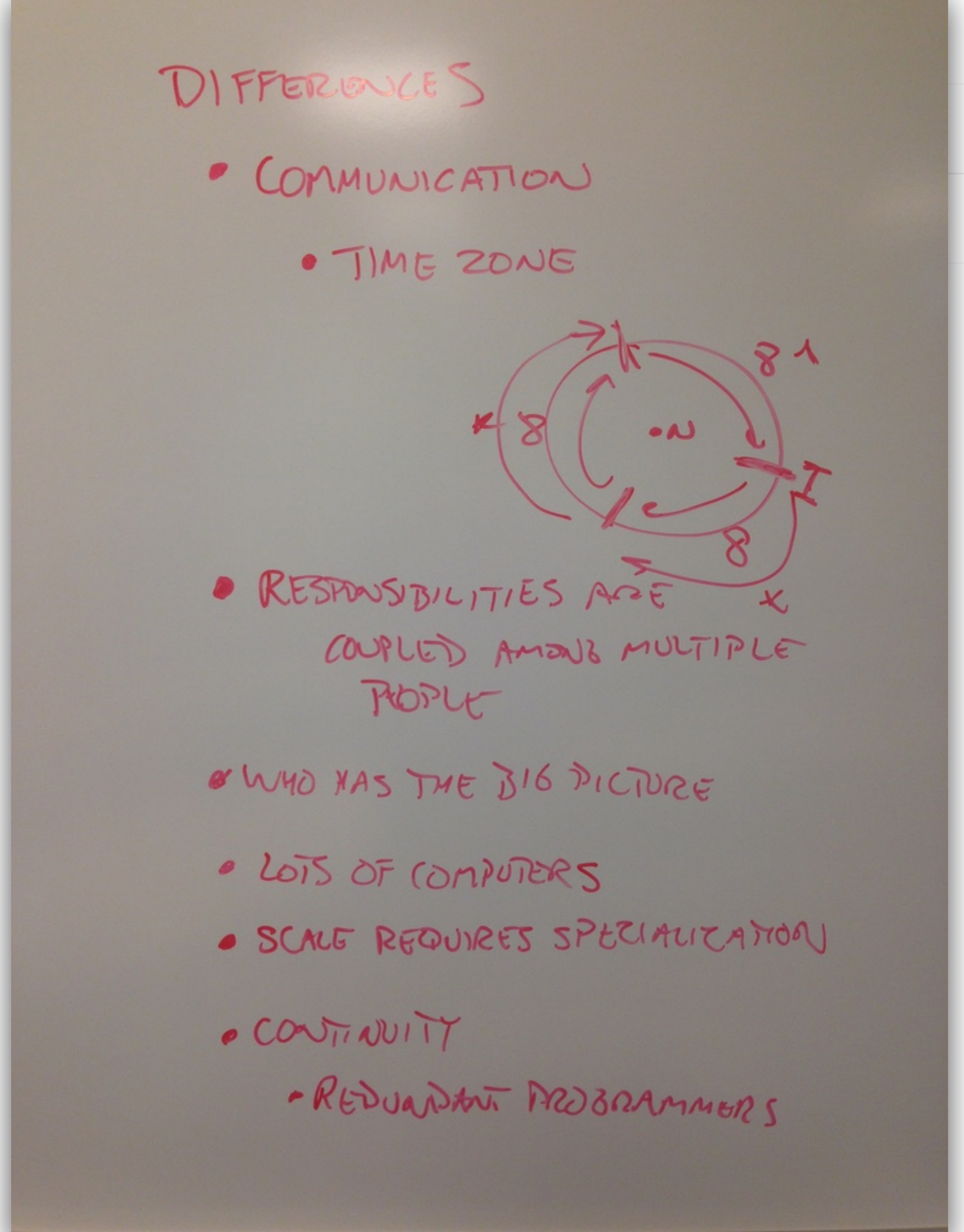
Building a System

Moving from
writing a program to building a system

- What's the difference?

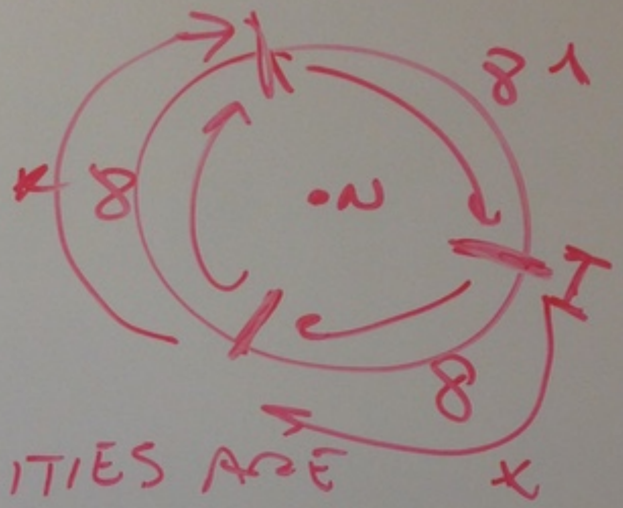
- Size, which only matters because of increased complexity

https://flic.kr/p/a8724x https://flic.kr/p/r3meit



DIFFERENCES

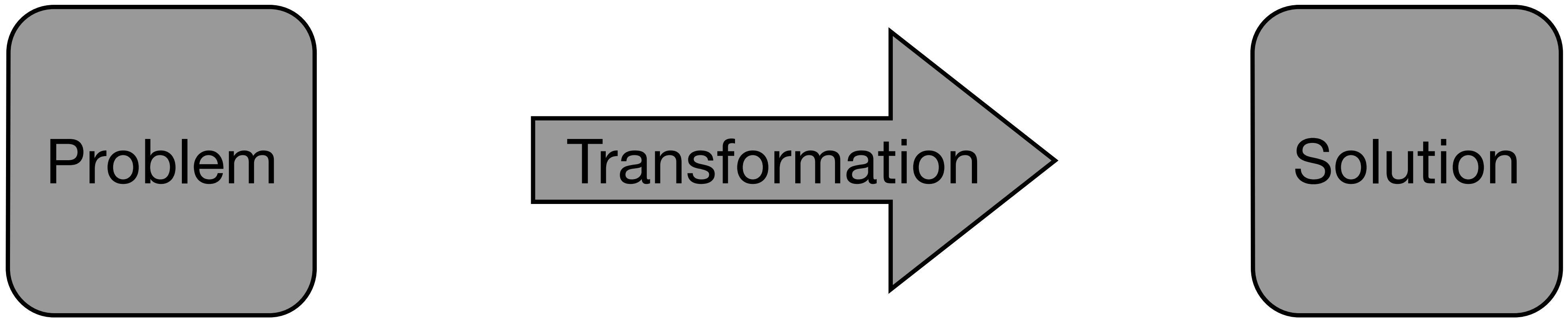
- COMMUNICATION
  - TIME ZONE



- RESPONSIBILITIES ARE COUPLED AMONG MULTIPLE PEOPLE
- WHO HAS THE BIG PICTURE
- LOTS OF COMPUTERS
- SCALE REQUIRES SPECIALIZATION
- CONTINUITY
  - REDUNDANT PROGRAMMERS

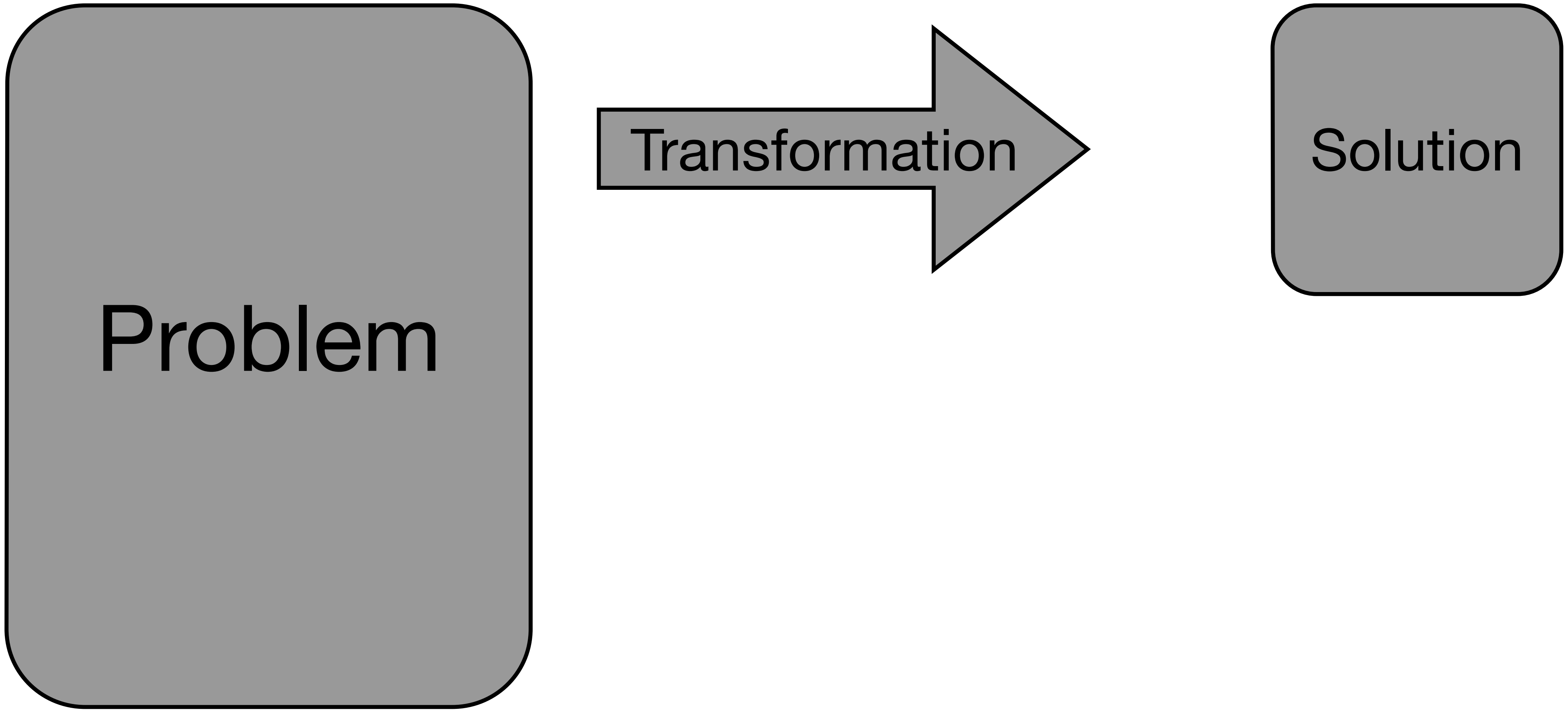
Building a System

Complexity Increases Everywhere



Building a System

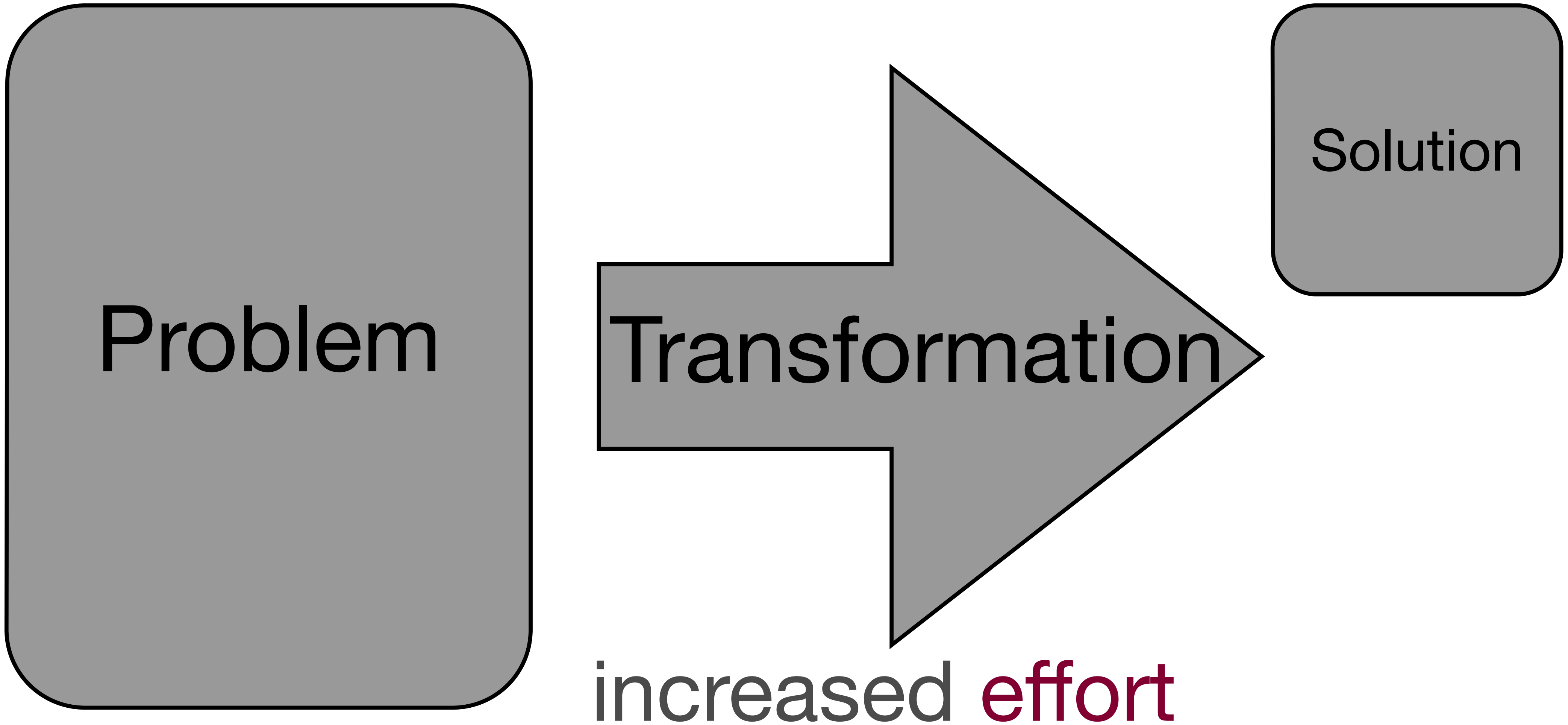
Complexity Increases Everywhere



increases in size
and complexity

Building a System

Complexity Increases Everywhere

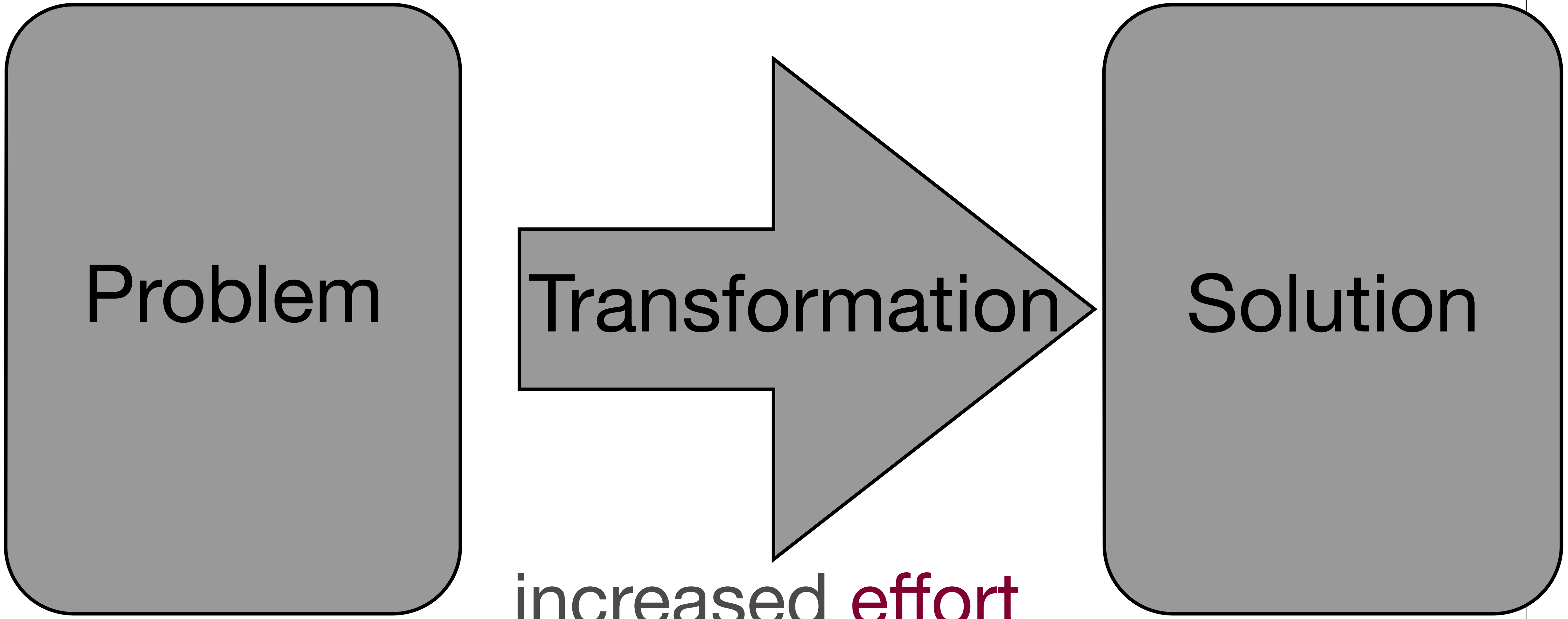


increased effort due to size and complexity

increases in size and complexity

Building a System

Complexity Increases Everywhere



increases in size and complexity

increased effort due to size and complexity

increases in size and complexity

Building a System

Complexity

- Breadth
  - The sheer number of issues to be addressed
    - More major functions
    - More features in each functional area
    - More varieties of interfaces to users, internal and external systems
    - More simultaneous users, more types of users
    - More data, types of data and data structures

Topical Heading

For our Assignment 1

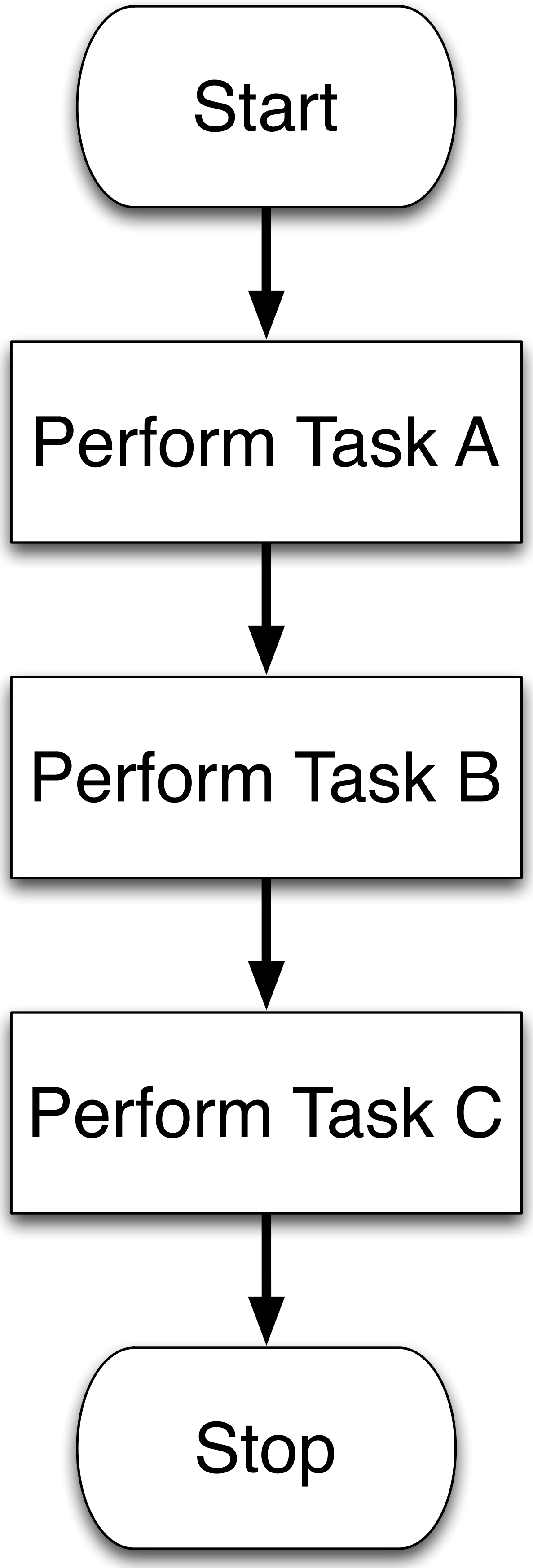
- What is it again?
- How would our solution change if the input size was increased to 1 trillion?
- How would our solution change if the numbers were very large?

Building a System

Complexity

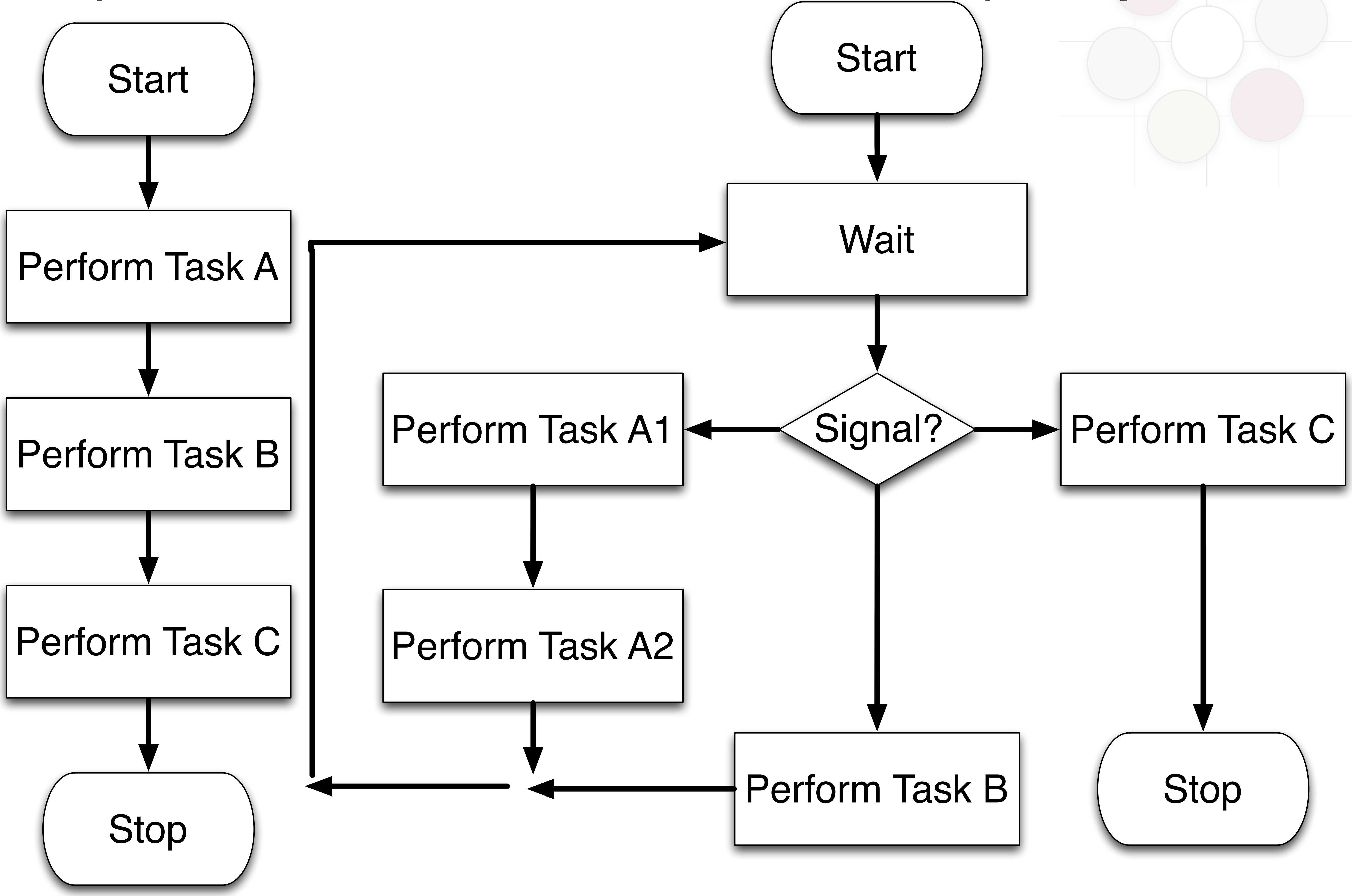
- Depth
  - More linkages and connections
    - Data sharing among the functionalities & logic
    - Control Passing among functionalities

Simple Task



Simple Task

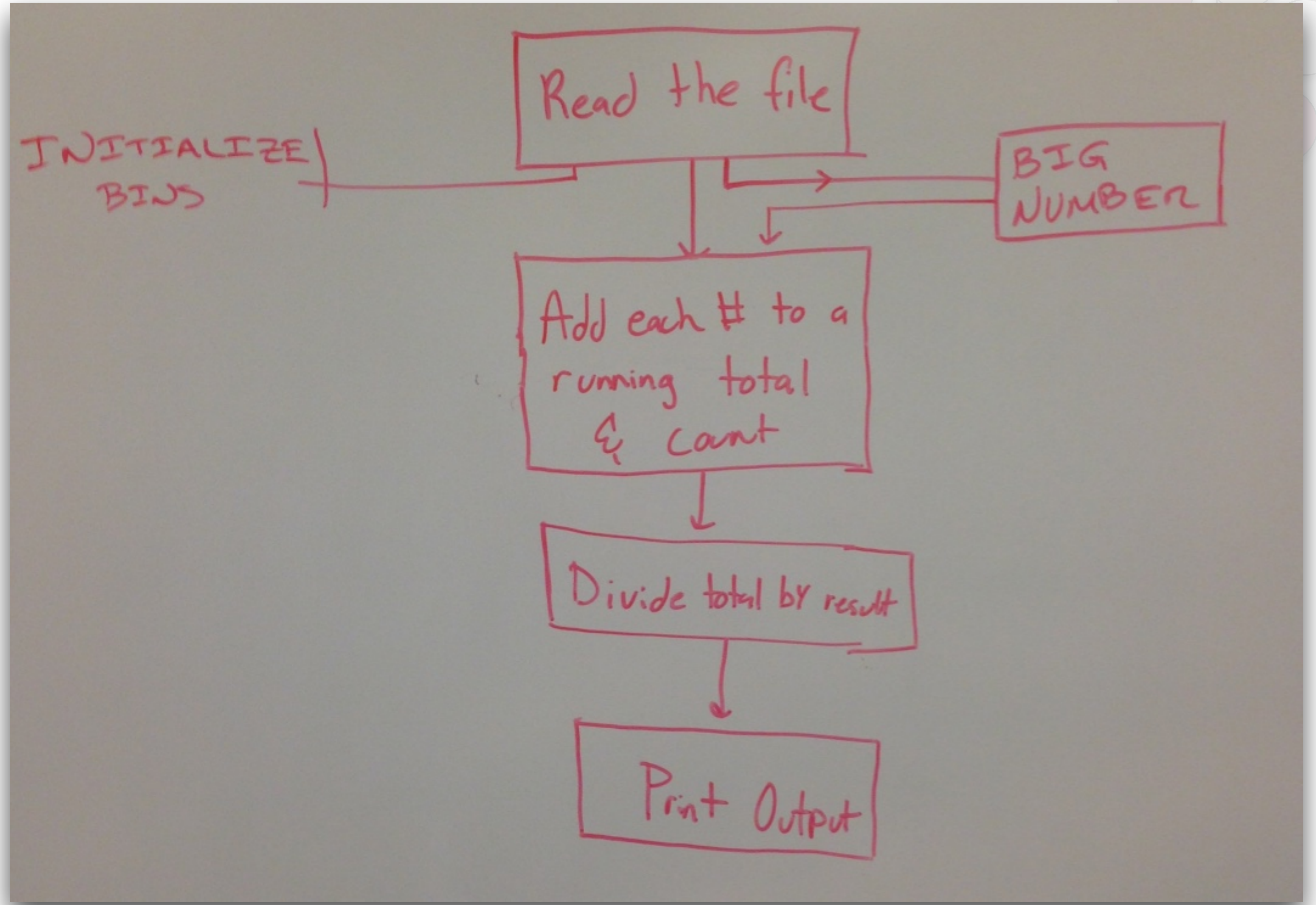
Increased Complexity



Building a System

Board work - Modified Assignment 1

- Compute and show the average of the read-in numbers
- Additionally show the largest and smallest of the read-in numbers
  - Where is the increased complexity?
- Additionally show the numbers in sorted ascending order
  - Where is the increased complexity?


Read the file
INITIALIZE BINS
BIG NUMBER
Add each # to a running total & count
Divide total by result
Print Output

Building a System

Handling complexity

- Strategy 1: Simplification
  - Decomposition of the problem and the solution
  - Modularization of the solution
  - Separation of concerns of the problem and the solution
  - Possibly reduce the problem
  - Incrementally address the problem components

Building a System

Handling complexity

- Strategy 2: Better technology and tools
  - Database to handle information and structures of information
  - Programming and development platforms
  - Computing network
  - Multi-developer configuration management
  - Modeling techniques
  - Automated testing

At first this doesn't seem to be reducing complexity

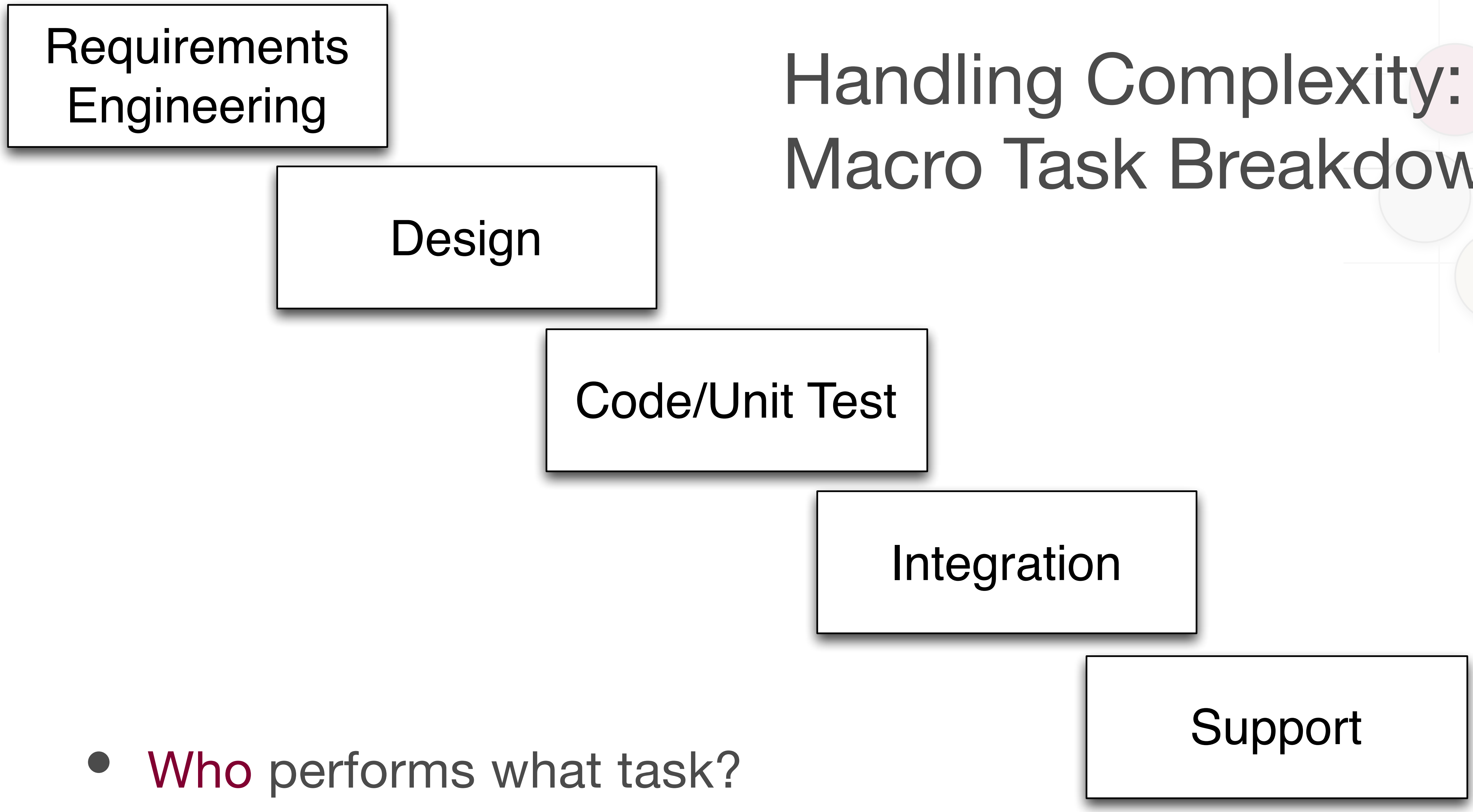
Building a System

Handling complexity

- Strategy 3: Improve process and methodology
  - Coordinate multiple and different people performing different tasks
  - Guidance for overlapping incremental tasks
  - Guidance for measuring separate artifacts and outcomes

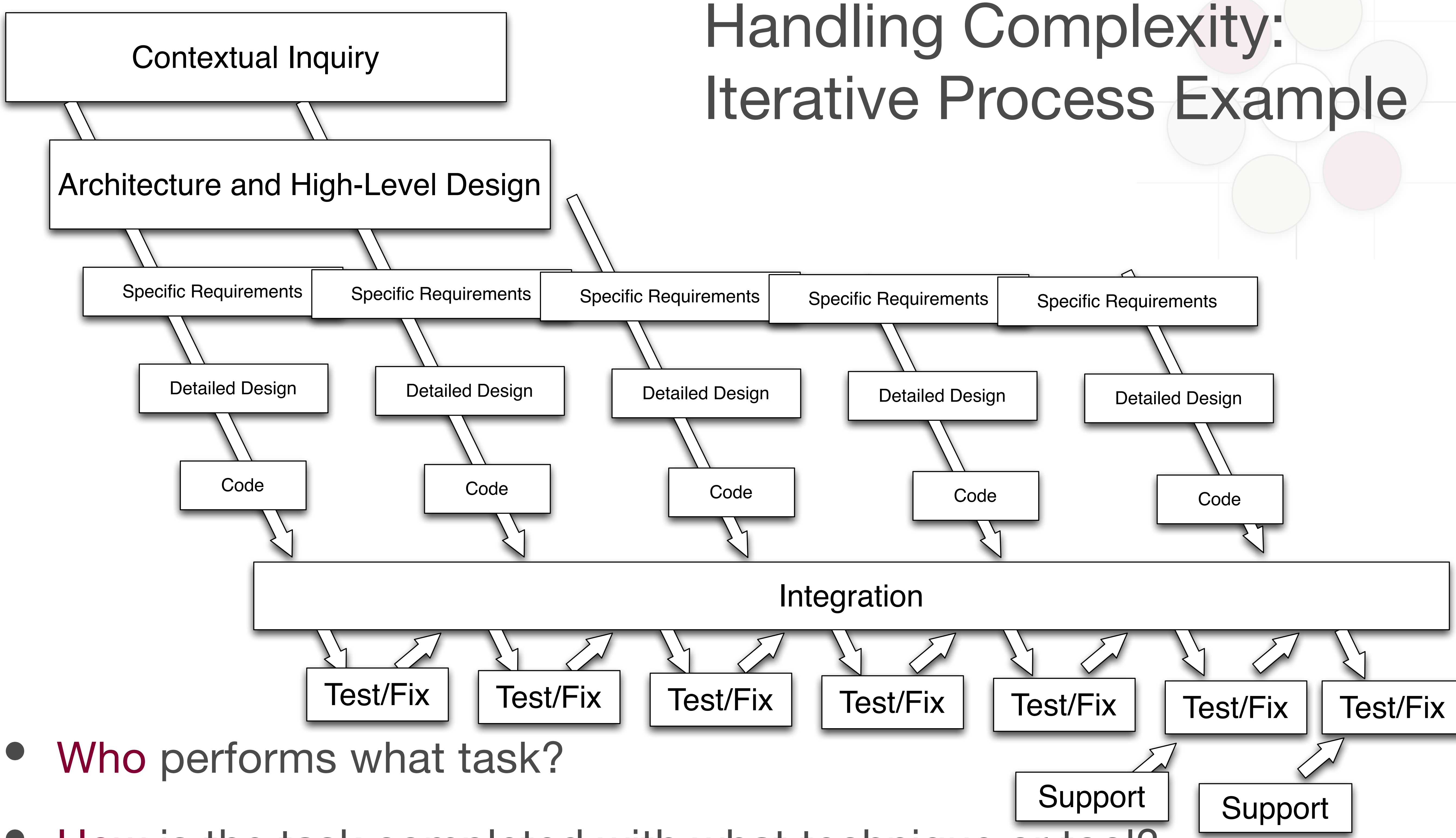
Again at first this doesn't feel like it is reducing complexity

Handling Complexity: Macro Task Breakdown



- Who performs what task?
- How is the task completed with what technique or tool?
- When should which task start and end?
- Who should coordinate the people and the tasks?

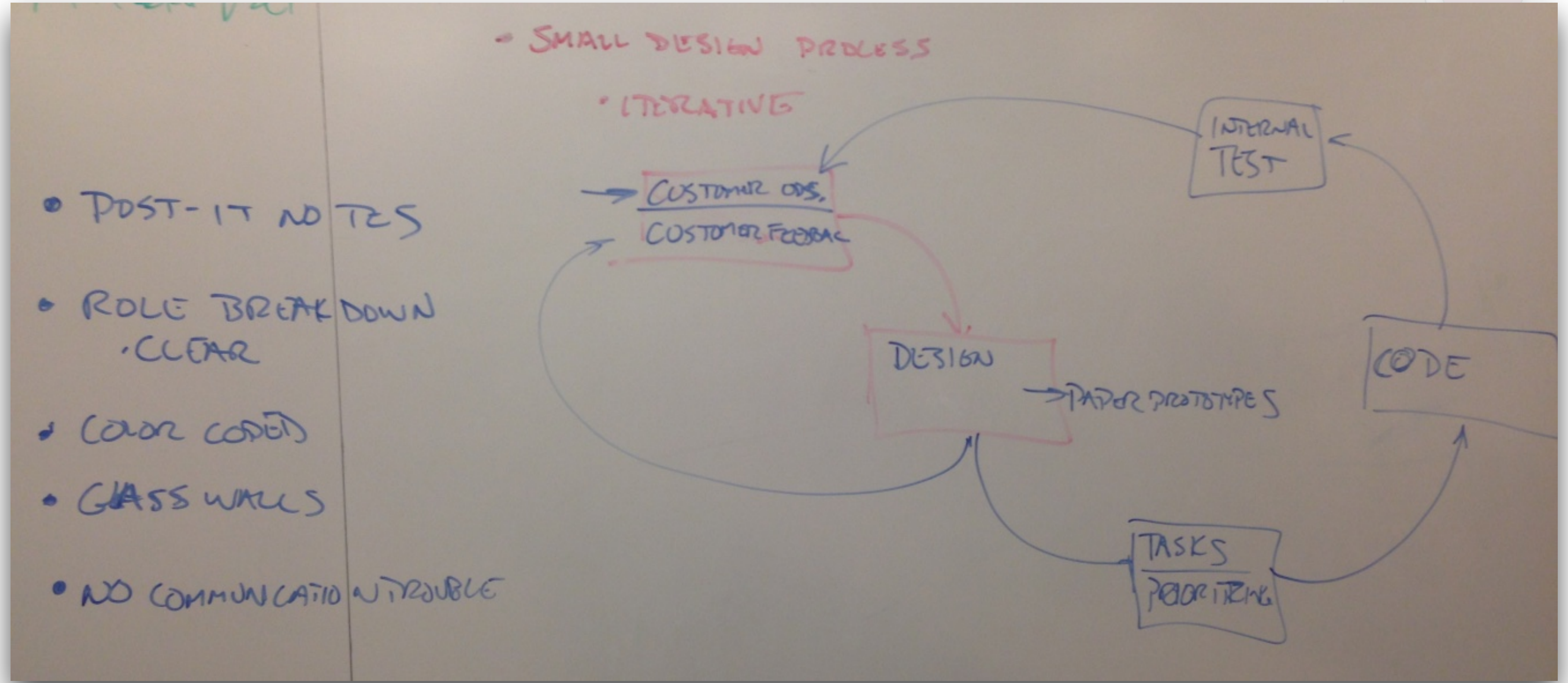
Handling Complexity: Iterative Process Example



- Who performs what task?
- How is the task completed with what technique or tool?
- When should which task start and end?
- Who should coordinate the people and the tasks?

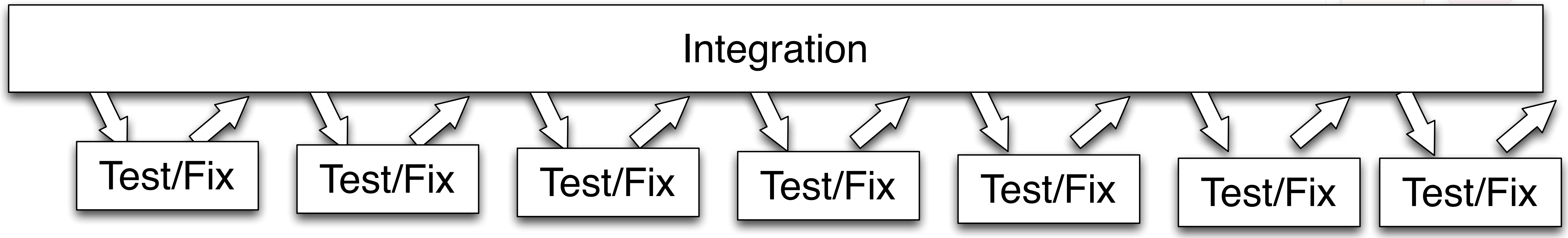
NORDSTROM

INNOVATION LAB


- POST-IT NOTES
- ROLE BREAKDOWN
  - CLEAR
- COLOR CODED
- GLASS WALLS
- NO COMMUNICATION TROUBLE

- SMALL DESIGN PROCESS
  - ITERATIVE

Handling Complexity: Separating out the details is not trivial



- Seemingly “simple” Test/Fix and Integrate steps:
  - Should there be separate & independent test group?
  - How should problem be reported and to whom?
  - How much information must accompany a problem report?
  - Who decides on the priority of the problem?
  - How is the problem fix returned?
  - Should all problems be fixed?
  - What should we do with non-fixed problem?
  - How are fixes integrated back to the system

Building a System

Non-Technical Considerations for Developing and Supporting a System

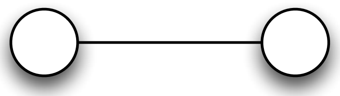
- Effort & Schedule Expansion
  - How does one estimate and handle this?

- Assignment and Communications Expansion?
  - Do we need some process?
  - Do we need some tools?

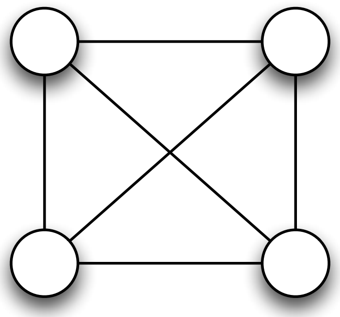
Building a system

Increased complexity means increased human resources

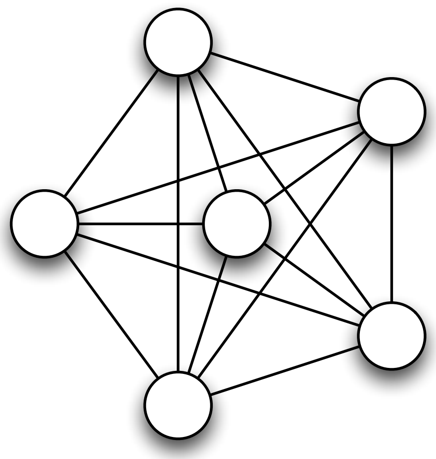
$$\frac{(n)(n-1)}{2}$$



2 people
1 path



4 people
6 paths



6 people
15 paths

Consider communication errors as well

Building a system

A Large, Complex System

- Building “Mission critical” or “Business critical” system (e.g. payroll) requires (1) several separate activities performed by (2)more than 1 person (e.g. 50 ~ 100):
  - Requirements: gathering, analysis, specification, and agreement
  - Design: abstraction, decomposition, cohesion, interaction and coupling analysis
  - Implementation: coding and unit testing
  - Integration and tracking of pieces and parts
  - Separate testing: functional testing, component testing, system testing, and performance testing
  - Packaging and releasing the system

Building a system

Need to support the system (for real production)

- Pre-release: preparation for education & support:
  - Number of expected users
  - Number of "known problems" and expected quality
  - Amount of user and support personnel training
  - number of fix and maintenance cycle

- Post-release: preparation for user and customer support:
  - Call center and problem resolutions
  - Major problem fixes and code changes
  - Functional modifications and enhancements

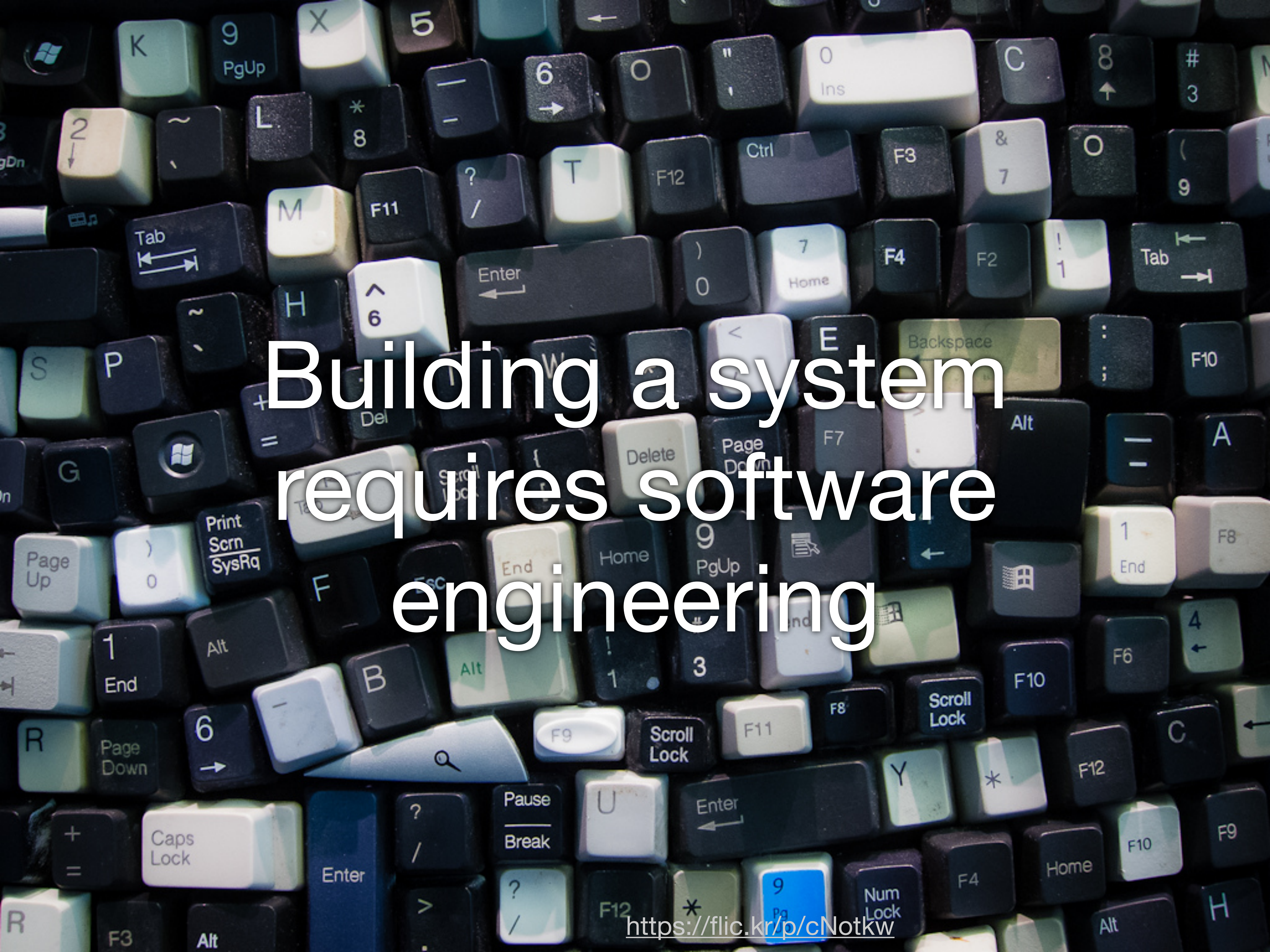
https://flic.kr/p/eeQ512

https://flic.kr/p/dx3QAu

Building a system

Coordination Efforts Required in Systems Development and Support

- Because there are
  - more parts,
  - more developers
  - more users to consider in “Large Systems” than a single program developed by a single person for a limited number of users, there is the need for Coordination of (3P’s):
    - ‘Processes’ and methodologies to be used
    - Final ‘product’ and intermediate artifacts
    - ‘People’ (developers, support personnel, and users)



Building a system requires software engineering

https://flic.kr/p/cNotkw

WESTMONT COMPUTER SCIENCE