UNIVERSITY OF SAN FRANCISCO
CHANGE THE WORLD FROM HERE

# Principles of Software Development

Sami Rollins

## Welcome to CS 601!

- **Learn to think like a software developer**
  - Modular design
  - Reusable code
  - Problem solving

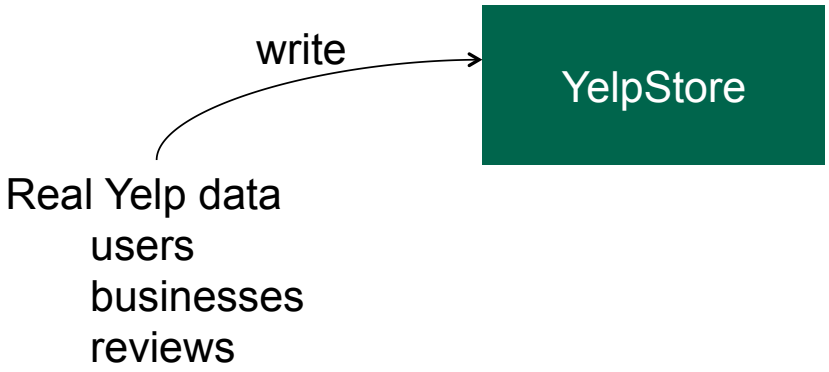
- **Mechanics**
  - Concurrency
  - Networking
  - Web and HTTP
  - Distributed topics

# A practical approach


- Learn by doing – build a user review web application!
- Labs – practice fundamentals
- Project – get creative!

# Lab 1

- **Practice with data structures**

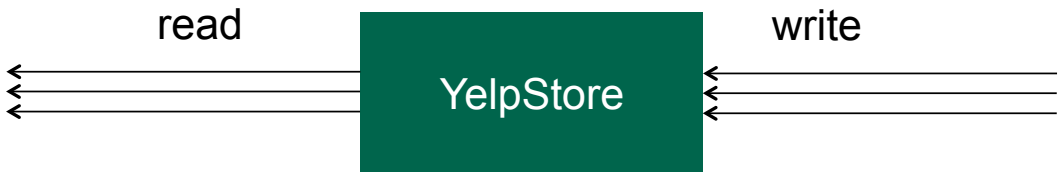


## Lab 2

- **More practice with data structures**

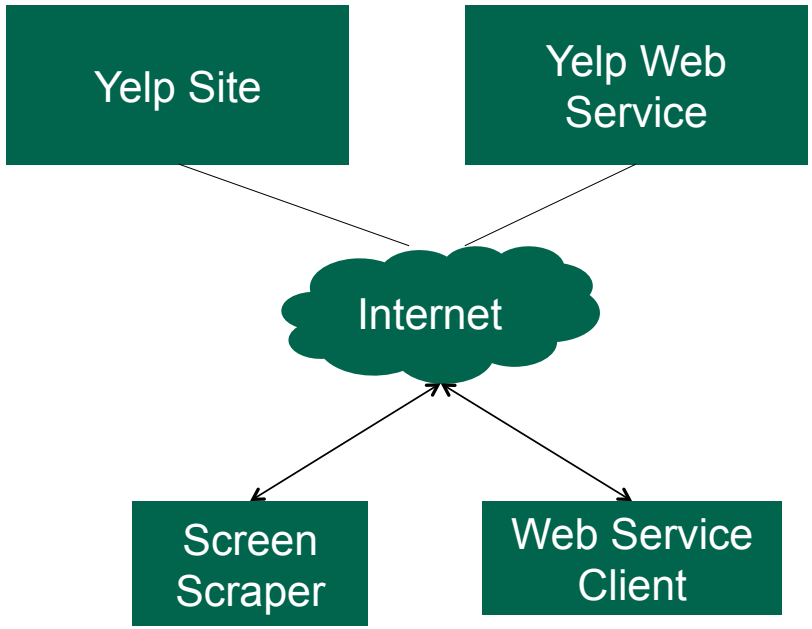
## Lab 3

- **Concurrency and multithreaded programming**



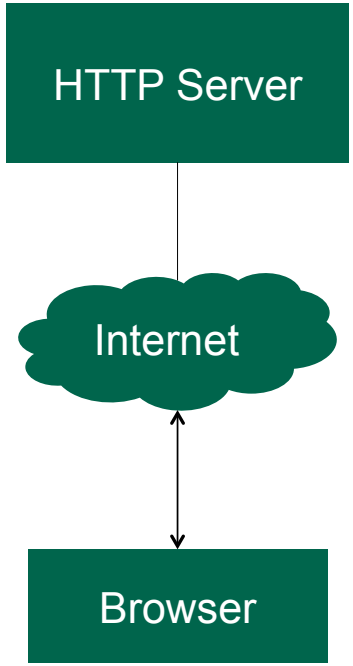
## Lab 4

- **Web clients**
  - Screen scraper
  - Web service client



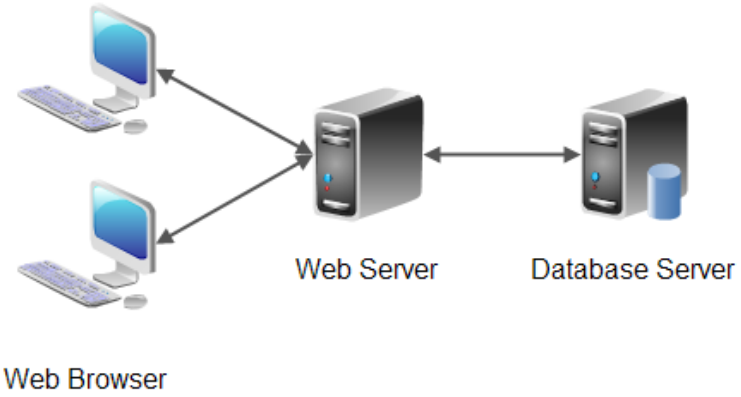
## Lab 5

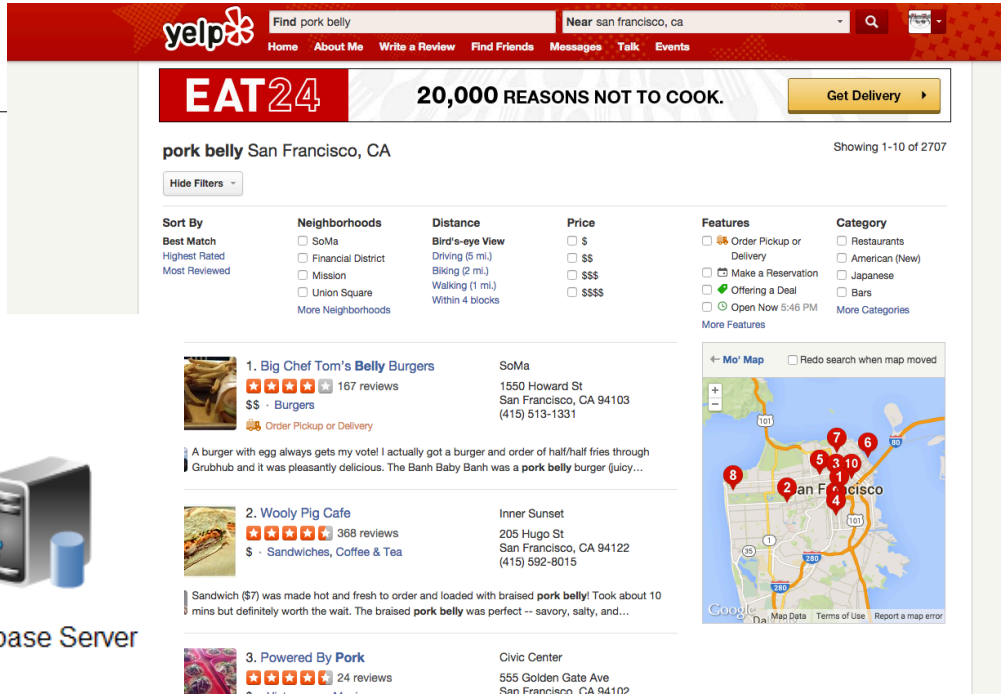
- **HTTP Server**
  - Using raw sockets!



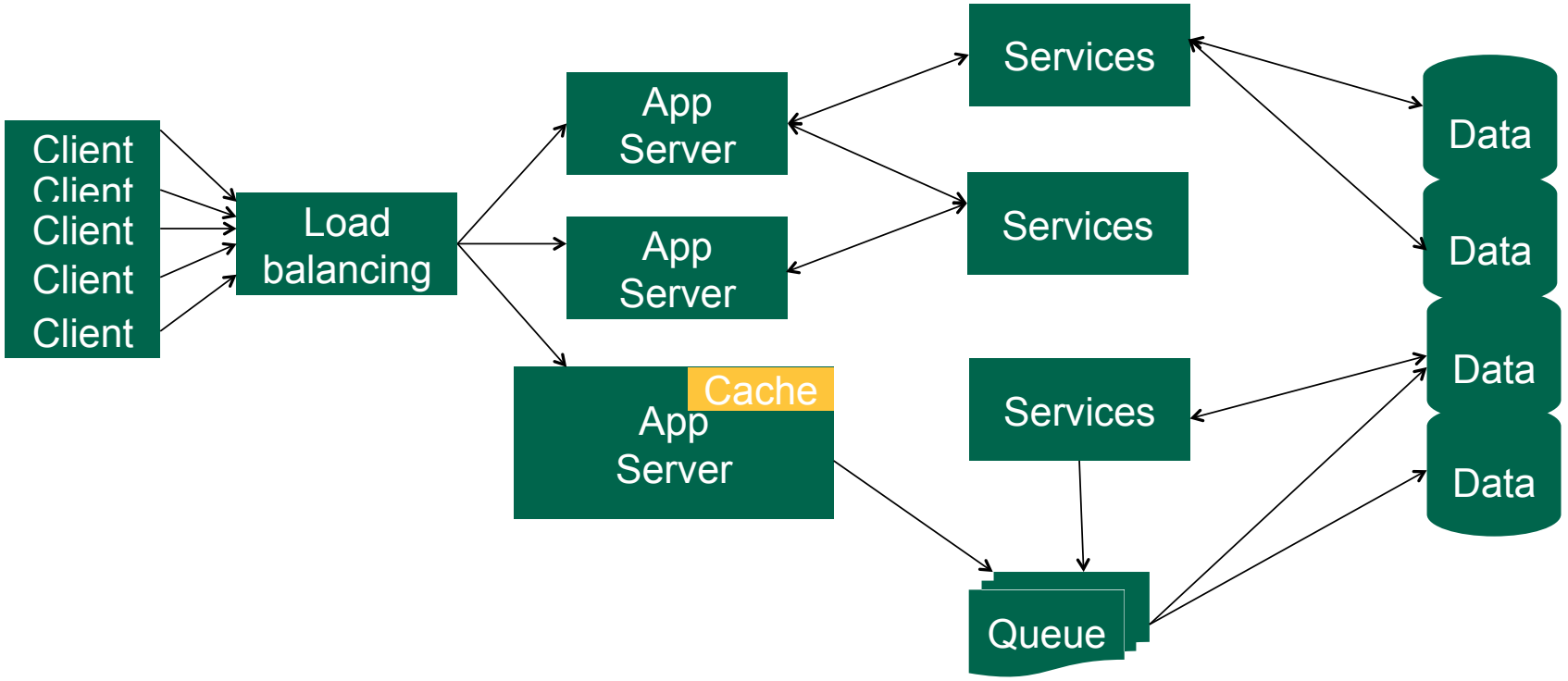
# Project

- **Fully functional website**

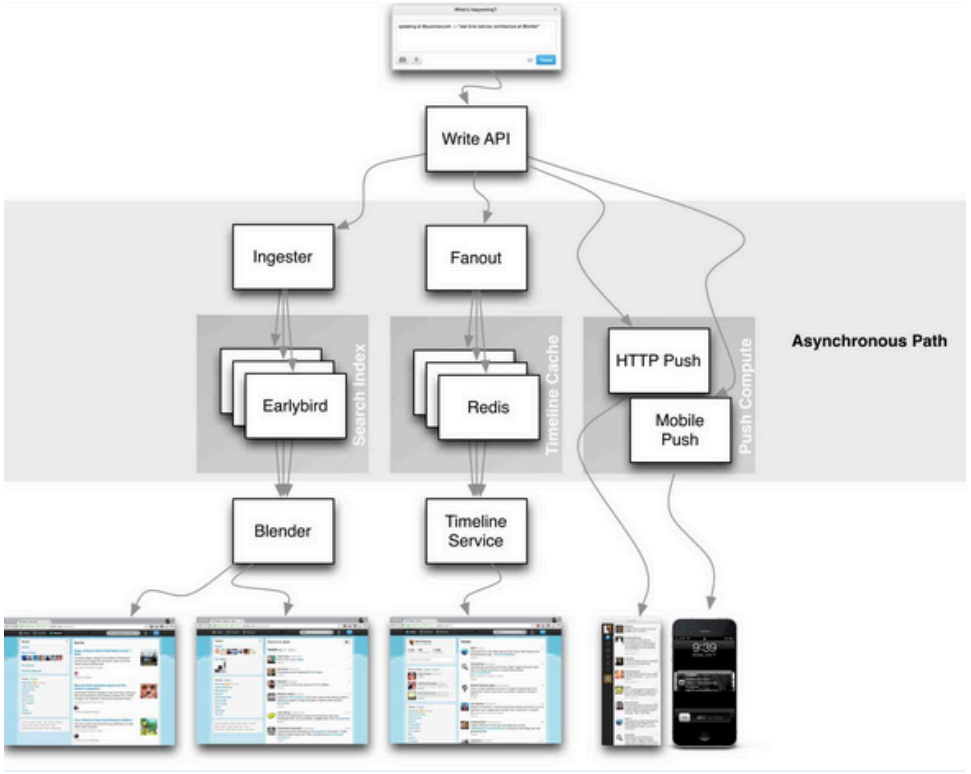


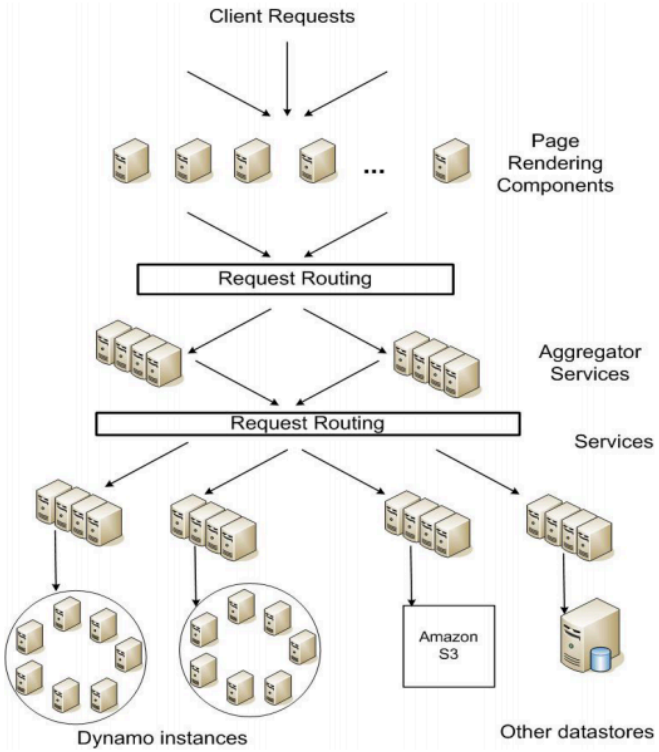
## Scaling


Client
Client
Client
Client
Client
Load balancing
App Server
App Server
App Server
Cache
Services
Services
Services
Queue
Data
Data
Data
Data

# Twitter – Service Oriented Architecture (SOA)



# Amazon – Original SOA



## Software Engineering

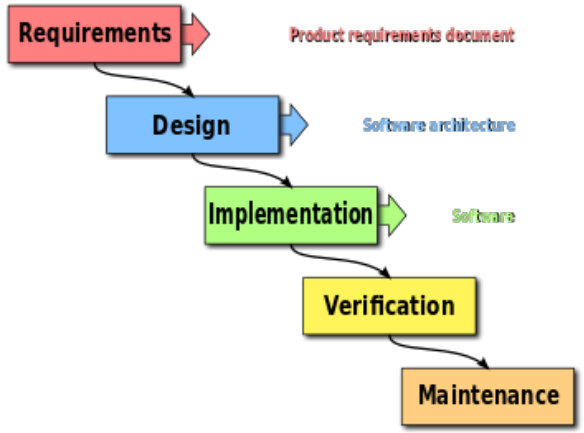
- **There are many examples of failed software projects**
  - Affordable care act website
  - Therac-25 radiation therapy machine killed patients because of a bug

- **Software Engineering**
  - Term coined in 1969. Discover more structured methods for building software.

- **Also see "Engineering Software as a Service: An Agile Approach Using Cloud Computing" by Fox and Patterson**

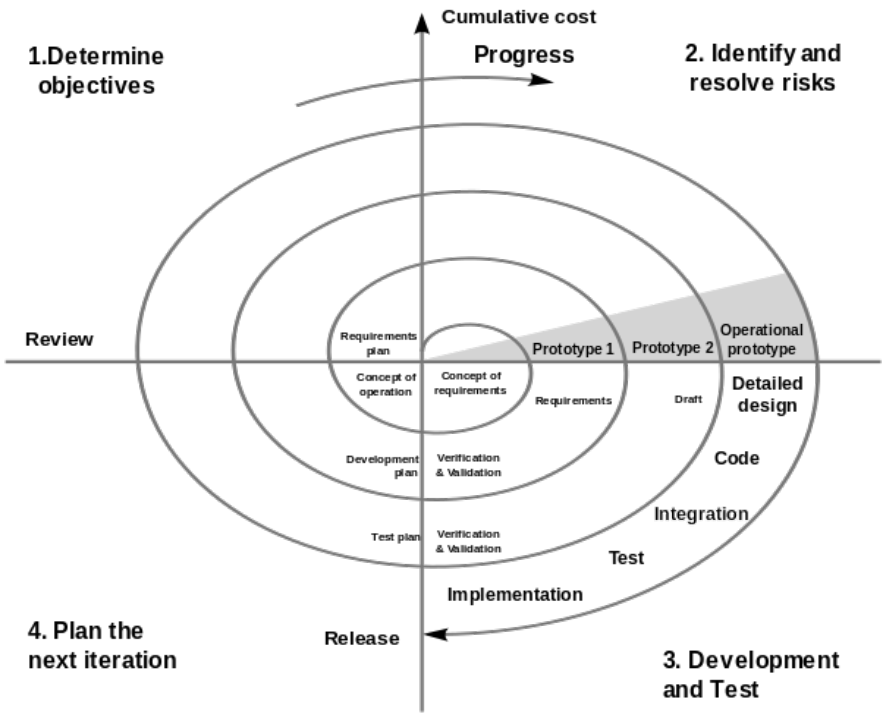
## Waterfall – 1970s

- **Each phase happens once**
- **Good for projects that require a lot of planning**
  - NASA applications
- "*Plan to throw one [implementation] away; you will, anyhow.*" -Fred Brooks, Jr.
- **Need user/client in the loop**
  - Early prototypes



# Spiral – 1980s

- Develop prototypes
- Consult client
- Iterate
- Iterations 6-24 months long



## Rational Unified Process – 2003

- **Four phases**
  - Inception
  - Elaboration
  - Construction
  - Transition

- **Each phase may have multiple iterations**

## The Agile Manifesto – 2001

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

## Exercise

- **Identify the ten applications you think are most important.**
- **For each, do you think agile would be an appropriate software development methodology?**