Intro & Retrospective

October 20, 2025

Some Things I've Learned

- Challenge assumptions & be facts-based.
- Document everything, always.
- Being right is awesome, but it's OK to be wrong.
- Specialize in generality. Stay curious.
- No one knows what they're doing.
- Ask "dumb" questions.
- Just because you can, doesn't mean you should.
- KISS.
- Play nice.
- Work to live, don't live to work.

What Software Does

October 20, 2025

What Software Is

- Software = a set of instructions that tell hardware what to do
- Hardware handles electricity; software gives it logic
- Software lives everywhere: laptops, phones, routers, cars, thermostats

Software Is Everywhere

- Laptops
- Phones
- Routers
- Cars
- Watches
- Each device runs multiple layers of software

Name 3 things you've used today that run software

Input/Process/Output Model

- Input: data from user or device
- Process: logic or calculations
- Output: result shown, stored, or sent elsewhere
- Examples:
  - Calculator: numbers -> math -> result
  - Weather app: location -> fetch forecast -> display temp

Can you think of other examples?

Categories of Software

- System software
  - Operating systems (Windows, macOS, Linux, Android, iOS)
  - Device drivers, firmware
  - Manages memory, CPU, files, networks
- Application software
  - Productivity tools (Google Docs, Excel)
  - Games, browsers, chat clients
  - User-facing and purpose-specific
- Embedded software
  - Hidden programs inside routers, cars, washing machines
  - Usually fixed-purpose, lightweight, real-time
- Service-oriented or “cloud” software
  - Runs on remote servers; accessed via Internet
  - Examples: Gmail, Netflix, Zoom, Spotify
  - More about this tomorrow!

Anatomy of a Software System

- User interface (UI): what you see and touch
  - Screens, buttons, menus, web forms
- Application Logic: rules, decisions
  - If this, then that; calculations; workflow
- Data layer: where information lives
  - Databases, files, external APIs

Separating these layers lets developers work in parallel & makes updates safer

Example: Food Delivery App

- UI = map + order button
- Logic = match restaurant, compute fees
- Data = menus, user profile, payment

Where does each layer live?

How Software Fails

- Bugs: wrong logic
- Bad data: wrong results (GIGO)
- Outages: unreachable services or failed components
- Humans: configuration mistakes, process breakdown
- Good software design includes planning for failure and recovery

Key Takeaways

- Software = instructions for hardware
- Everything follows: input -> process -> output
- Most systems have UI, logic, data layers
- Reliability & clarity > buzzwords (KISS!)

- Next: how programs communicate (clients, servers, APIs)

Project: Reverse Engineering a Favorite App

- Functional Breakdown
  - What does the app do for the user?
  - List main functions (browse, post, message, etc.).
- Identify the Software Layers
  - UI: what the user sees and interacts with
  - Logic: what the system decides or computes
  - Data: what is stored or retrieved
  - Sketch a block diagram on paper or whiteboard
- Data Flow Discussion
  - Trace a single user action—like “sending a message.”
  - Identify what happens behind the scenes.
- Mini-Presentations
  - Each group presents its diagram (5 min each)