Software Development

Goals

What We Will Cover

- What is software development?
- How is software built?
  - What are the steps and order?
  - Who does what?
  - What are common Software Methodologies?
- What are the different roles in software development?
- When we will cover different software engineering roles?

What is Software Development?

The Definition

Software development is the process of creating, designing, testing, and deploying software for customers to use.

> **Note: The Definition Varies**
>
> There are many different variations of this definition, and most companies will handle software development in their own way, but almost all companies will have the above steps as part of their process.
>
> - See IBM's definition https://www.ibm.com/topics/software-development
> - See Wikipedia's definition https://en.wikipedia.org/wiki/Software_development

Common Misconceptions

- There is one way to build software
- It's only for people who like math / science
- You can work in a silo
  - Communication isn't important
- You do it once, and then it's finished

🎨 Software is Art

"You can mass produce hardware; you cannot mass-produce software - you cannot mass-produce the human mind."

– Michio Kaku, Theoretical Physicist

- You don't need to be a "math person" to build software or write code
- You do need to be able to deal with ambiguity and not understanding every aspect of what you're building
- You do need to be detail-oriented and excited at the prospect of your code working
  - Even if your code does something kind of mundane, like adding two numbers together
- Writing code is more like writing an essay than solving a math problem

⚽ Building Software is a Team Sport

"Without requirements or design, programming is the art of adding bugs to an empty text file."

– Louis Srygley, Software Architect

- Much of the complexity of building software is in figuring out
  - What to build
  - The best way to build it
- Coders are one part of that equation

How is Software Built?

The Basic Process

1. Planning
2. Analysis of Requirements
3. Design
4. Implementation
5. Testing & Integration
6. Maintain

Rinse and repeat for additional software features

Common Software Methodologies

- Waterfall
- Agile
- Feature-Driven Development
- Lean
- RAD

Note: Check out these links for more information.

https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/ https://blog.planview.com/top-6-software-development-methodologies/

Software Stacks

Most Common Languages

Often easier to learn & understand

- Python
- JavaScript
- Ruby

More syntax rules to learn & follow

- Java
- C, C++

Other Languages

- Go, Rust: opinionated, newer back-end languages
- Swift: specific to iOS development
- Scala: functional programming language
- Kotlin: multi-platform (mobile + web) Java relative
- And so many more!

What is a stack?

A **tech stack** is a set of technologies that work together to form the core languages and frameworks of a project, application, or product.

A stack usually encompasses:

- Database
- Web framework (Back-end)
- Front-end framework
- Back-end language / environment / server

MEAN/MERN Stack

“Full Stack” JavaScript: JavaScript on the back-end and front-end

M ongoDB: A NoSQL database

E xpress.js: A web framework

A ngular.js: A front-end framework / **R** eact.js: Another front-end framework

N ode.js: A back-end “language” / server

Typical Python Stack

- Flask or Django: Web frameworks
- React.js: A front-end framework
- PostgreSQL: A database
- Python: Back-end language

Companies and Technologies

- Most companies have primary languages / technologies they use
- But, it's common that a tech company has products in multiple languages and stacks
- Sometimes, companies undergo **migrations**, or large conversion projects from one language to another

Developers and Technologies

- Everyone learns a single stack to start
- Learn core concepts, things that are common between tech stacks
- First job might be in the stack you learned
  - Or, the company might help you learn their stack
- Over your career, you'll certainly learn new stacks/languages for different jobs
- **Learning how to learn is important**

What are the roles in software?

Engineering Roles

- QA Engineer
- Mobile Developer (Android, IOS, hybrid platforms)
- Software Developer (Works in various areas)
- Web Developer (Full-Stack, Front-End, Back-End…)
- DevOps Engineer
- Security Engineer

The Software Team

- Project Manager
- Engineering Manager
- Tech Lead (maybe)
- Business Analyst (maybe)
- Developers / Engineers
- Architect

Other Roles

- Data Analytics
  - Work with business or engineering to make data-driven decisions
- Developer Operations / Production Engineering
  - Ensure that code is successfully deployed & installed
  - Ensure code stays working on the public-facing side of the product
- Security Engineering
  - Build tools and enforce best practices around data privacy
  - Ensure systems are protected against malicious actors, etc.

Web Developer? Software Engineer?

- Huge overlap
- Some unique aspects to each

Web Development

- Web development is a type of software engineering
  - Full Stack JavaScript (React + Express/Node) is most common
- Lots of patterns (like recipes)
- Requires specialized knowledge and skills
- Complex in it’s own right
- Focused on a very specific outcome: web applications!

Software Engineering

- Very broad term
- Very broad skillset
  - Many languages
  - Many possible products
  - Might work on web application or a small specific part of a huge system architecture
- Almost always involves web development
  - Need to know web development patterns

Depends on company's product

- In many cases, core product is a web application
- In other cases, core product is simply **software**
  - Software to build web applications
  - Software that's part of an operating system
  - Software to build other software
  - Software to monitor the efficiency of web applications
  - etc. etc. etc.

Coming Up

- Running Code
- Making Decisions 1 and 2
- Agile Theory

Feedback

- Please fill out the feedback survey

https://docs.google.com/forms/d/e/1FAIpQLSeGdmaqE8kPK4SXppXLN1ThQg_bf1ZG8t9BI3NWDePPMDn2VQ/viewform?usp=pp_url&entry.432752290=Software+Development+and+Orientation+(Monday)

- Thank you!