



Software engineering

Workshop

Preparing for a Successful Software
Engineering Career!

01

Intro to SE

What exactly is Software Engineering?



What is SE

Not just coding or developing, but the whole process from A-Z when delivering a product.



Software engineering:

- 
- 
- 01 — Definition** — The application of engineering principles to the development, operation, and maintenance of software.
 - 02 — Objective** — To produce high-quality software that is reliable, efficient, affordable, and meets user needs.
 - 03 — Components** — Involves requirements gathering, system design, software construction, testing, maintenance, and project management.
 - 04 — Scope** — Encompasses a wide range of activities beyond coding, including documentation, version control, software architecture, and user interface design.

Software Engineering vs. Software Development

- | | |
|---|--|
| <ol style="list-style-type: none">1. A systematic, disciplined, and quantifiable approach.2. Broad scope, covering the entire software development lifecycle (SDLC).3. Ensures that software is delivered on time, within budget, and with desired quality.4. Uses engineering concepts and practices to solve software-related problems.5. A reliable, efficient, and scalable software system that meets the needs of users and businesses. | <ol style="list-style-type: none">1. The creative process of writing code to create software programs; focuses on the actual coding and building of software.2. Narrower scope, primarily focused on the coding and building phase.3. Focuses on creating functional software according to specifications.4. Involves technical tasks such as writing, debugging, and executing code.5. A working software product or application. |
|---|--|

Software Development LifeCycle

Waterfall Model

Sequential and Linear: The most traditional and straightforward approach

Lean Model

Efficiency and Waste

Reduction: Lean development focuses on creating more value for customers with fewer resources

Agile Methodology

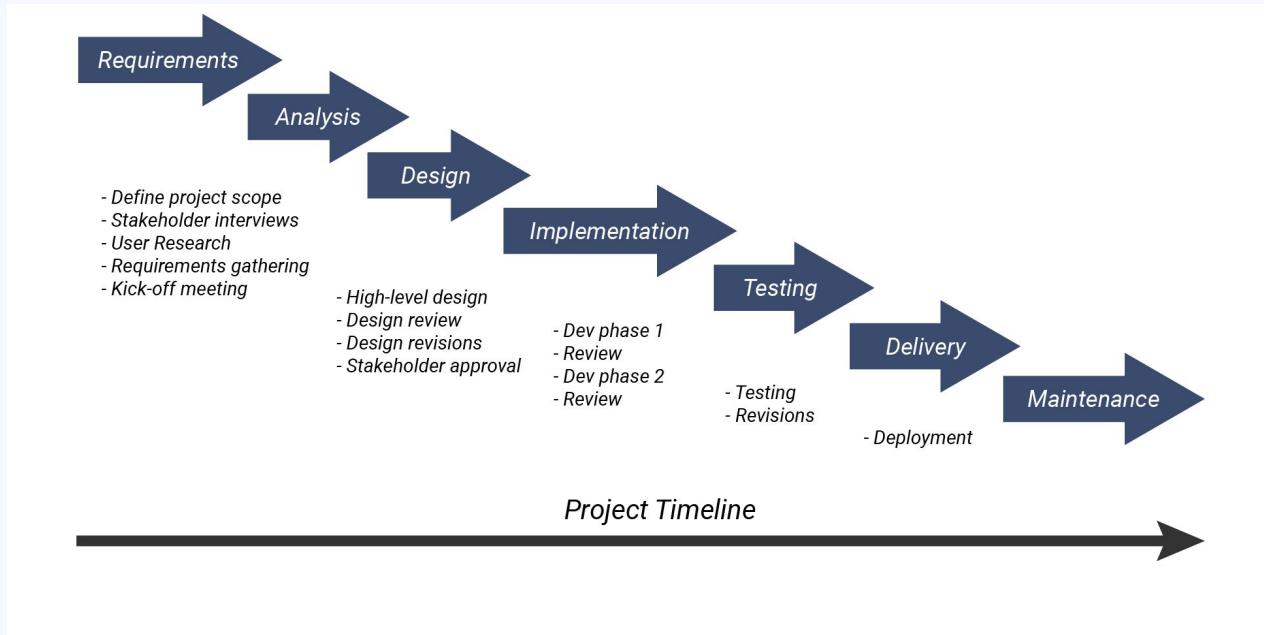
Scrum

Iterative and Incremental: Agile breaks the project into small increments with minimal planning, and without a long-term plan

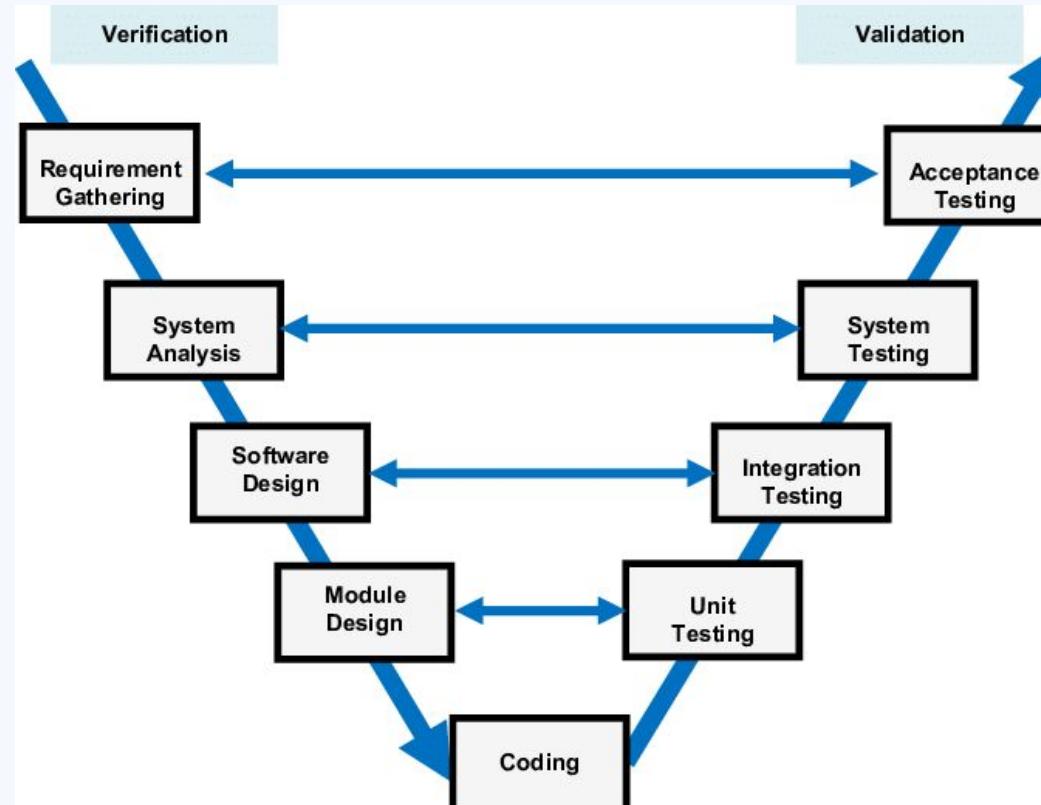
Spiral Model

Risk-driven: combines elements of both iterative and waterfall models but adds a focus on risk assessment

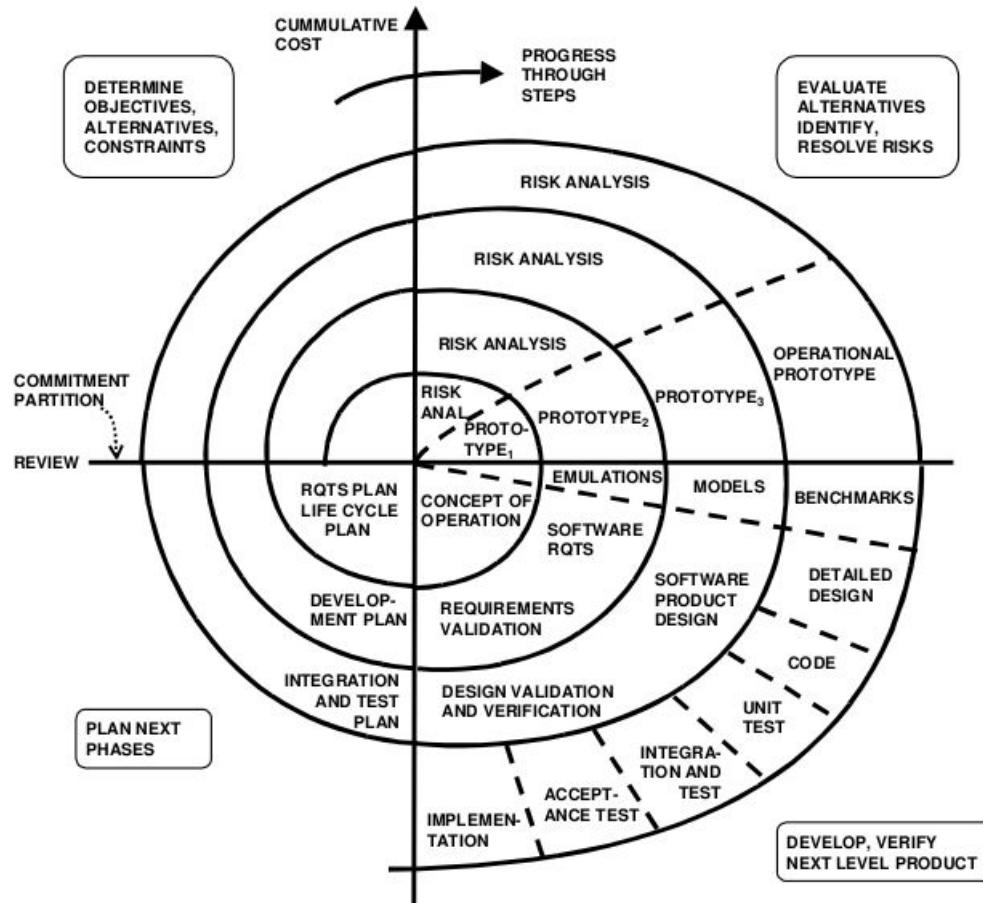
SDLC: Waterfall Model



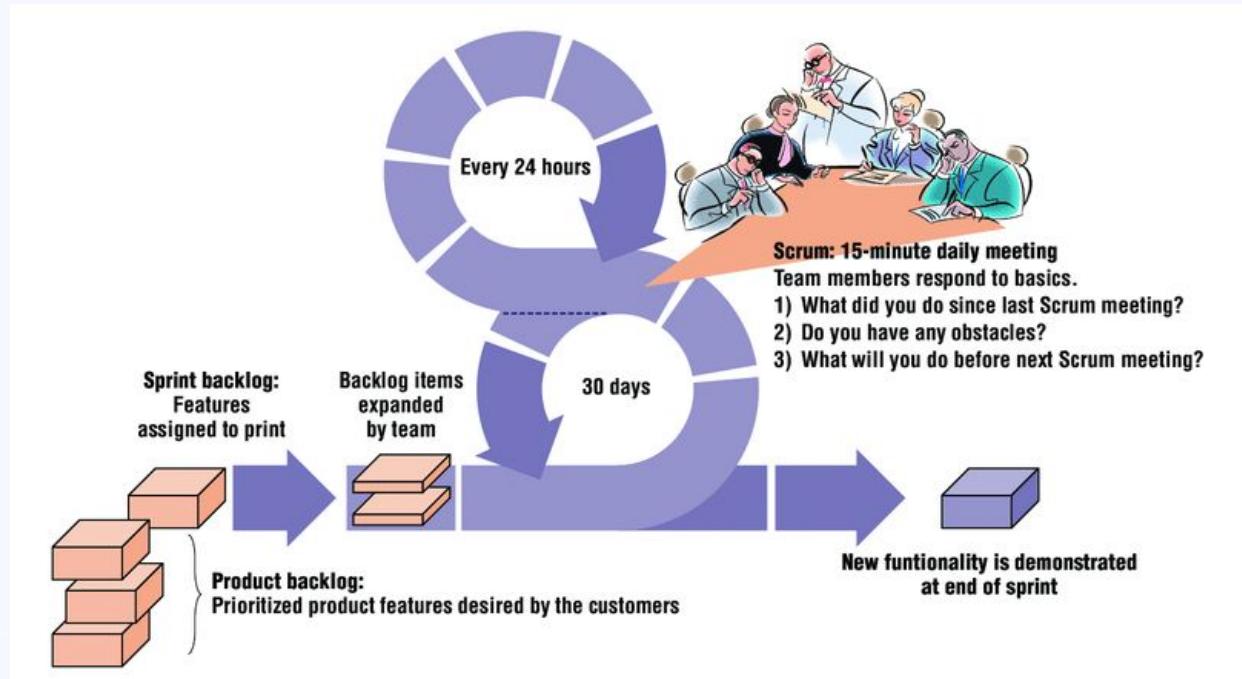
SDLC: V Model



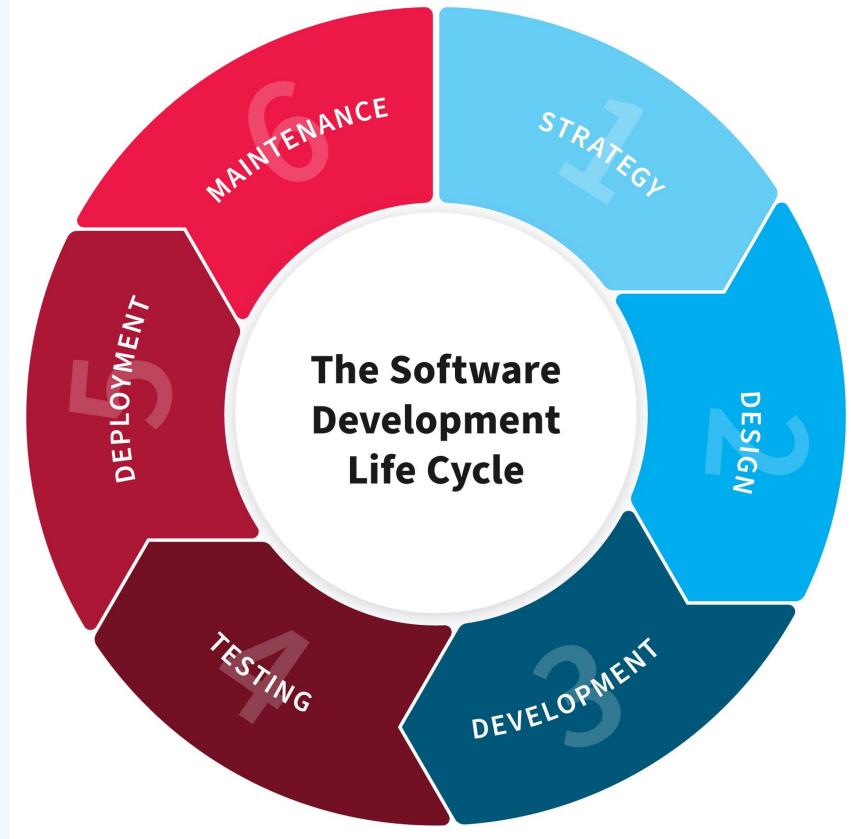
SDLC: Spiral Model



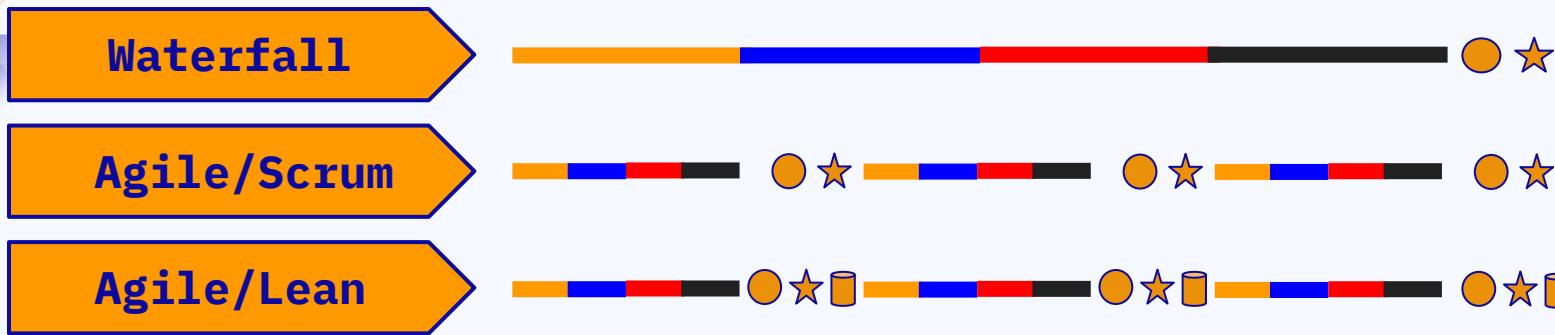
SDLC: Scrum Model



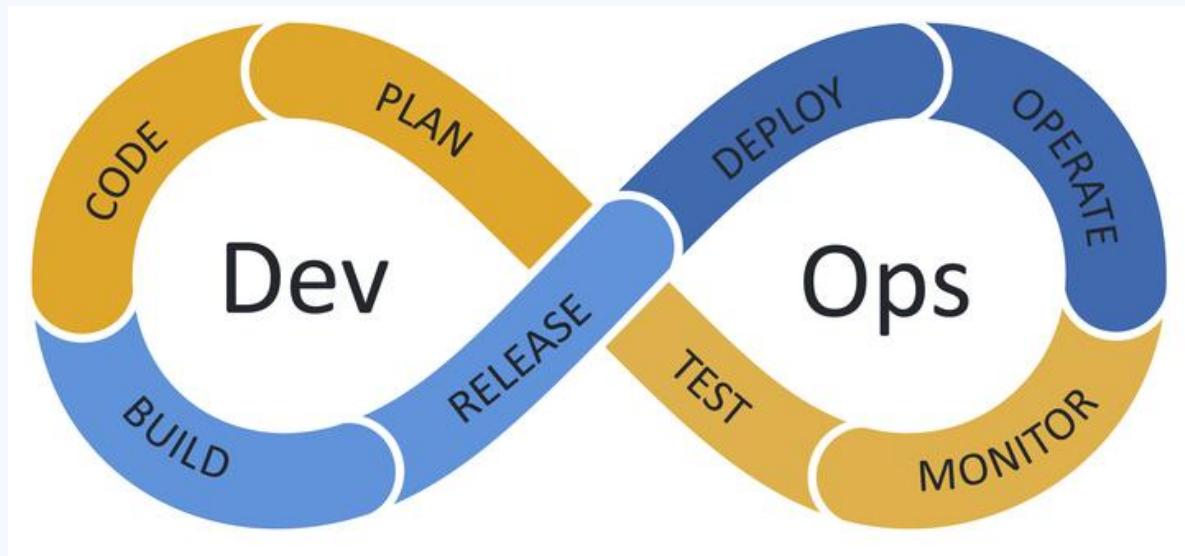
SDLC: Lean Model



SDLC: In Summary



SDLC: DevOps



More than Coding



Architecture



**Project
Management**



**Problem Solving
and Analytical
Thinking**

Architecture

Monolithic	Service-Oriented	Microservices
<ul style="list-style-type: none">• Single-tiered• User interface and data access code are combined	<ul style="list-style-type: none">• Service orientation• Services are provided to the other components by application components	<ul style="list-style-type: none">• Many loosely coupled and independently deployable smaller components or services• Unique process and communicates through well-defined, lightweight mechanisms
Simple applications, small teams, or when rapid development and deployment are required	Enterprise-level services integration	Complex applications requiring high scalability, flexibility

Serverless

- **Description:** An execution model where the cloud provider dynamically manages the allocation and provisioning of servers. A serverless architecture allows developers to build and run applications and services without thinking about servers.
- **Use Cases:** Microservices-based applications, web applications, and for handling variable workloads and auto-scaling needs.



Project Management in Software Engineering



**Scope
Management**



**Time
Management**



**Cost
Management**



**Quality
Management**



**Risk
Management**



**Team management
(if lead)**

Problem Solving and Analytical Thinking



Core to Engineering



Breakdown and Analysis



Creative Solutions



Critical for Optimization

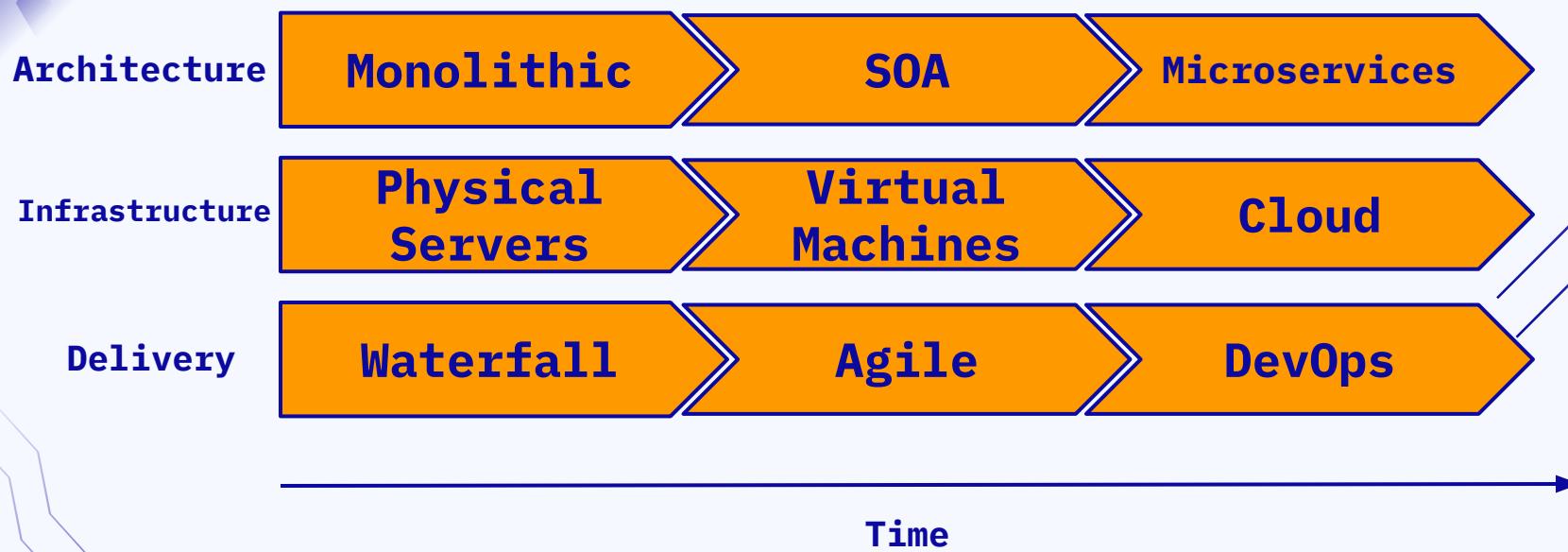


Adaptability



Impact on Quality

Application Modernization



02

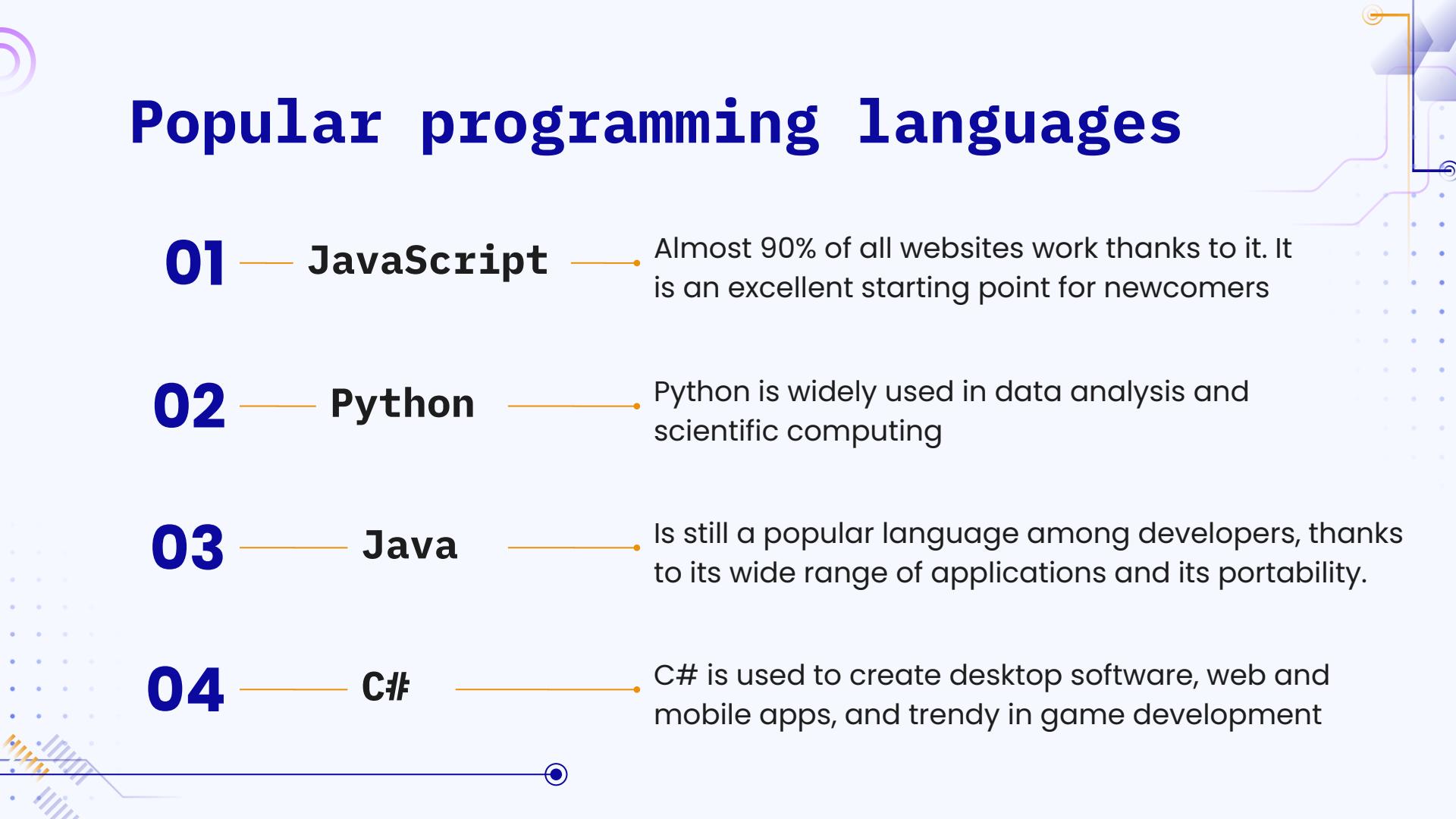
Foundational

Technical Skills

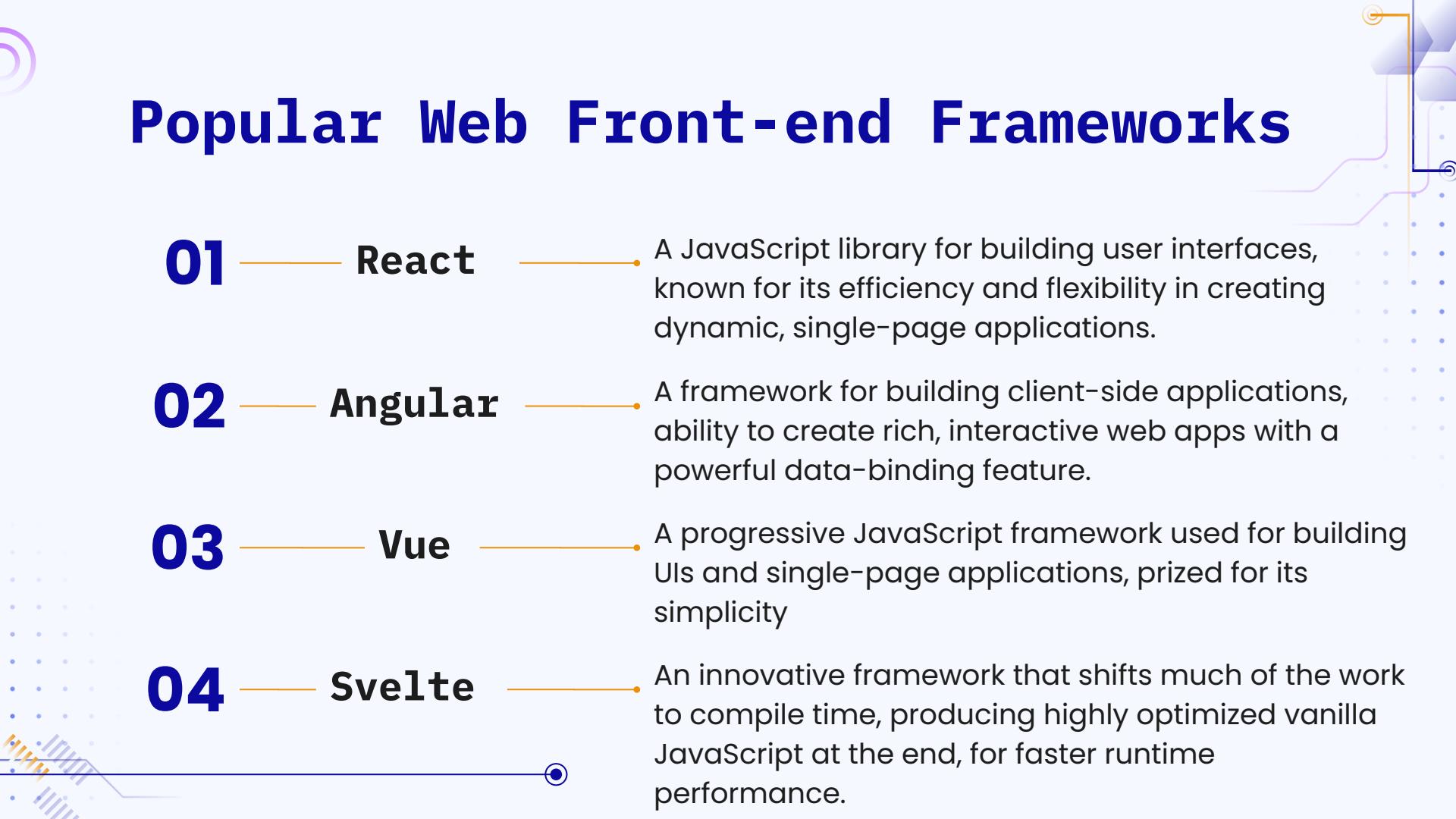
What will you work with



Popular programming languages

- 
- 01 — JavaScript** Almost 90% of all websites work thanks to it. It is an excellent starting point for newcomers
 - 02 — Python** Python is widely used in data analysis and scientific computing
 - 03 — Java** Is still a popular language among developers, thanks to its wide range of applications and its portability.
 - 04 — C#** C# is used to create desktop software, web and mobile apps, and trendy in game development

Popular Web Front-end Frameworks

- 
- 01 — React** A JavaScript library for building user interfaces, known for its efficiency and flexibility in creating dynamic, single-page applications.
 - 02 — Angular** A framework for building client-side applications, ability to create rich, interactive web apps with a powerful data-binding feature.
 - 03 — Vue** A progressive JavaScript framework used for building UIs and single-page applications, prized for its simplicity
 - 04 — Svelte** An innovative framework that shifts much of the work to compile time, producing highly optimized vanilla JavaScript at the end, for faster runtime performance.

Popular Mobile Front-end Frameworks

- 01 — Flutter** — Google's open-source framework for developing native Android and iOS apps from a single codebase
- 02 — React Native** — An open-source, cross-platform framework from Facebook, allowing for Android and iOS app development with JavaScript
- 03 — Ionic** — A framework for building native and web-optimized apps using Angular and Apache Cordova
- 04 — Xamarin** — A Microsoft-owned framework that uses .NET and C# for creating native apps for Android, iOS, and other platforms

Popular Back-end Technologies

01

Node.js

Perfect for web applications requiring high performance and scalability

02

Django/Flask

Python continues to be a favorite due to its simplicity and readability, with Django and Flask as its standout frameworks.

03

Golang

Designed with performance in mind, Golang is ideal for data-intensive and complex applications

04

ASP .NET

Powered by C#, is a robust framework from Microsoft, offering extensive support and a strong community.

Popular DBMS Tools

01 — MySQL

02 — PostgreSQL

03 — SQLite

04 — MongoDB

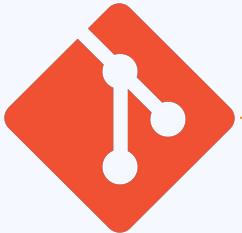
Version Control Systems

Collaboration

**Track Changes
and History**

Popular VCS Tools

**Benefits of
Using VCS**



Testing

Unit Testing Tools

JUnit (for Java) and
PyTest (for Python)

Performance Testing Tools

Apache JMeter and
LoadRunner

Integration Testing

Postman for API testing

Automated Testing Frameworks

Selenium



Security

01

SAST Tools

SonarQube

02

DAST Tools

OWASP ZAP

03

Dependency Scanners

WhiteSource

04

WAF

Cloudflare

Project Management Tools



Project

Tools like Jira, ZenHub or any project management tool



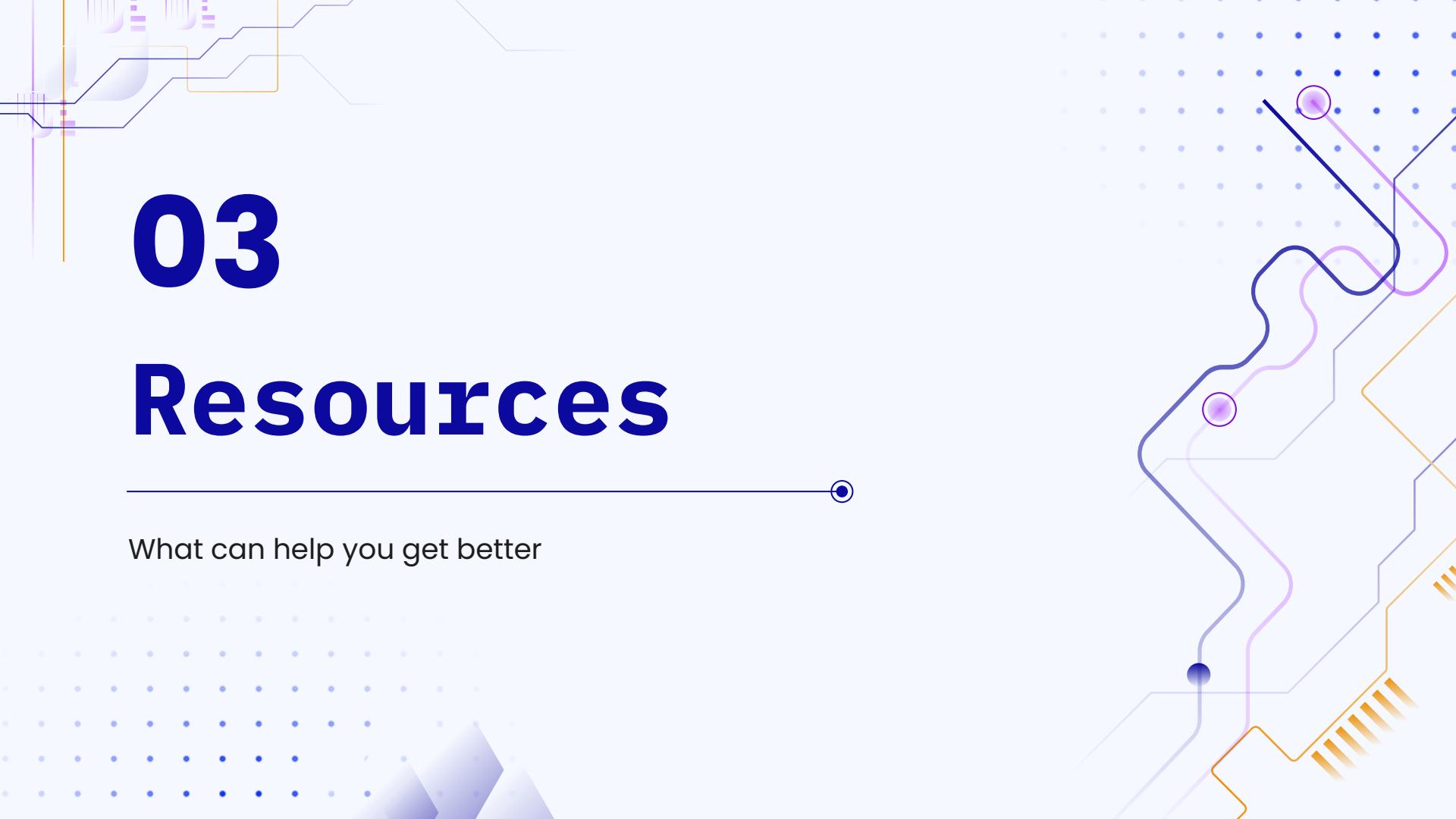
Communication

Apps like Slack or Teams can be used to organize communication and work among teams

03

Resources

What can help you get better



Official Websites/ Documentation

Knowledge
resource

Best
practices

Up-to-Date
Information

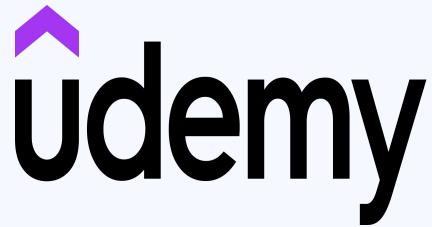
Troubleshooting

Foundational
for
Self-Learning

Online Learning Platforms



Coursera



Udemy



Simplilearn

Job Preparation Platforms



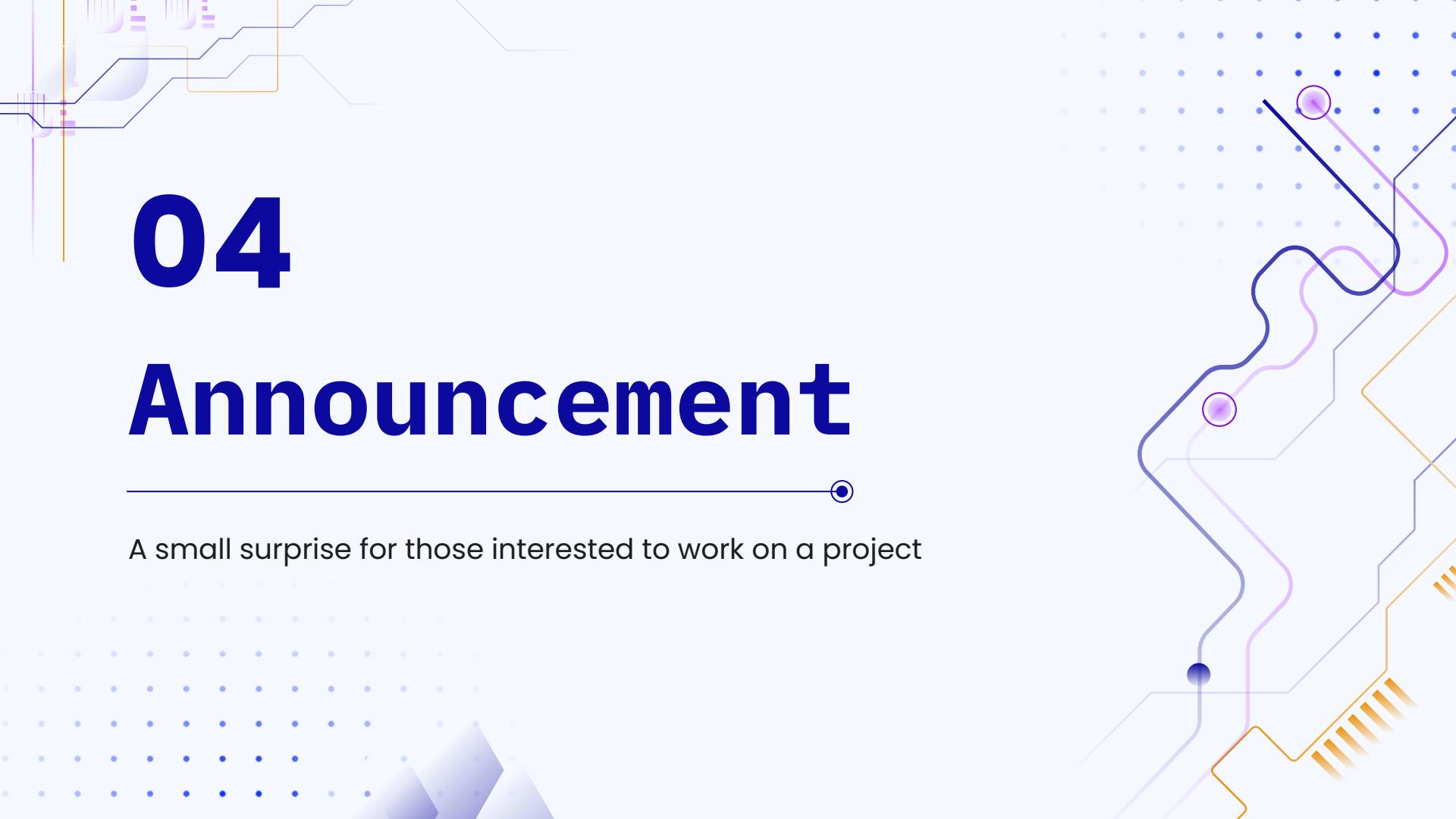
LeetCode



InterviewBit



Simplilearn



04

Announcement

A small surprise for those interested to work on a project

Scan This QR CODE



Thanks!

Email: aah126@mail.aub.edu

Github: [Ayla-Hmadi](https://github.com/Ayla-Hmadi)

Let's go eat now!!



Linked in

Thanks !

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)

Please keep this slide for attribution