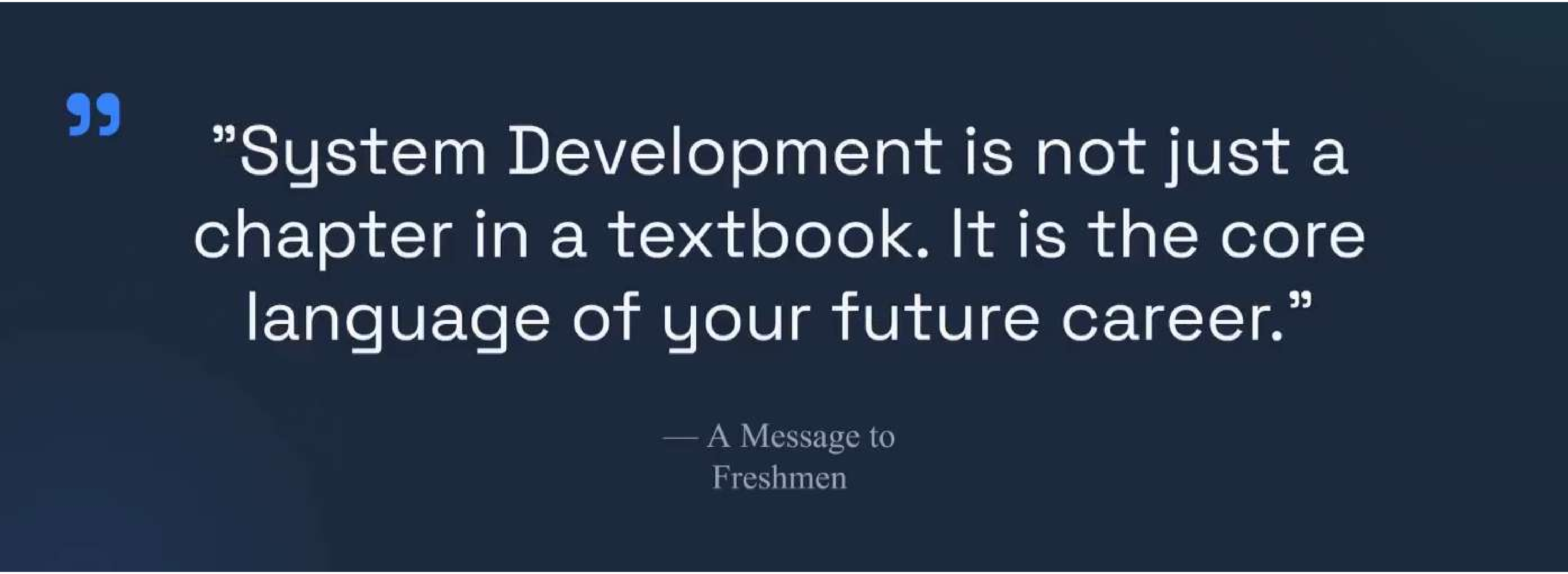
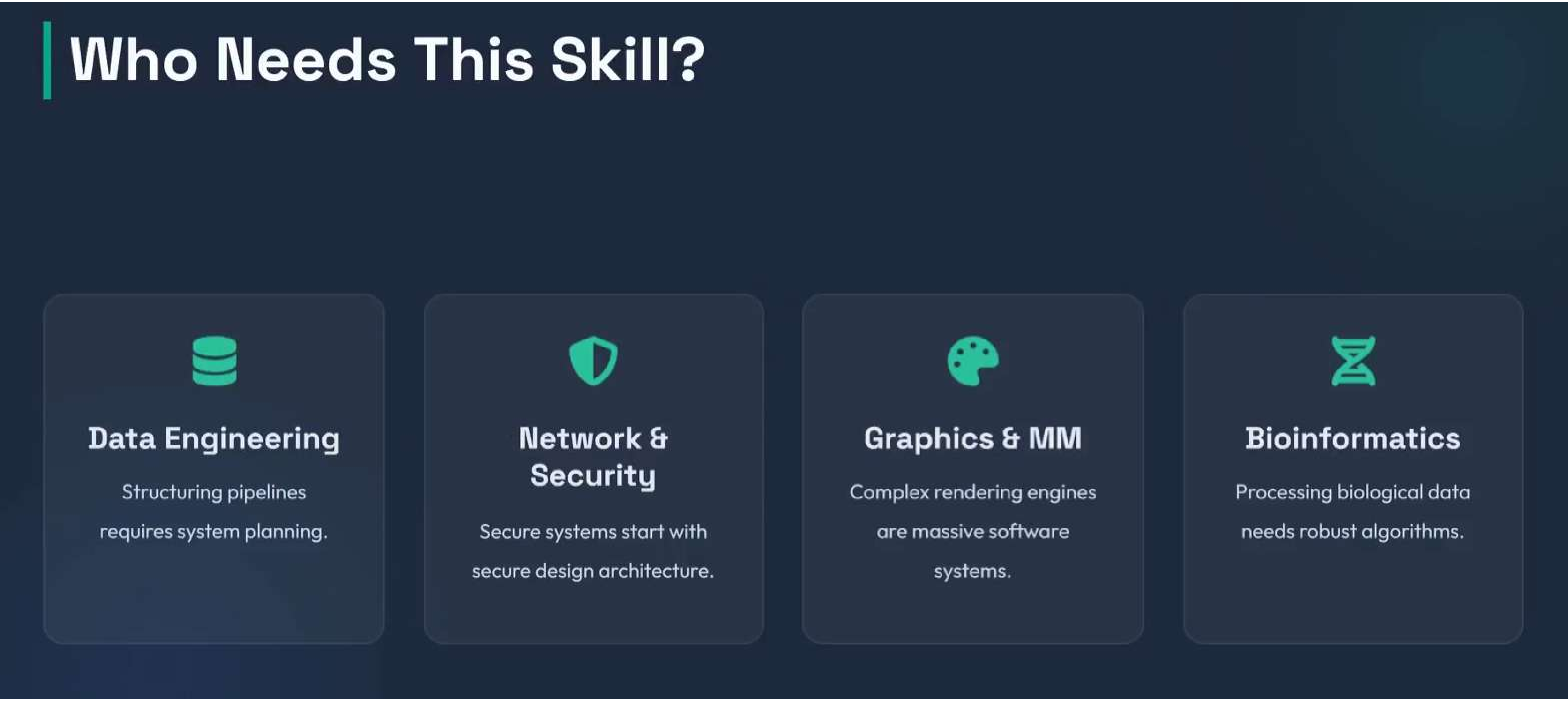
Project Management & System Development

Building the CORE Foundation for Your Computer Science Journey



> ”System Development is not just a chapter in a textbook. It is the core language of your future career.”

— A Message to Freshmen



Who Needs This Skill?

Data Engineering

Structuring pipelines requires system planning.

Network & Security

Secure systems start with secure design architecture.

Graphics & MM

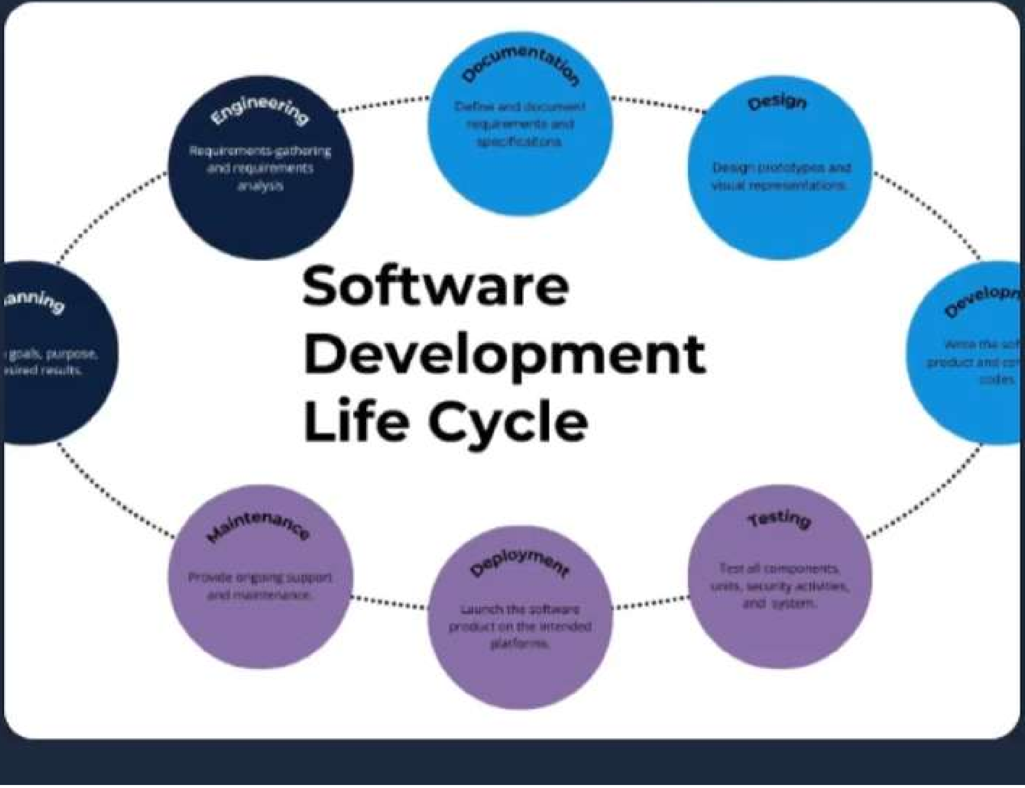
Complex rendering engines are massive software systems.

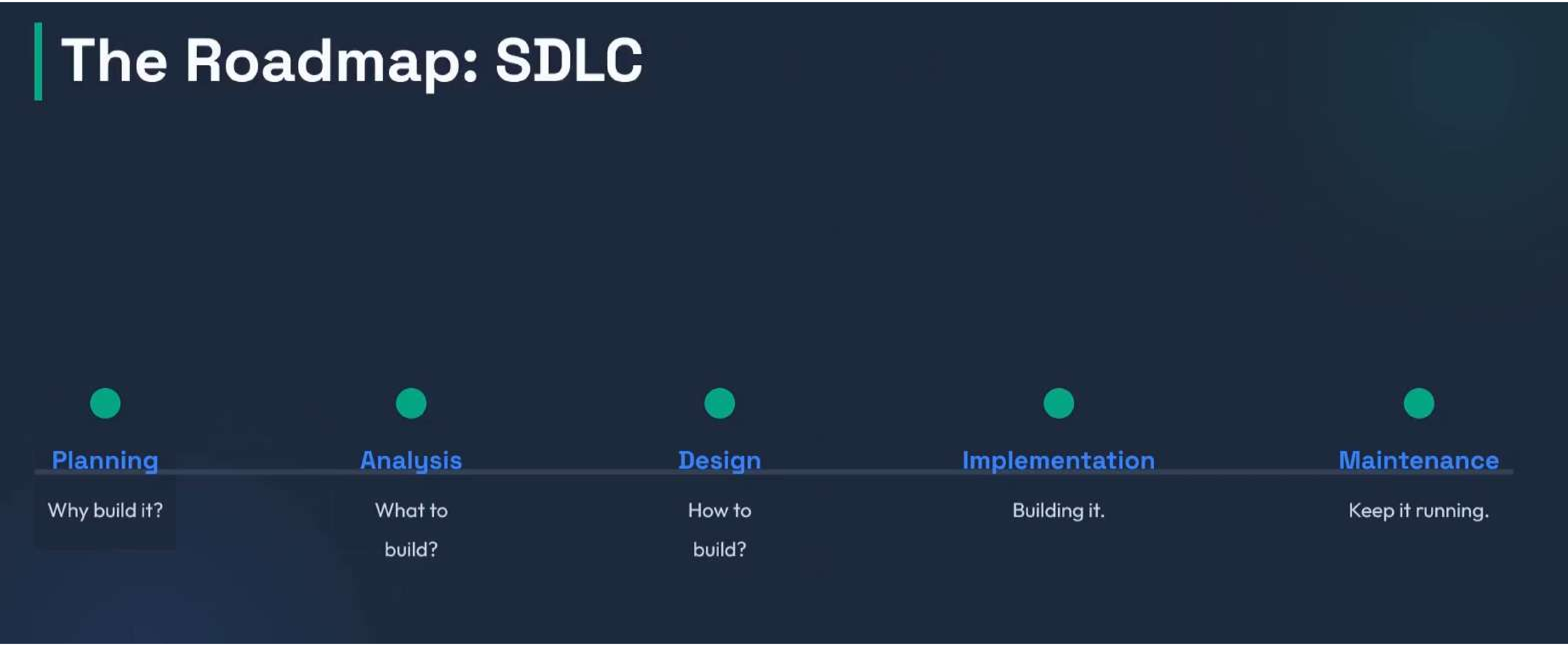
Bioinformatics

Processing biological data needs robust algorithms.

What is System Development?

- It is **NOT** just writing code.
- It is the **entire process** of defining, designing, testing, and implementing a software application.
- Think of it like building a house: You don't just start laying bricks. You need a blueprint first.
- Without it, you have chaos, bugs, and failed projects.


Software Development Life Cycle
Engineering
Requirements-gathering and requirements analysis
Documentation
Define and document requirements and specifications
Design
Design prototypes and visual representations.
Develop
Write the sof product and co codes.
Testing
Test all components, units, security activities, and system.
Deployment
Launch the software product on the intended platforms.
Maintenance
Provide ongoing support and maintenance.
anning
goals, purpose, sired results.



The Roadmap: SDLC

Planning

Why build it?

Analysis

What to build?

Design

How to build?

Implementation

Building it.

Maintenance

Keep it running.

Why Project Management?

Controlling Chaos

Software projects are complex. Project Management (PM) ensures you stay on track, on time, and within budget.

Team Synergy

In CS, you rarely work alone. PM skills help you coordinate with designers, testers, and other developers effectively.

PM Methodologies: The ”How”

1. Waterfall (Traditional)

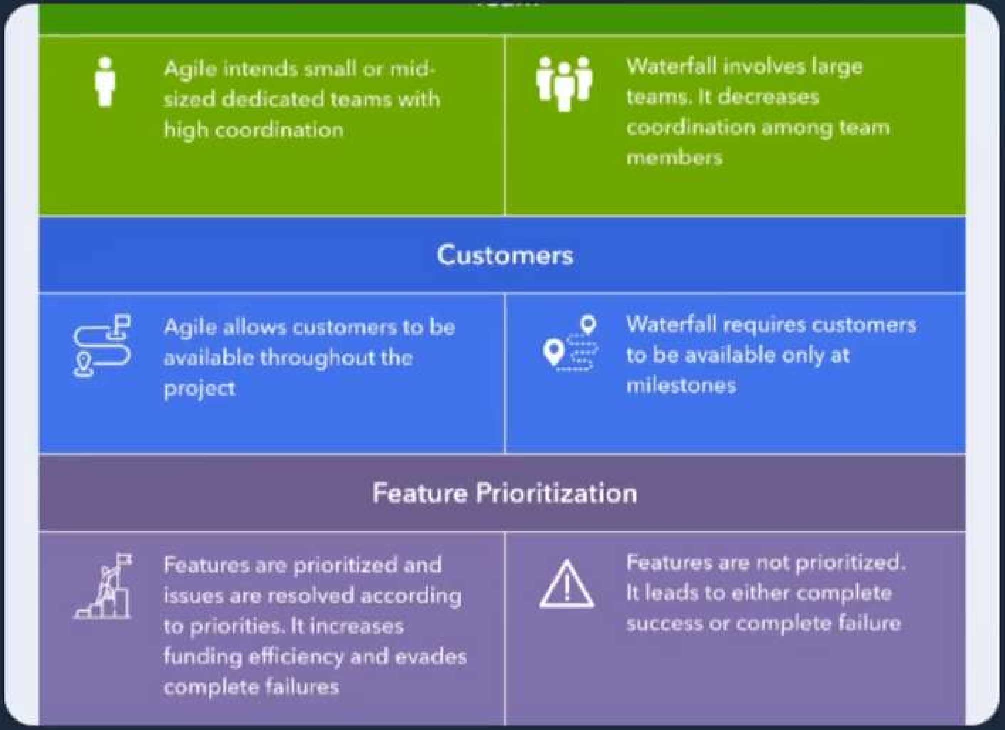
A linear, sequential approach. You finish one phase before starting the next.

- **Best for:** Projects with fixed requirements.
- **Pros:** Clear milestones, easy to manage.
- **Cons:** Hard to change course once started.

2. Agile (Modern)

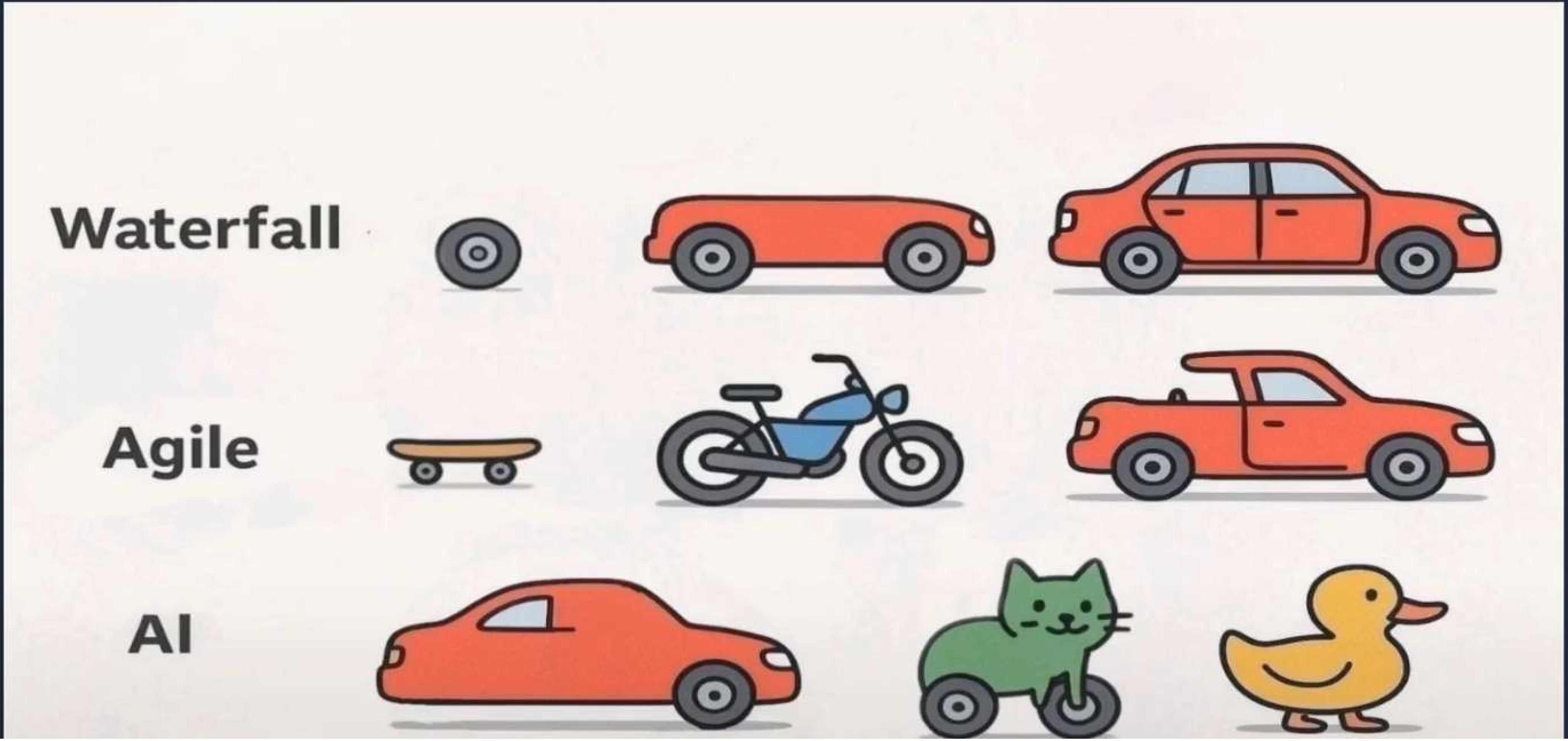
Iterative approach. Work is broken into small "sprints".

- **Best for:** Rapidly changing software needs.
- **Pros:** Flexible, faster feedback.
- **Examples:** Scrum, Kanban.



Agile intends small or mid-sized dedicated teams with high coordination	Waterfall involves large teams. It decreases coordination among team members
Customers	
Agile allows customers to be available throughout the project	Waterfall requires customers to be available only at milestones
Feature Prioritization	
Features are prioritized and issues are resolved according to priorities. It increases funding efficiency and evades complete failures	Features are not prioritized. It leads to either complete success or complete failure

Waterfall vs Agile

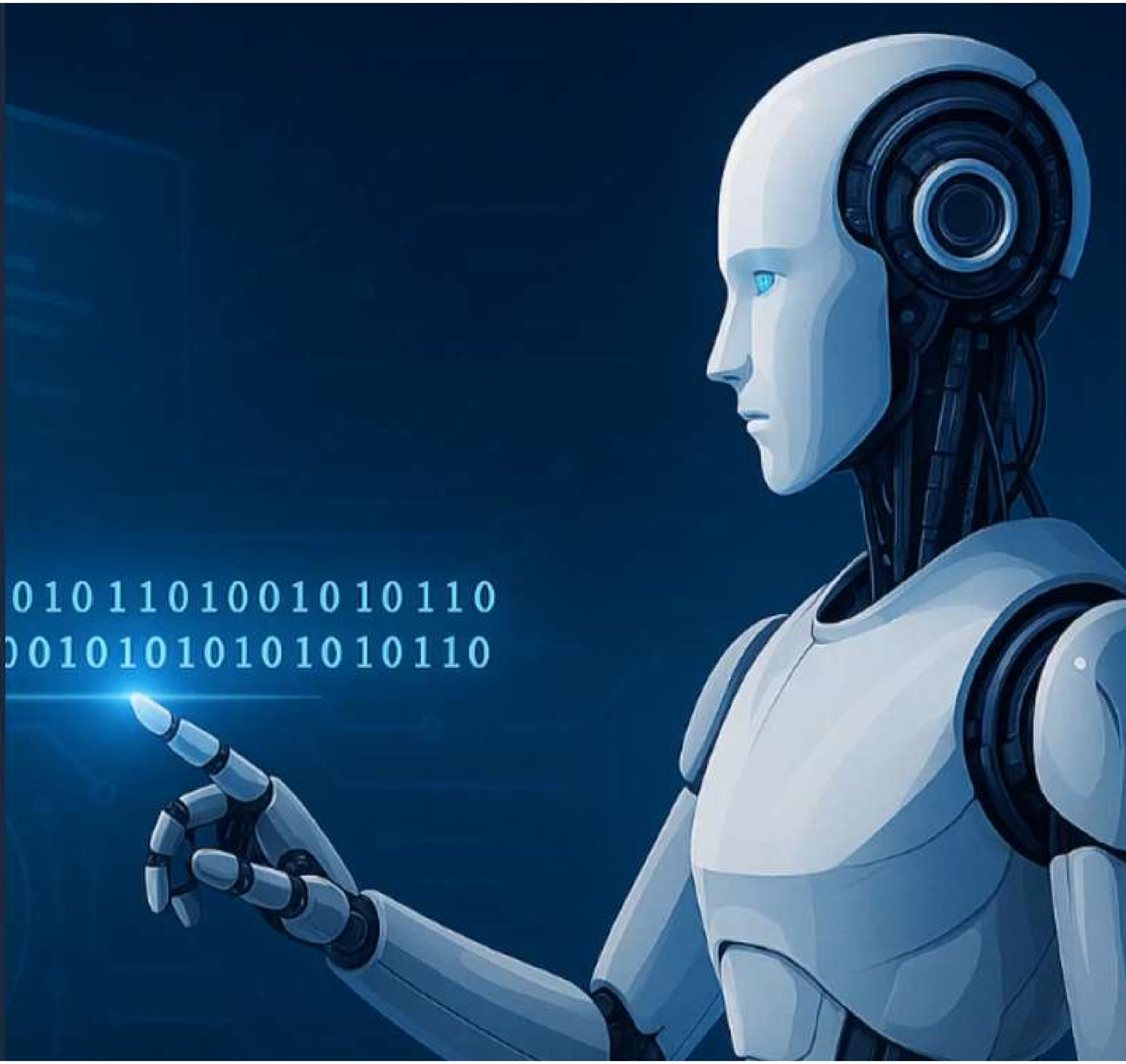


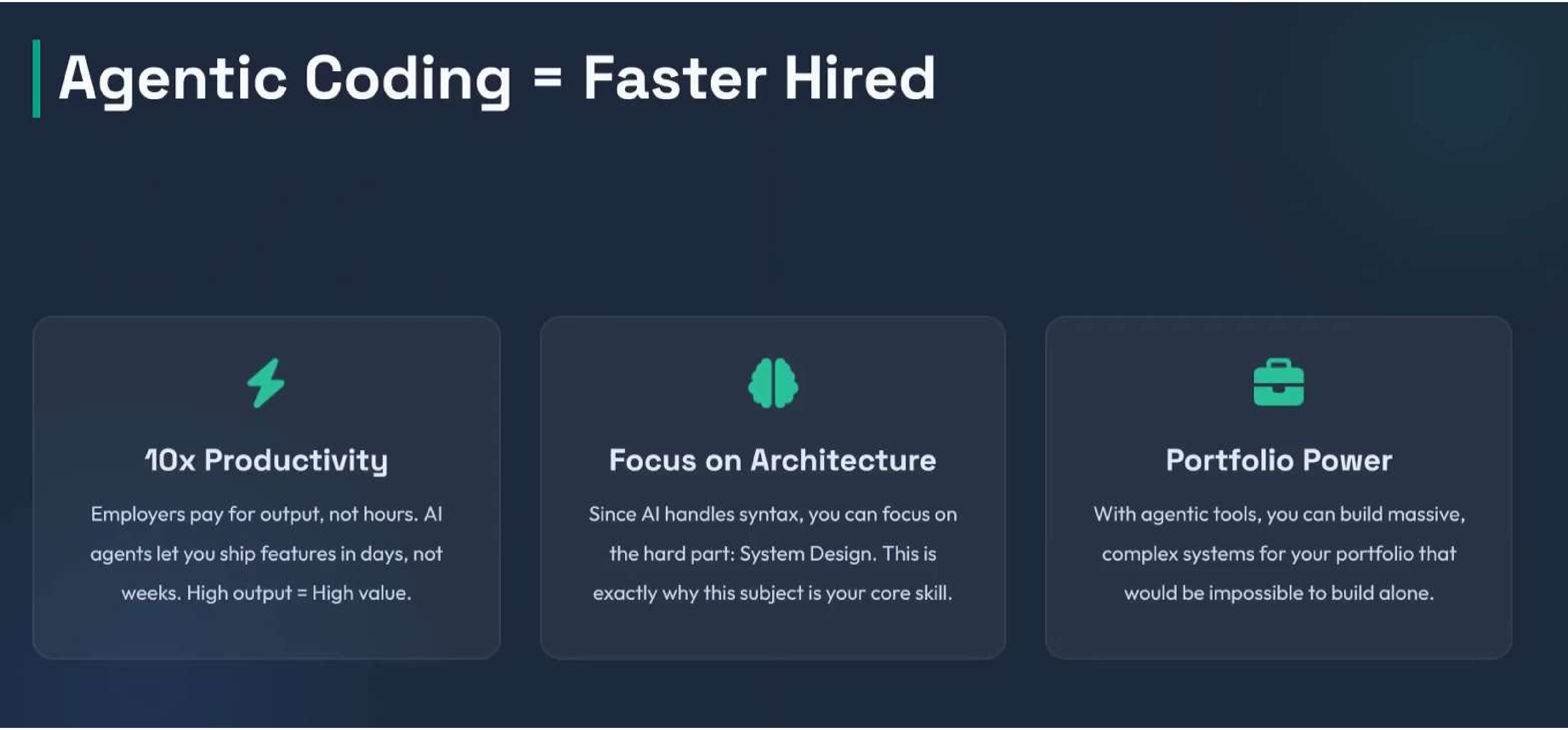
The New Era:
Agentic Coding

It is no longer just about "writing code." We are entering the age of **AI Agents**.

Tools like Devin or advanced Copilots act as autonomous developers. They can plan tasks, write code, debug errors, and deploy fixes—all on their own.

Your Shift: You stop being just a "Coder" and start being a "System Architect." You manage the AI that builds the system.





Agentic Coding = Faster Hired

10x Productivity

Employers pay for output, not hours. AI agents let you ship features in days, not weeks. High output = High value.

Focus on Architecture

Since AI handles syntax, you can focus on the hard part: System Design. This is exactly why this subject is your core skill.

Portfolio Power

With agentic tools, you can build massive, complex systems for your portfolio that would be impossible to build alone.

The "FYP" Trap

Don't ignore this until Year 4.

Many students treat this as "just a chapter." When Final Year Project comes, they panic.

Your FYP requires you to build a complete system. If you forget these skills, you will struggle to graduate.

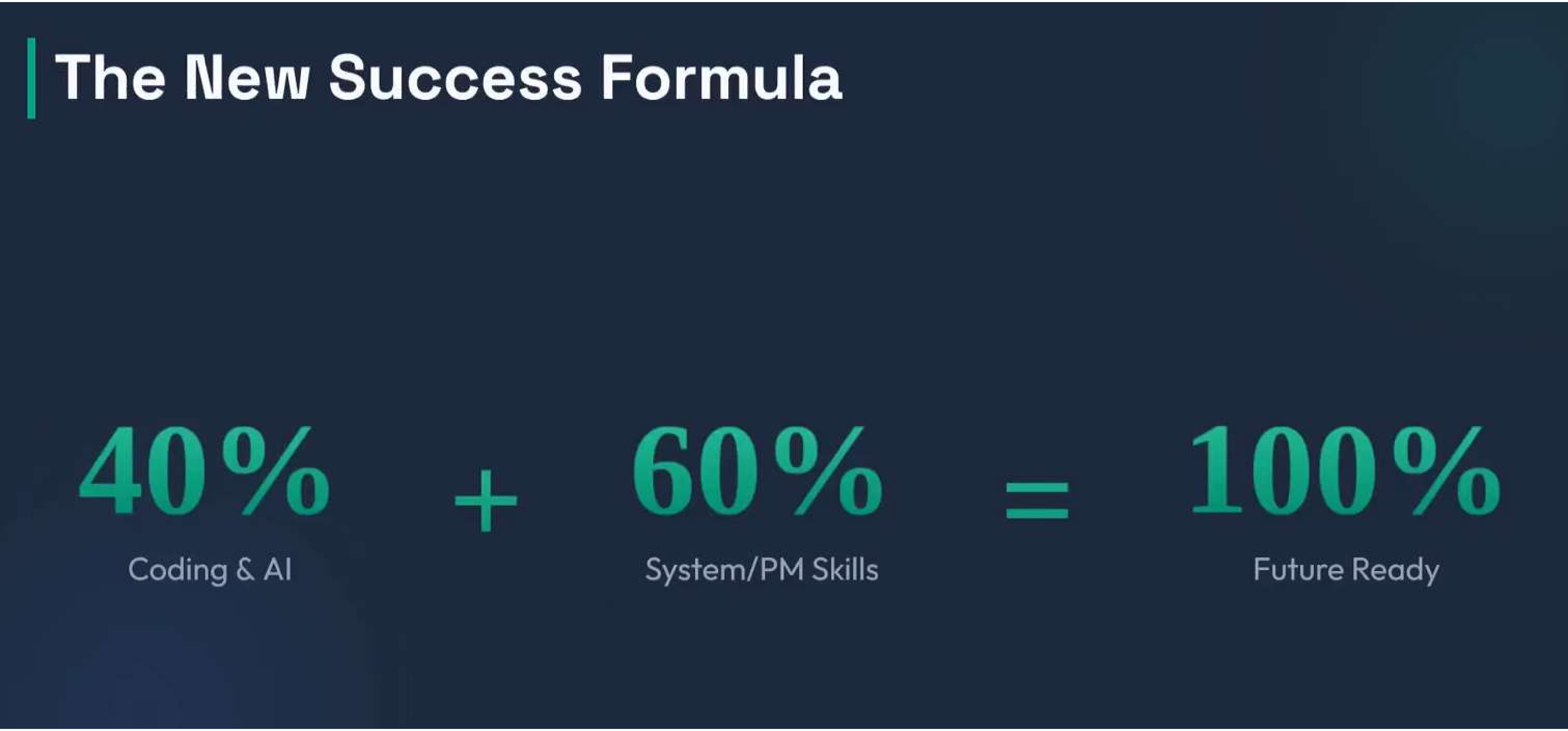
Start mastering this mindset today.

Core Skills to Master

Requirement Gathering

System Design

AI-Assisted Dev



The New Success Formula

40% + 60% = 100%

Coding & AI

System/PM Skills

Future Ready