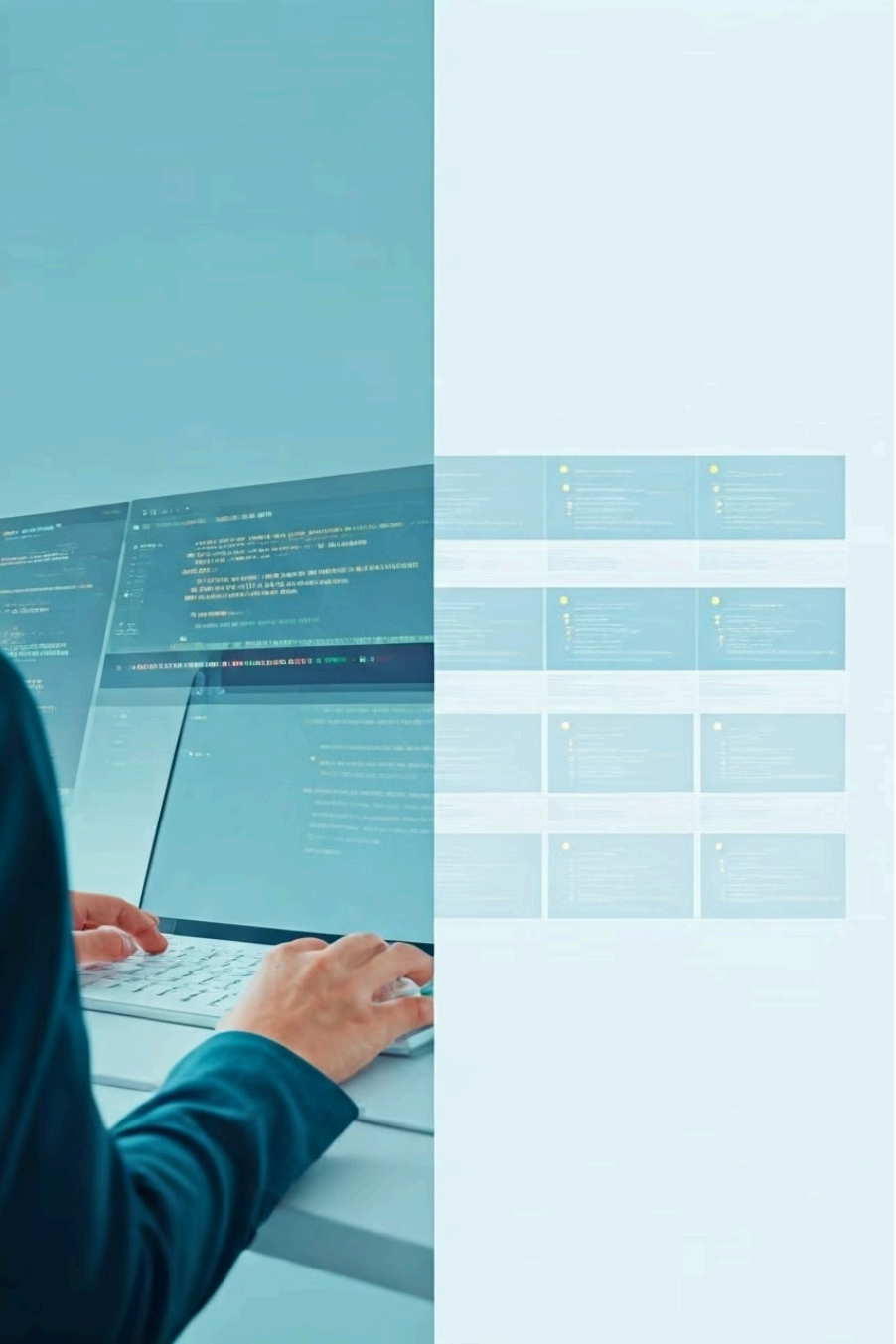


Programming vs. Software Engineering: Key Differences

Exploring time, scale, and trade-offs in software development

by Tanvi Shinde

Time: The First Key Difference

Programming

Focuses on generating new software

Software Engineering

Involves development, modification, maintenance over time

Time: Code Lifespan Impact

1. **Short-lived Code**

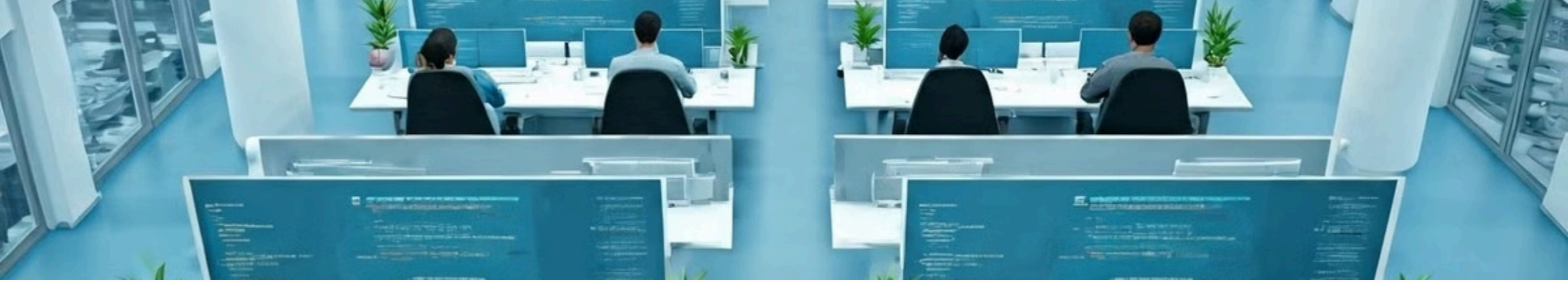
   Minutes to hours, less affected by time

2. **Long-lived Code**

   Years to decades, must adapt to changes

3. **Adaptation Needed**

   Dependencies, libraries, technologies evolve



Scale: The Second Key Difference

1

Programming

Often an individual task

2

Software Engineering

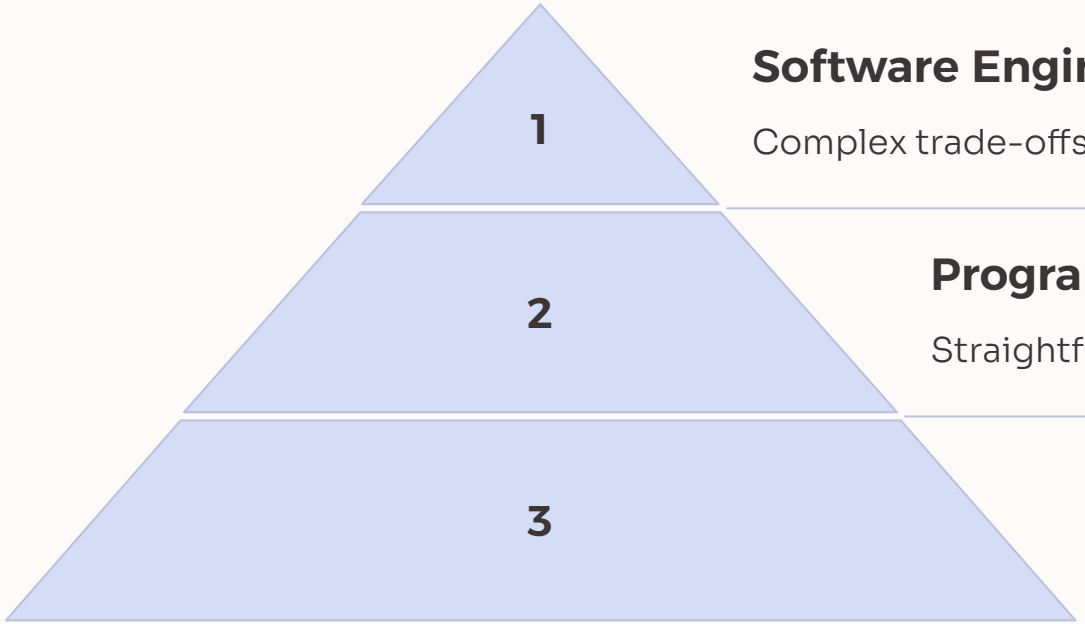
Team effort, collaboration, coordination

3

Large-scale Projects

Require organization, workflows, policies

Trade-offs: The Third Key Difference



Software Engineering

Complex trade-offs, high stakes

Programming

Straightforward decisions

Balancing Act

Technical debt, sustainability, scaling costs

Google's Software Engineering Insights

Integrating Programming Over Time

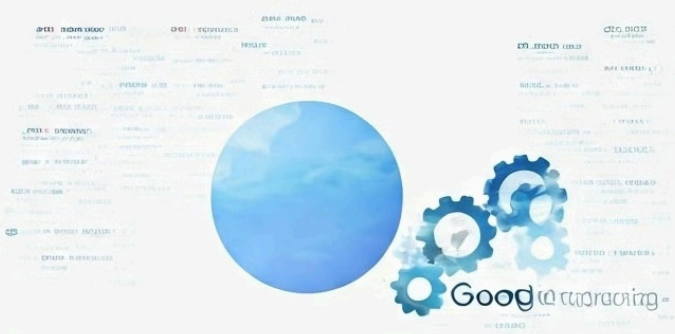
Consider project lifespan and evolution

Team Dynamics

Collaboration crucial for large-scale success

Strategic Decision-Making

Long-term viability through informed choices



Applying Large-Scale Practices to Smaller Projects

Sustainability

Design for long-term maintainability

Adaptability

Prepare for future changes and updates

Scalability

Build with potential growth in mind

Key Takeaways

Time Matters

Consider code lifespan and future changes

Scale Impacts Process

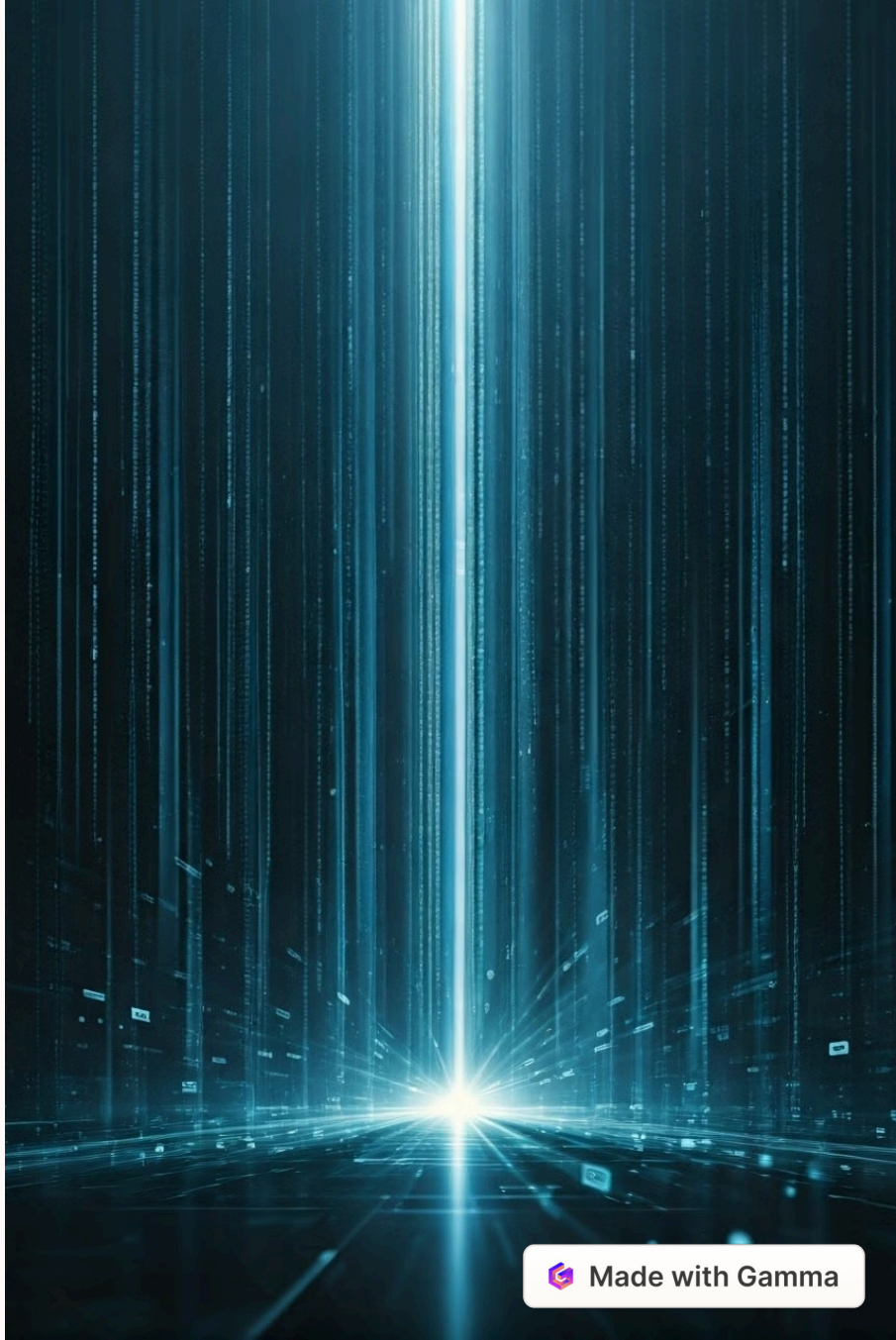
Adapt practices to project size and team

Balance Trade-offs

Make informed decisions for long-term success

Apply Best Practices

Use large-scale insights in smaller projects



Connect & Stay Tuned

Email

Reach out to me at: tanvishinde.arch@gmail.com

LinkedIn

Connect with me on LinkedIn: **linkedin.com/in/tanvi-shinde-72a40320**

GitHub

Check out the series on GitHub:
github.com/itstannus/software-engineer