IT's Tinkoff Java Meetup · April 4, 2023

Where should you grow after reaching Senior Software Engineer?

Career tracks, Staff+ levels, growth process, and self-development



Alexander Polomodov

Context Speaker

Technical Director & Fellow, T-Technologies

- Led Tinkoff digital ecosystems engineering
- Owned architecture topics across the company
- Owned our System Design Interview
- Interviewed technical managers and SREs

Five questions after reaching Senior

Conversation map 5 parts

From the career fork to concrete development practice

- SDE career paths after Senior
- Staff+ levels and IC archetypes
- Tinkoff engineering growth process
- A general self-development approach
- Growing an engineer toward Staff+

Part 1

01

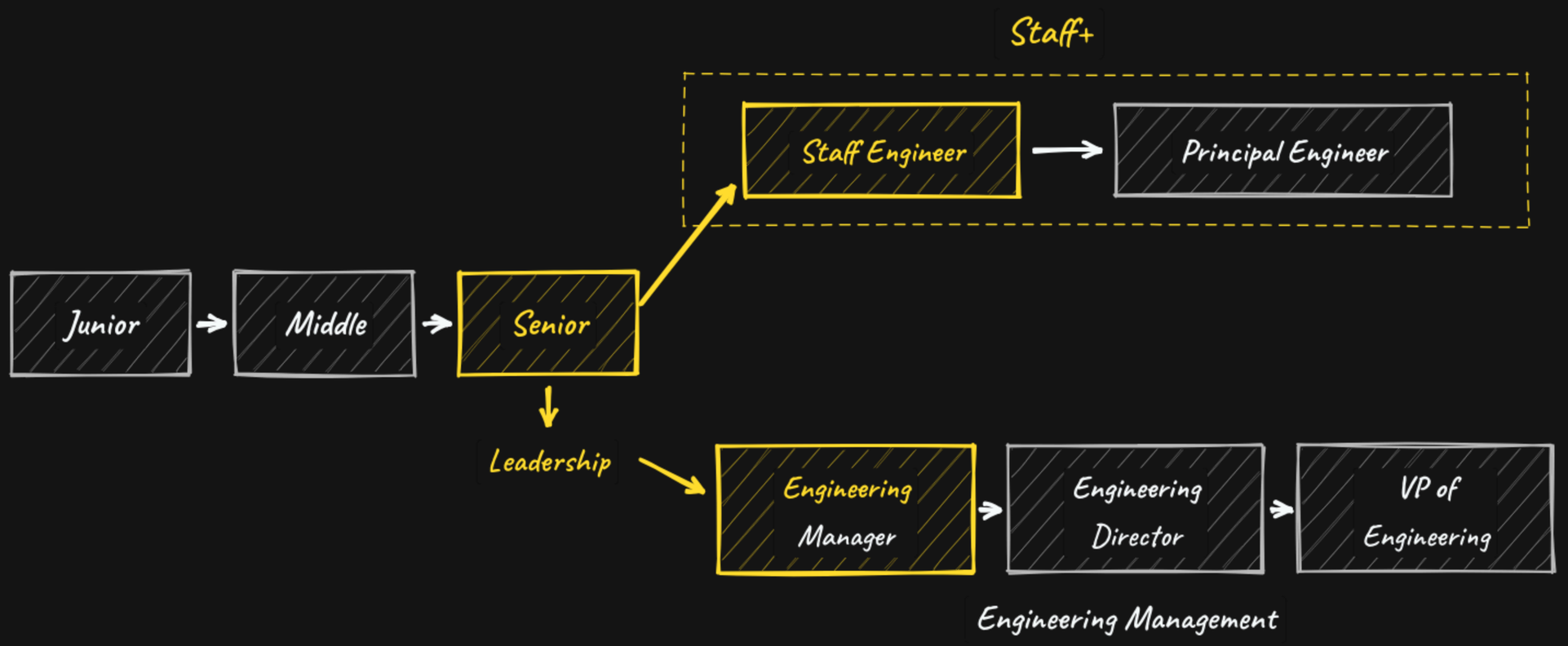
After Senior, the ladder becomes a fork

Staff+ and Engineering Management scale impact differently

Two tracks begin after Senior

Career path IC vs EM

Leadership is a transition, not a separate grade



Staff+ matters under two conditions

Precondition Staff+

Wanting a longer career ladder is not enough

- The company already has such engineers
- The company actually needs such engineers

Part 2

02

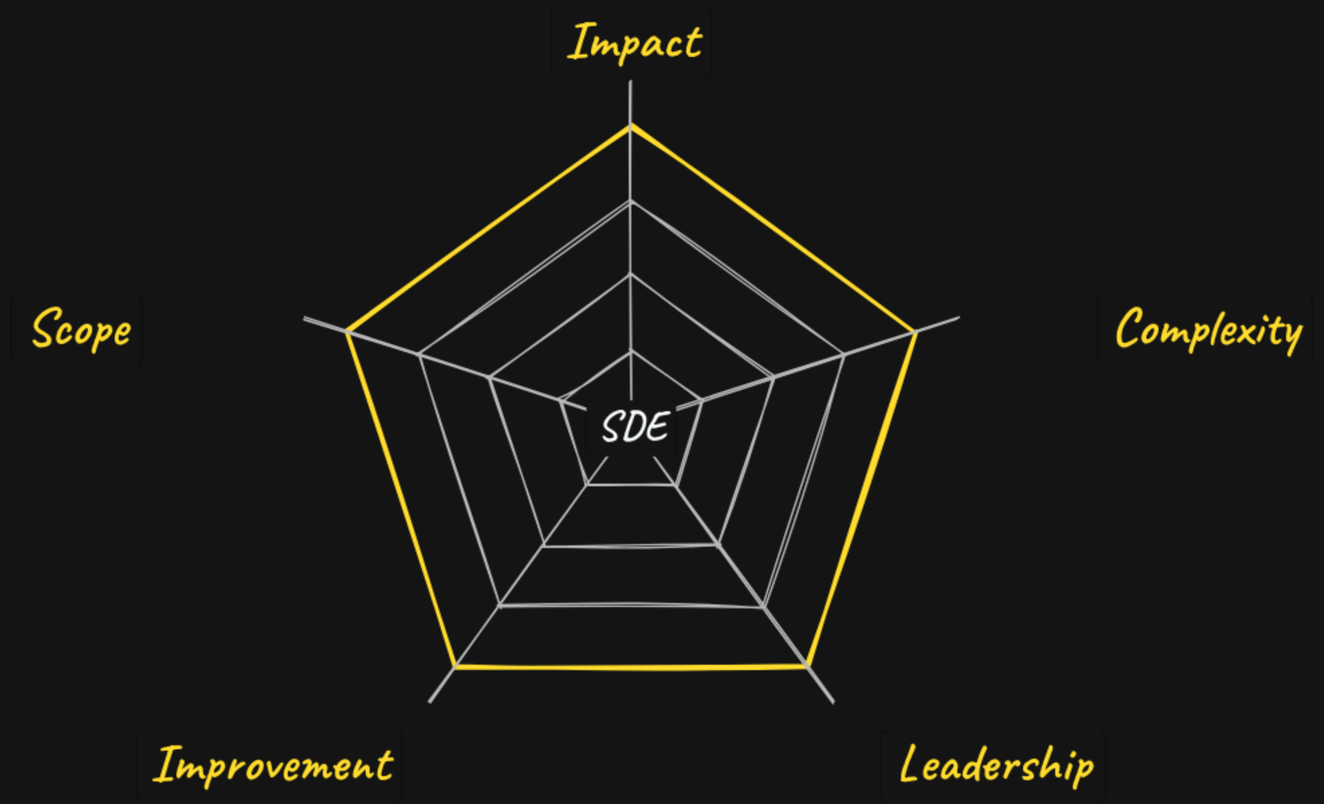
Staff+ is defined by sustained impact

The matrix, levels, and archetypes describe different dimensions of work

Growth appears across five axes

SDE matrix 5 axes

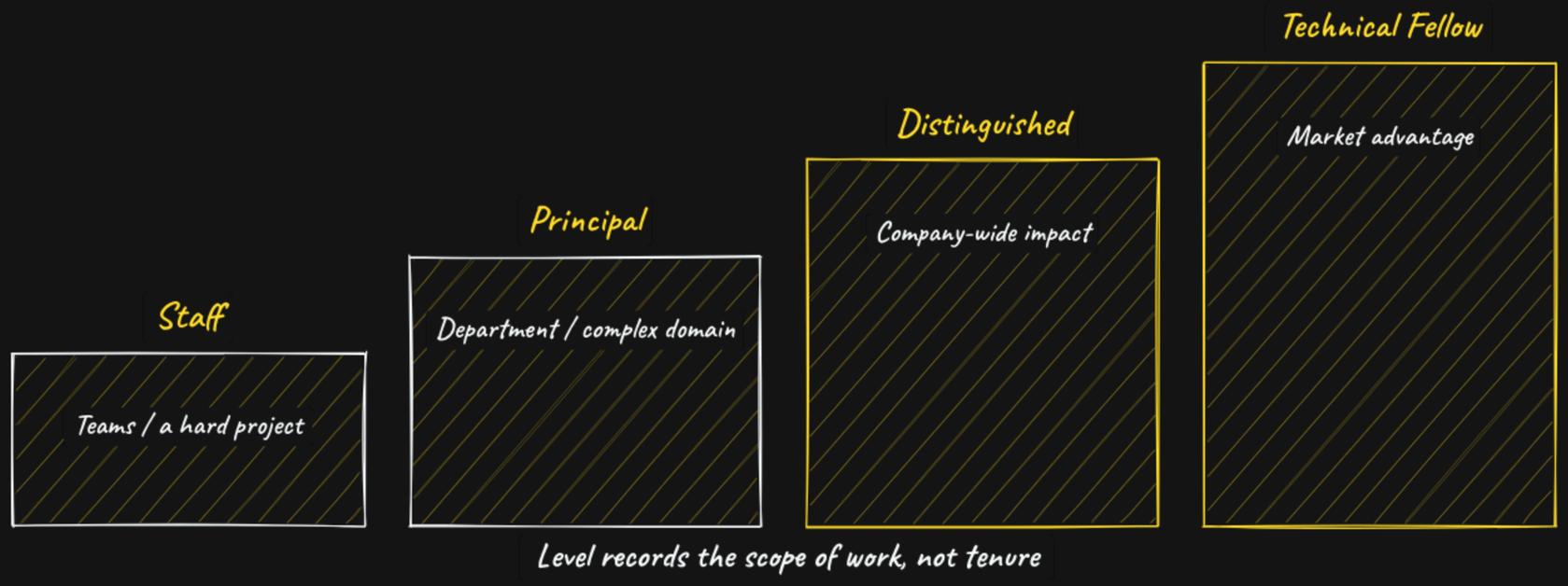
Scope, Impact, Complexity, Leadership, and Improvement



Level grows with the scale of impact

From a hard project to a unique company advantage

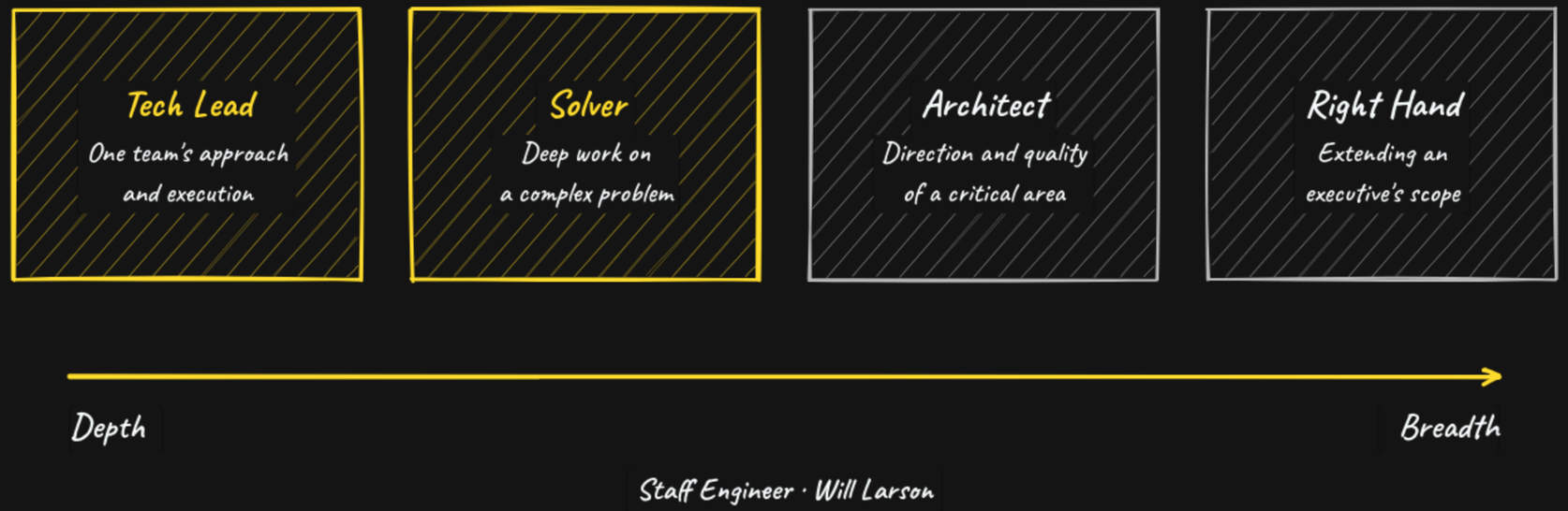
Staff+ 4 levels



Staff+ spans different kinds of work

Tech Lead, Solver, Architect, and Right Hand

Staff Engineer · Depth → Breadth



No track yet Algorithm

A Staff+ role can emerge bottom-up

Start with a real organizational need

- Hard problem exists — take it as Senior
- No request — formalize problem and solution
- Nothing found or bought — change companies

Part 3

03

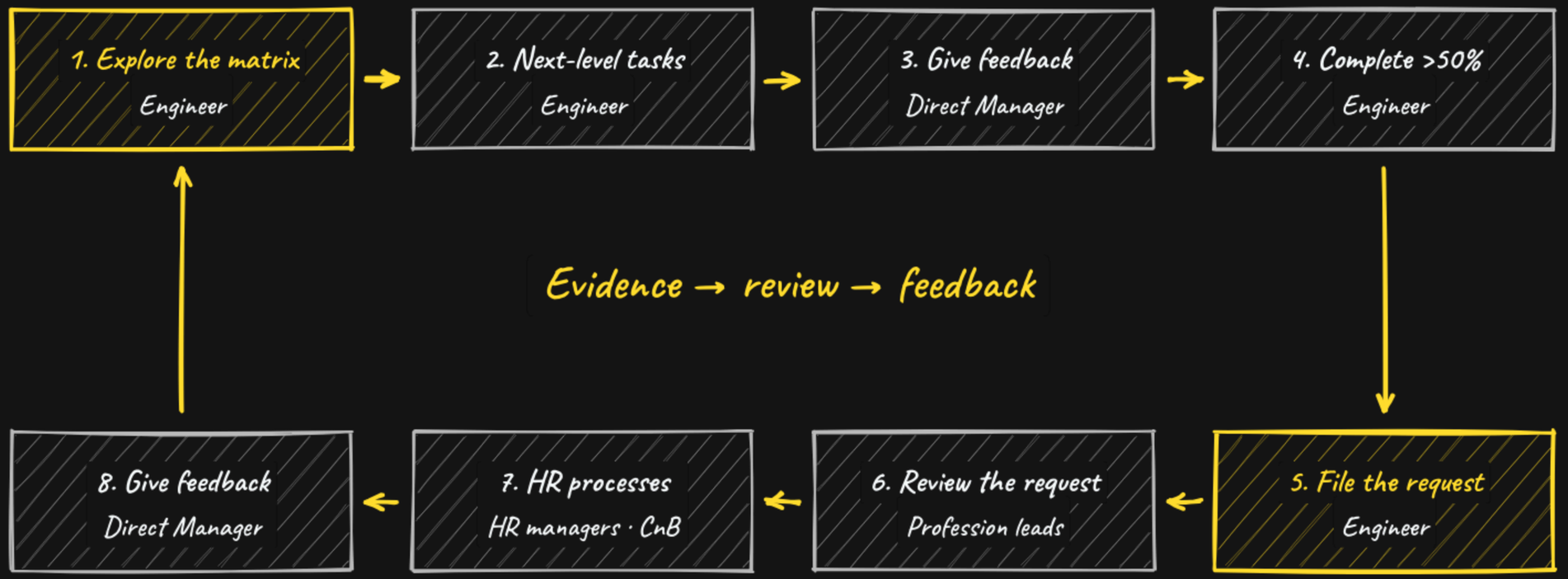
Growth is a closed evidence loop

Matrix, next-level tasks, review, and feedback

Eight steps close with feedback

Growth process 8 steps

Every step has an explicit owner



Part 4

04

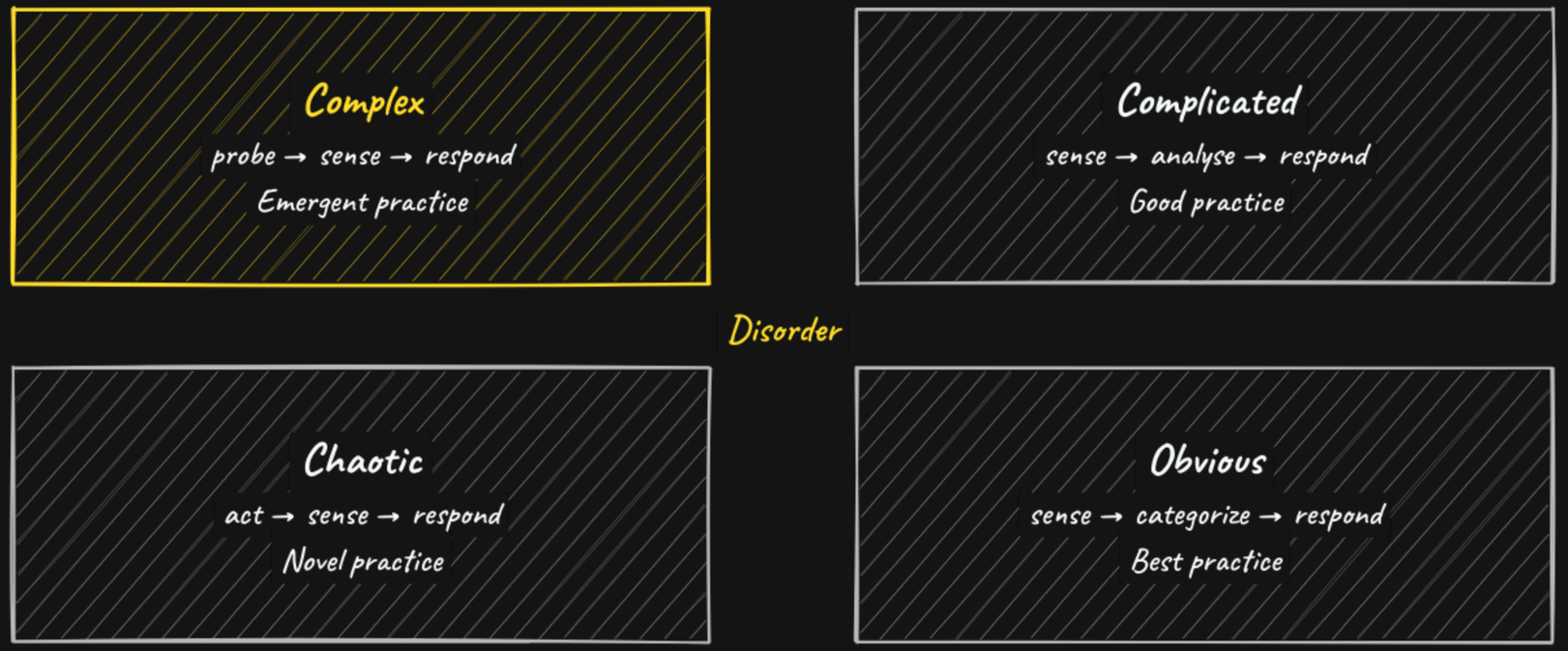
Self-development starts with the problem type

Complex work offers fewer ready-made recipes

Complex has no ready-made recipe

Probe → sense → respond instead of best practice

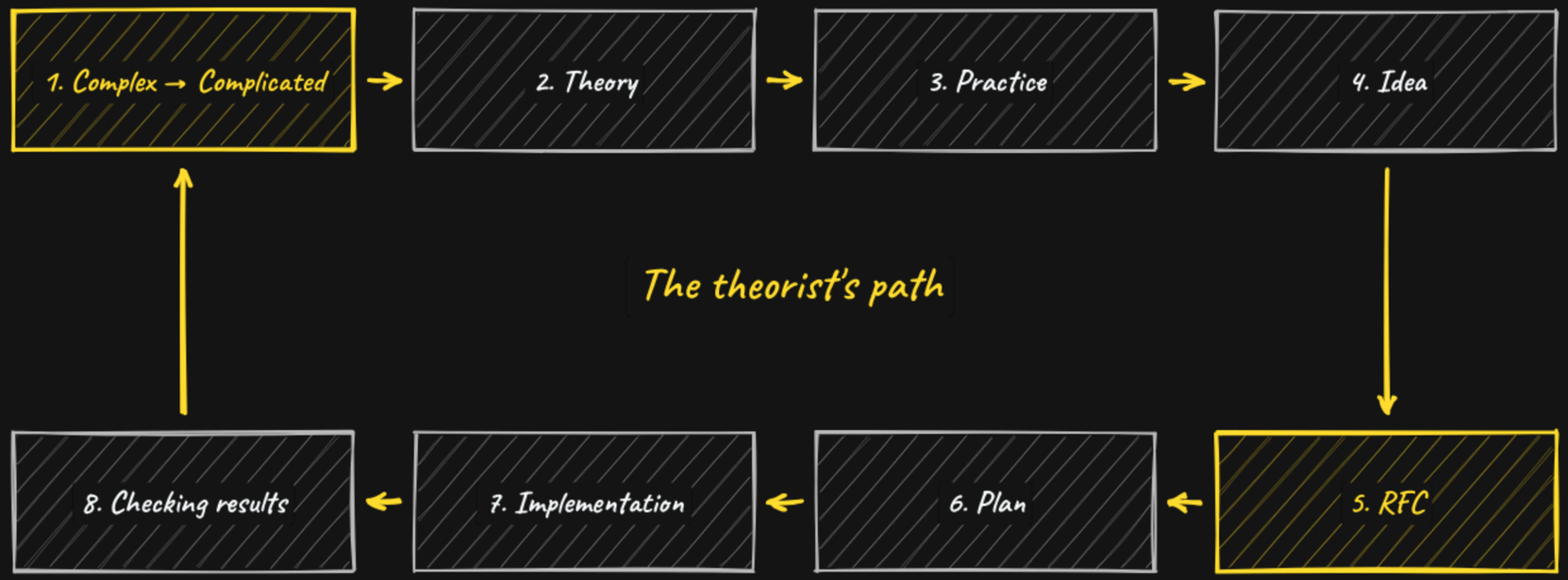
Cynefin 4 domains



Theorists reduce uncertainty first

Complex problem Theory

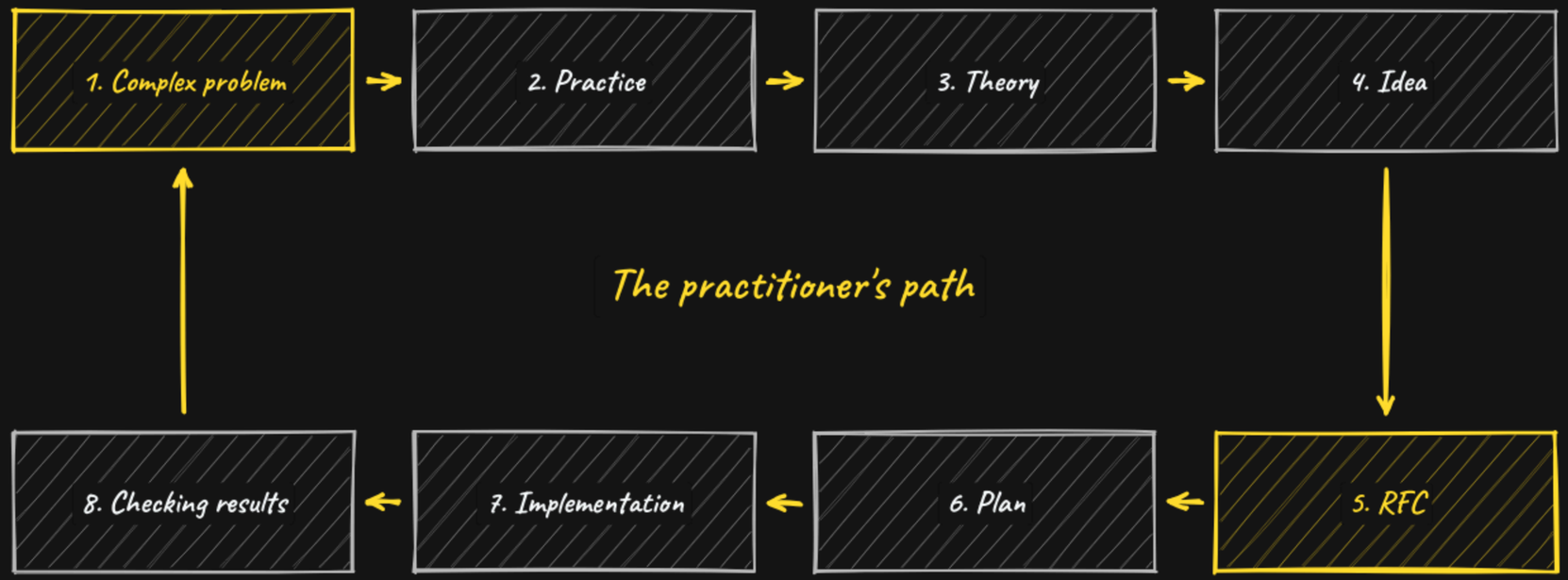
Complex Problem becomes Complicated Problem



Practitioners begin with action and observation

A mature loop brings RFC and Plan back

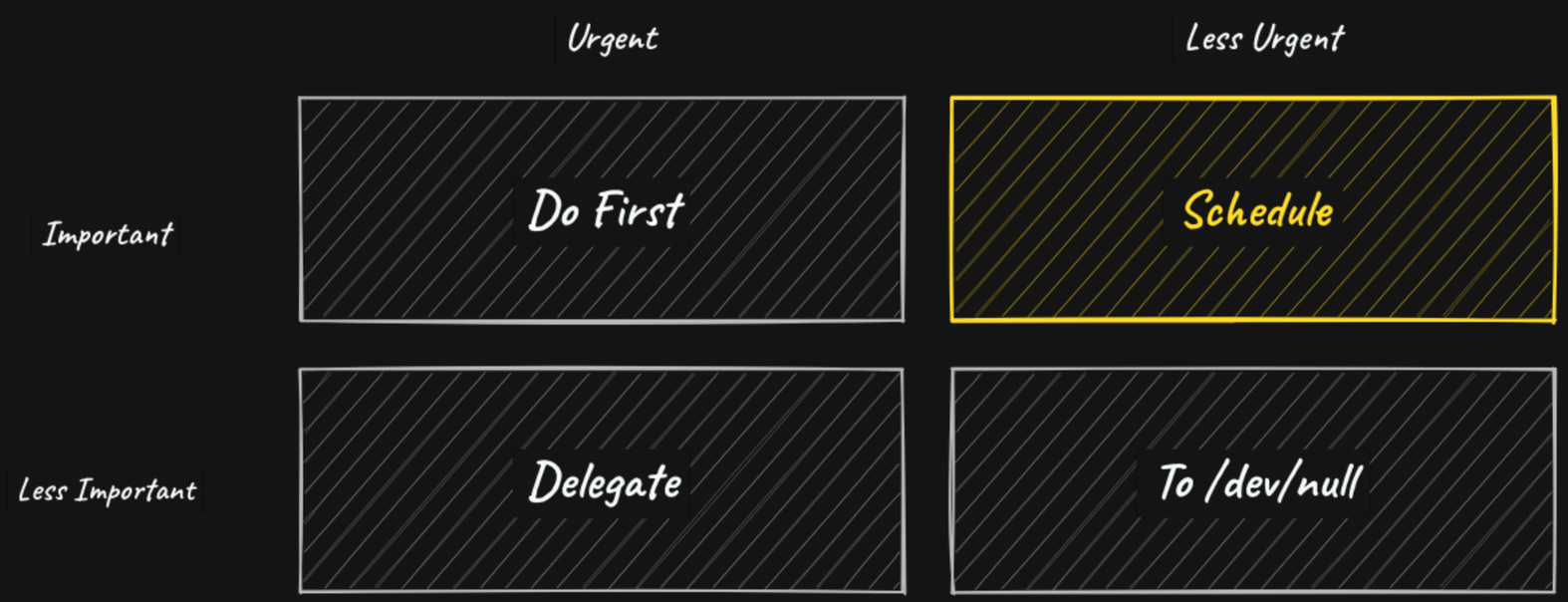
Complex problem Practice



The right work lives outside urgency

Motivation Eisenhower

Do First · Schedule · Delegate · To /dev/null



Growth lives in important, non-urgent work

Schedule 4 steps

Goal → plan → focus on steps → enjoy results

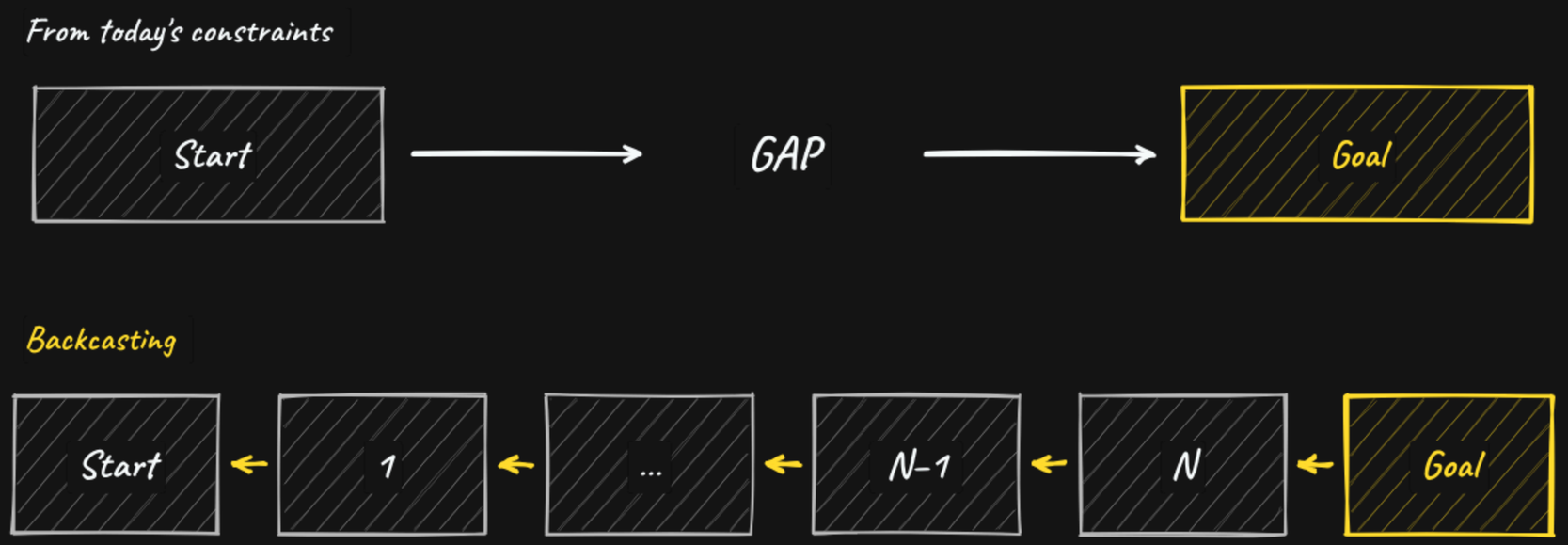


Important but non-urgent work needs space in the schedule

Planning Backcasting

Build the plan backwards from the goal

The standard approach boxes thinking inside today's situation



Part 5

05

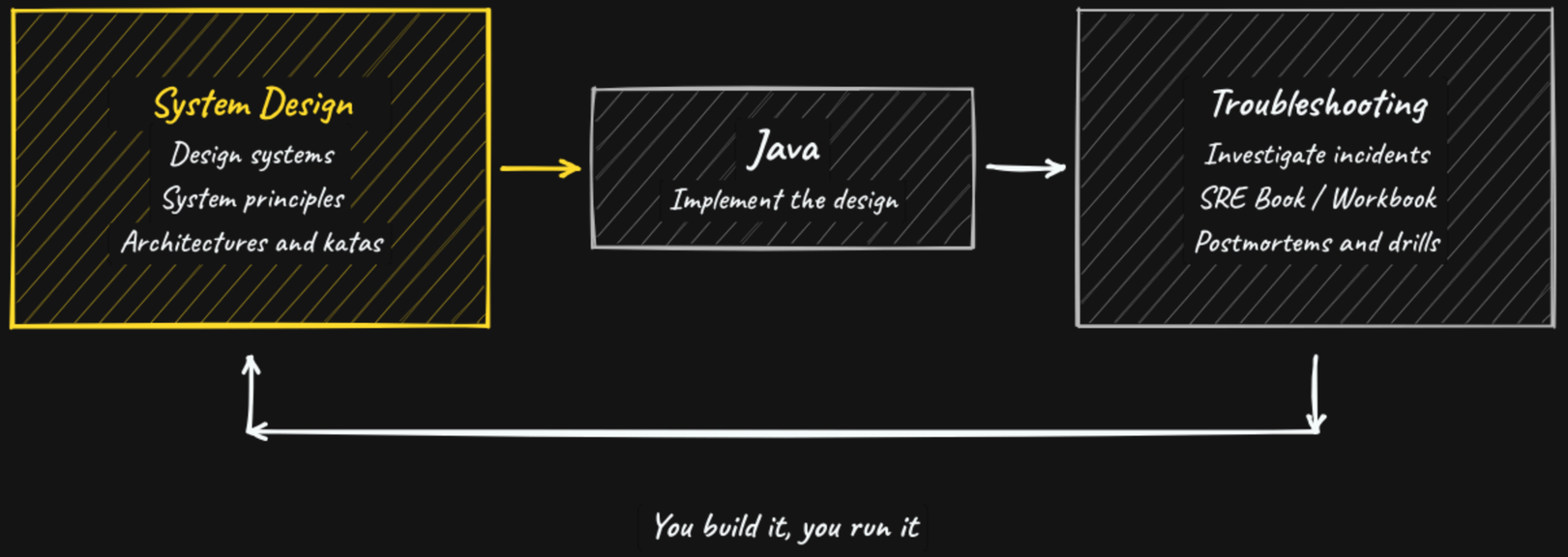
An engineer grows along three axes

Design it, build it, debug it in production

Design, implement, and investigate incidents

System Design + Java + Troubleshooting

You build it, you run it 3 skills



After Senior, choose how to create impact

Takeaways 5 lessons

The career track follows the work, not the other way around

- Two career tracks diverge after Senior
- Staff+ requires demand and proven impact
- Growth runs through matrix and review
- Complex problems connect theory with practice
- Engineers develop three connected skills

- Choose the work and mode of influence; the level records the result.

Materials for further study

Materials Links

Links that appeared in the talk

- Talk materials: ↗t.me
- SRE Book and SRE Workbook — Google
- Staff Engineer — Will Larson
- Cynefin framework

polomodov.tech

Thank you!

Slides, links, and materials are in the Telegram channel

Alexander Polomodov

Technical Director & Fellow,
T-Technologies



@book_cube