DotNext Spring · June 27, 2022

How to grow when you are already Senior

Senior Software Engineer: the fork, the branches, and how to walk them

DOTNEXT

Alexander Polomodov

Speaker Hello

I lead digital ecosystem development at Tinkoff

- I own architecture questions across the company
- I curate our System Design Interviews
- I interview technical leaders on management
- I run troubleshooting interviews for SRE engineers

From the fork after Senior to goals and learning

What we'll cover

- Why growing past Senior is a problem at all
- The two branches: team and tech
- Approaches to learning and to motivation
- My recommendations for the technical branch

Part 1

01

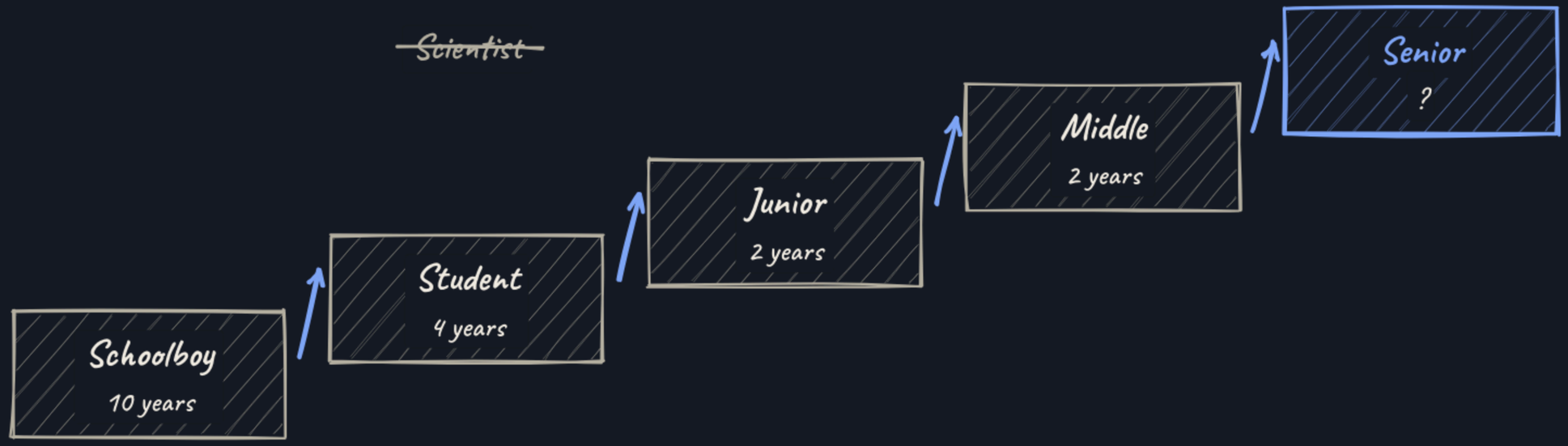
What standard growth looks like

Levels against a timeline

The ladder ends at Senior

Ladder `Standard growth`

What standard developer growth looks like



The talk's growth model · durations are illustrative

After Senior comes a choice of direction, not another rung

Part 2

02

Growing into management

Why a team lead exists, and what comes after

The team lead is the team’s interface to the customer

Why a team lead is needed

- A cross-functional team closes the whole loop
- The loop: idea → requirements → design
- …development → deploy → operations
- The customer brings the idea and gets the results

How I answered this for myself ten years ago

Why a Senior Engineer would want this

- More influence over the end result
- ...both in quality and in quantity
- Improving the development process inside the team
- Raising the level of total compensation

The CTO role moves along three axes, not a ladder

Beyond After the lead

What comes after the team lead

- Process Orientation and Technical Capabilities
- The third axis is Scale: ~10, ~50, ~150
- Startup Coder, Engineering Manager, org structure
- Delivery manager and Architect are axis extremes

Part 3

03

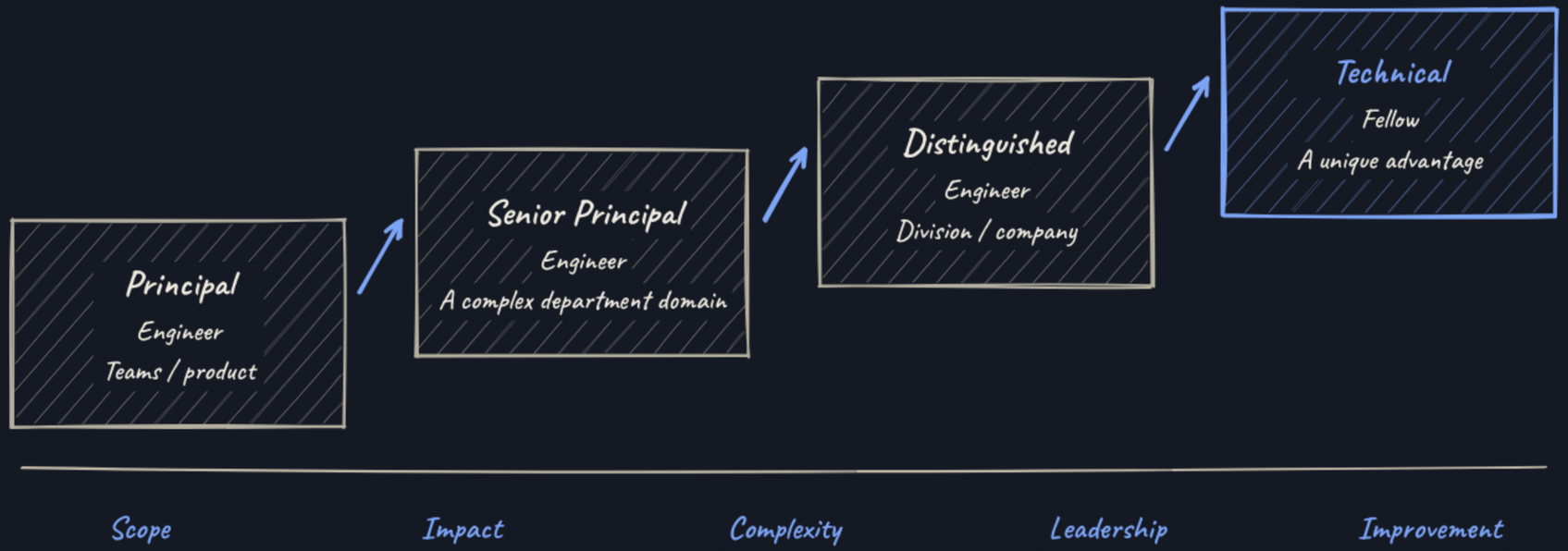
Growing into the technical branch

Individual contributor grades

Grades IC

IC growth means broader, harder problems

High levels of individual contributors



Part 4

04

Motivation and goals

How not to quit halfway

Growth is important and never urgent

Priorities Eisenhower

How to keep yourself motivated to grow

- Urgent and important — Do First
- Important and not urgent — Schedule
- Urgent and less important — Delegate
- Neither — To /dev/null

Set the goal — then get it out of your head

What to do with important but non-urgent tasks

Four steps

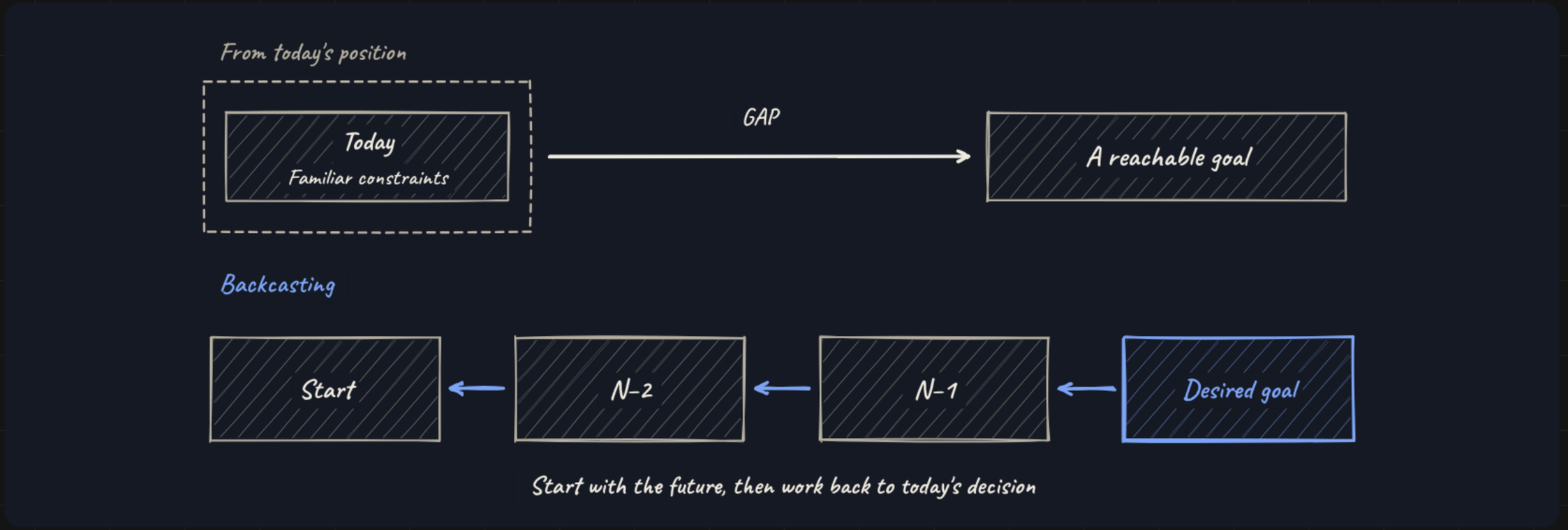
Important, not urgent

- Set a goal that serves several sides of your life
- Make a clear and achievable plan
- Get the goal out of your head so it stops demotivating
- Move forward, enjoying the small steps

Plan backwards from the desired future

Planning Backcasting

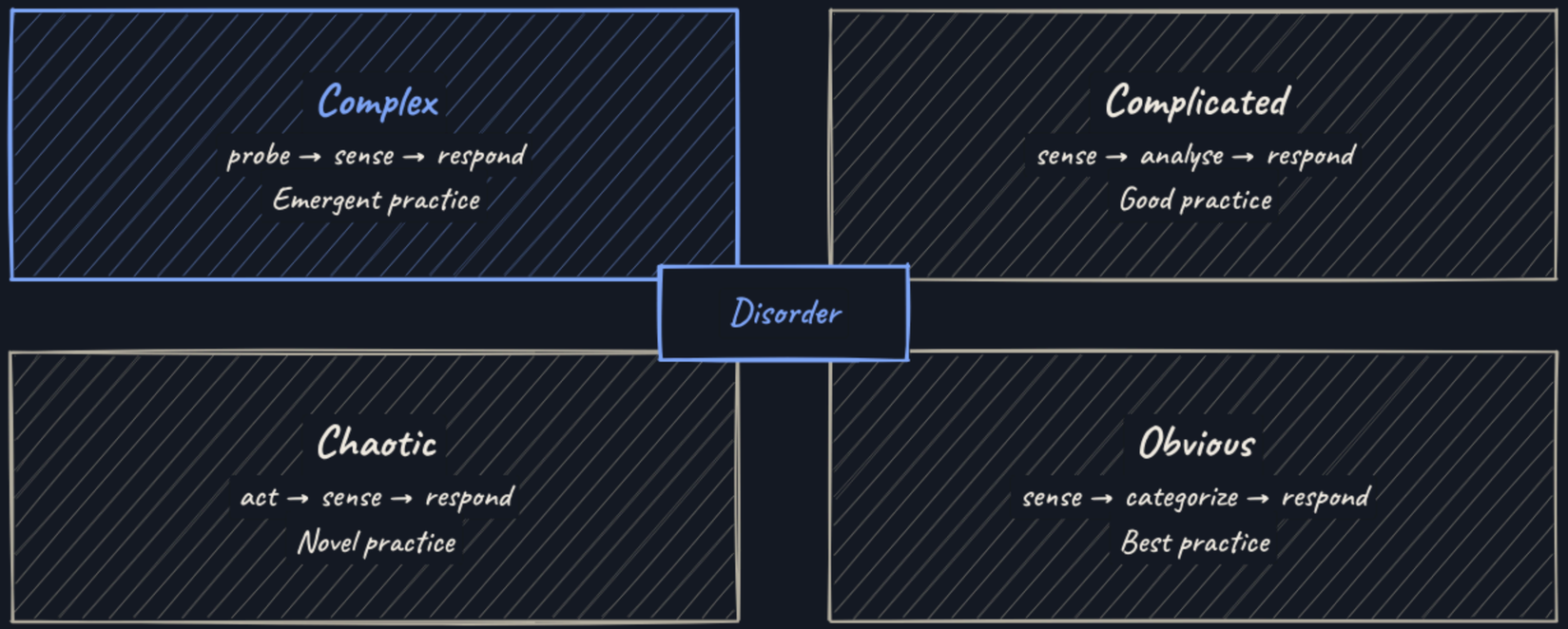
How to set goals and build plans effectively



The problem domain determines the response

Domains Cynefin

The Cynefin framework



Part 5

05

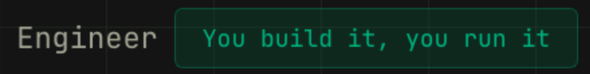
How engineers should grow

You build it, you run it

An engineer owns the system end to end

What «you build it, you run it» means

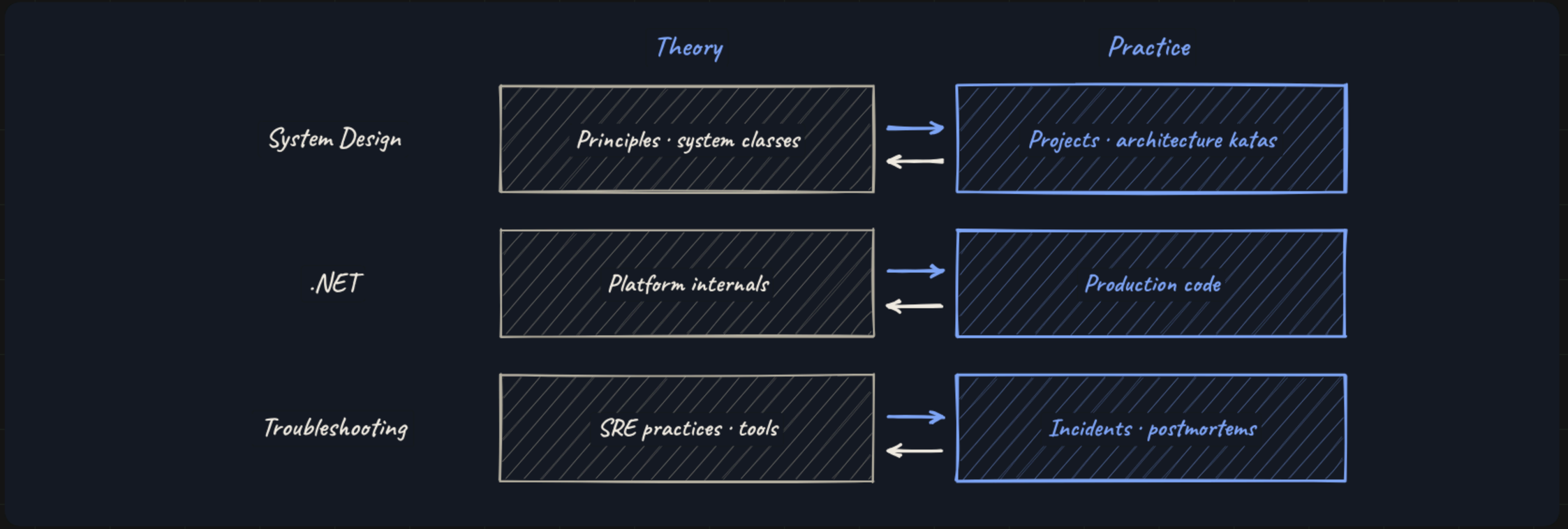
- Can design
- Can implement what was designed
- Can handle incidents in their own systems



Growth · Theory and practice

Three tracks pair theory with practice

And how an engineer should level up



Wrap-up Takeaways

There is no ladder after Senior — there is a fork

What to take with you

- Standard growth ends at Senior
- Past it comes a fork: the team or the tech
- The tech branch has its own grades and matrix
- Set goals backwards from the goal, not from yourself
- Growth is always important and never urgent

Links for further study

The sources from the talk's closing slide

- An article based on this talk
- On motivation: «The Motivation Myth», «Peak Performance», «Lifelong Kindergarten»
- On System Design: two articles and a public interview
- On troubleshooting and on growing into management

polomodov.tech

Thank you!

All slides and links are in the Telegram channel

Alexander Polomodov

Technical Director & Fellow,
T-Technologies



@book_cube