facebook

INFRASTRUCTURE

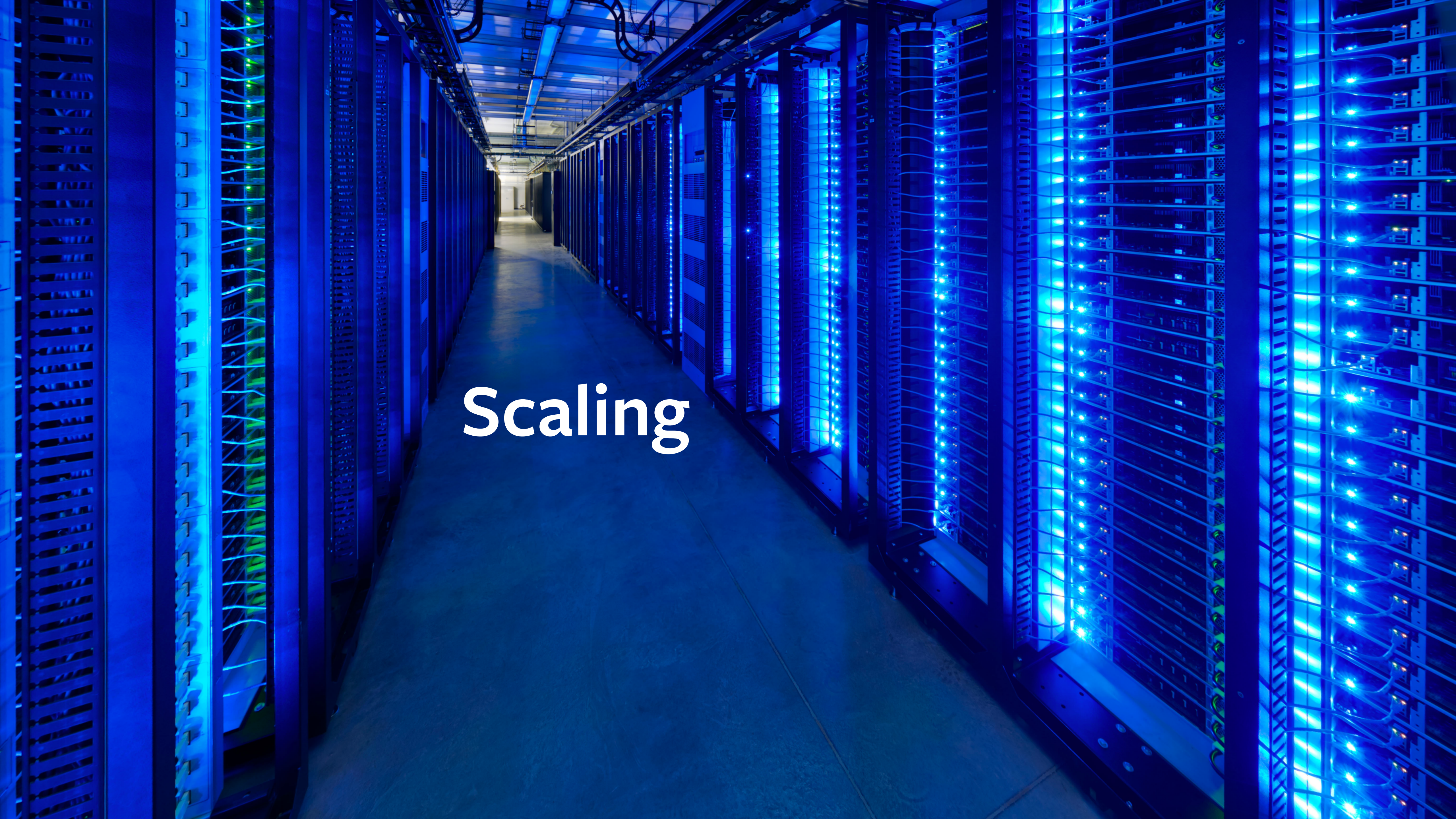
How (Not) to Scale a Project: A Post-Mortem

Giacomo Bagnoli

Production Engineer, Synthetic Network Monitoring

Agenda

- Scaling
- Keep your customer in mind
- Production Engineering
  - System design
  - Software principles
- Project phases



Scaling

Project Phases

Start Up

Scale

Awesomize

Project Phases

Start Up: From prototype to POC

Scale

Awesomize

Project Phases

Start Up: From prototype to POC

Scale: Make it to production

Awesomize

“Infinite” infra: a curse in disguise

rot

Scale: M

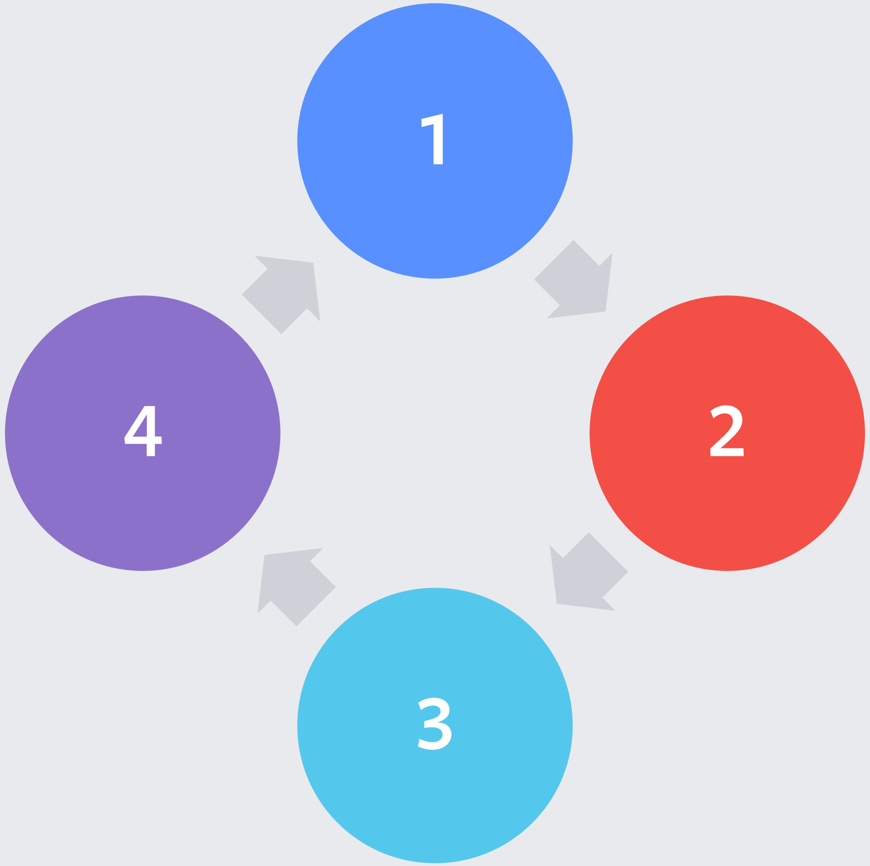
Make it bet

Customers

Keep your customer in mind

1

Put yourselves in user´s shoes - feel the pain (attend postmortems)



2

When planning features, keep the customer in the loop

3

Involve customers in review/test of the feature

4

Communicate early and a lot: set expectations

Production Engineering

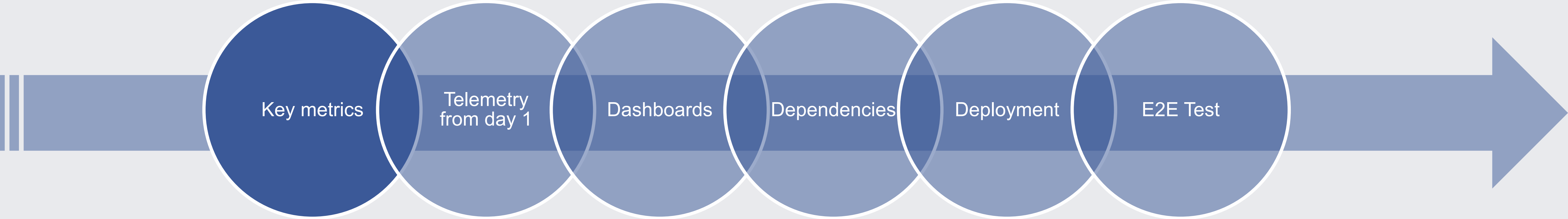
System design

Make it easy to operate

- Key metrics
- Telemetry from Day 1
- Dashboards
- Dependencies
- Deployment
- E2E Test

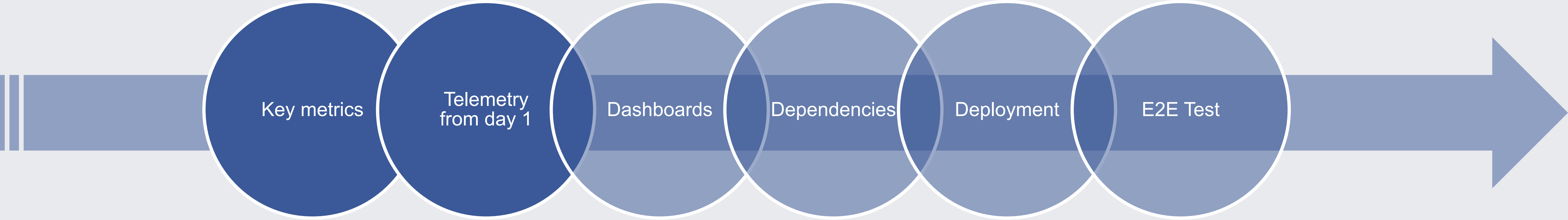
System design

- Key metrics
  - Decide on service metrics with your customers
  - Decide on service health metrics within the team



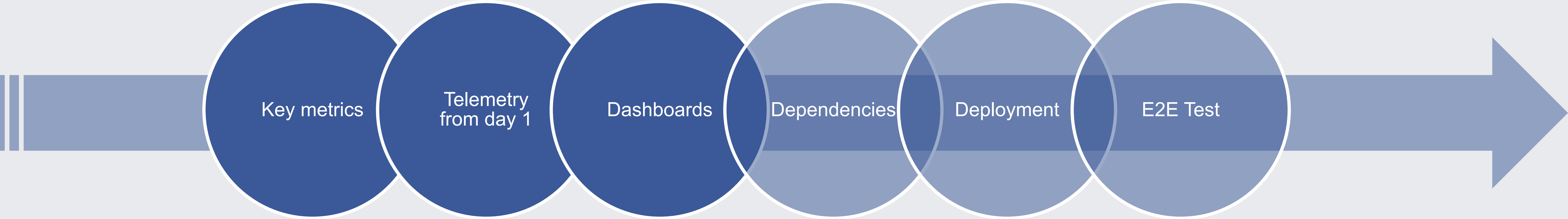
System design

- Measure your key metrics from day 1
  - Use timeline view for your key metrics



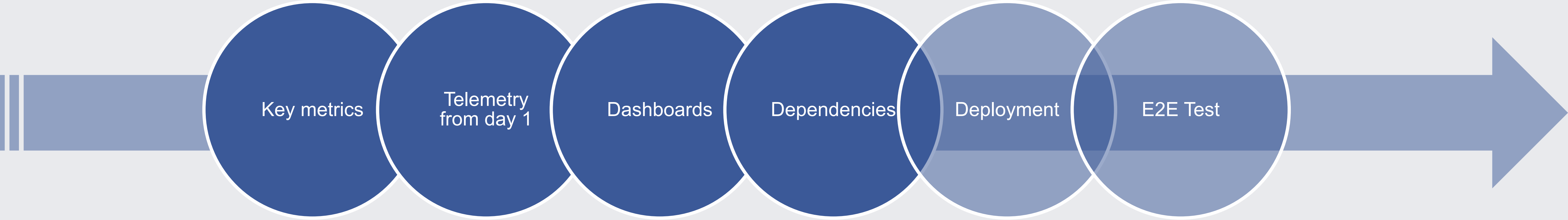
System design

- Different views:
  - Customer dashboard with service outcome
  - Owner dashboard with service health



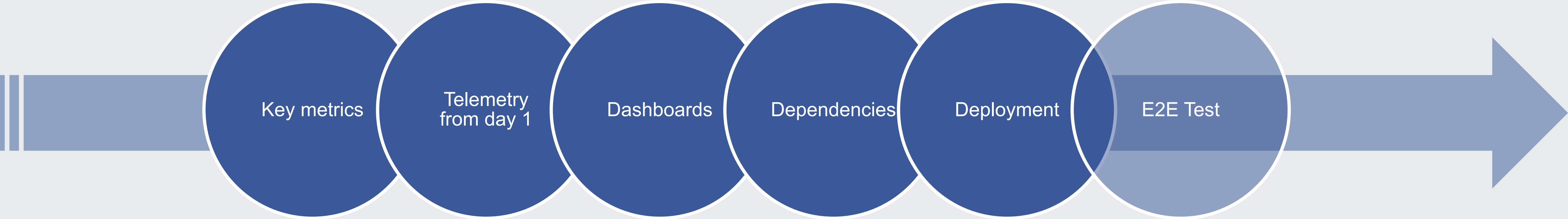
System design

- Dependencies
  - Track response times and failure rates
  - Graceful dependency failure



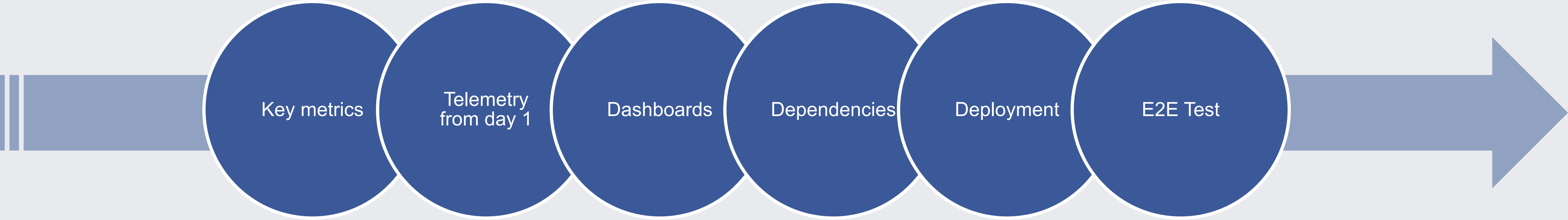
System design

- Deployment
  - Small and often
  - Create and use automated deployment tools



System design

- E2E Test
  - Super integration test
  - Completely independent of your system



Software Principles

Make it easy to maintain and evolve

Software Architecture

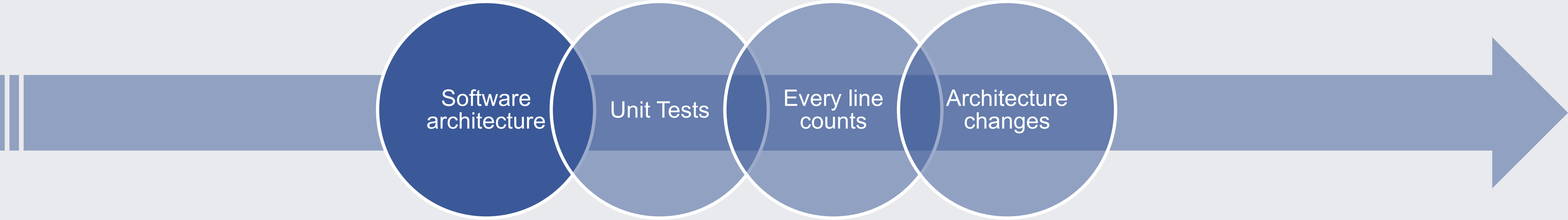
Unit Tests

Every line counts

Architecture Changes

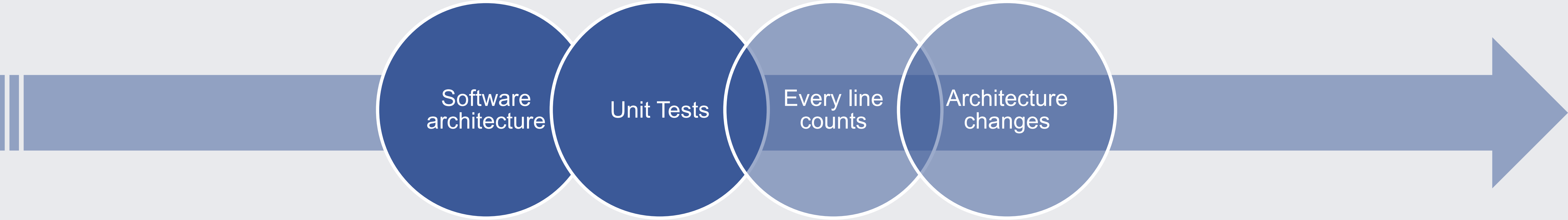
Software principles

- Software architecture
  - Design meetings
  - Changing a production system



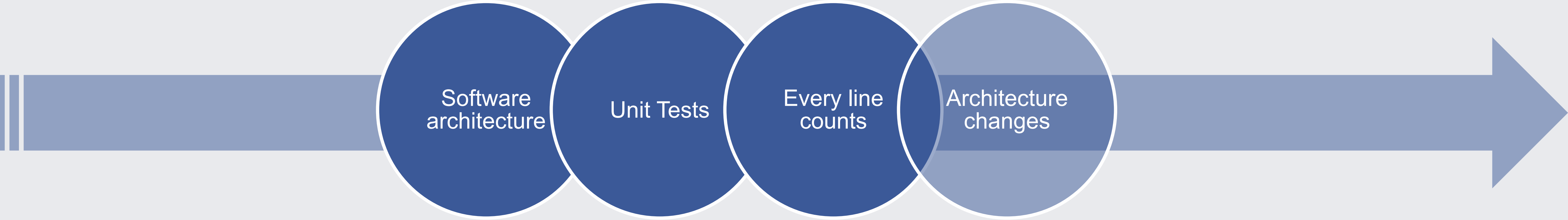
Software principles

- Unit Tests
  - Write testable code
  - Clean interfaces hiding implementation
  - Test plan



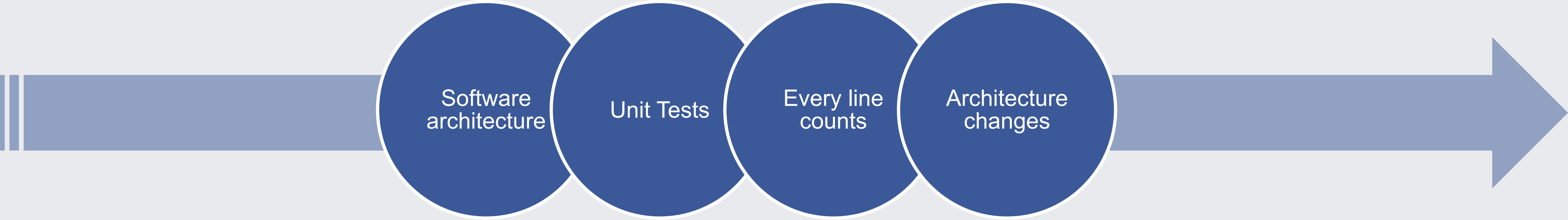
Software principles

- Every line is a liability
  - Test and code are equally important
  - Avoid the premature optimization trap
  - Focus on the problem to solve



Software principles

- Architecture changes
  - Discussed as team internally
  - Announce expected improvements in key metrics to customers

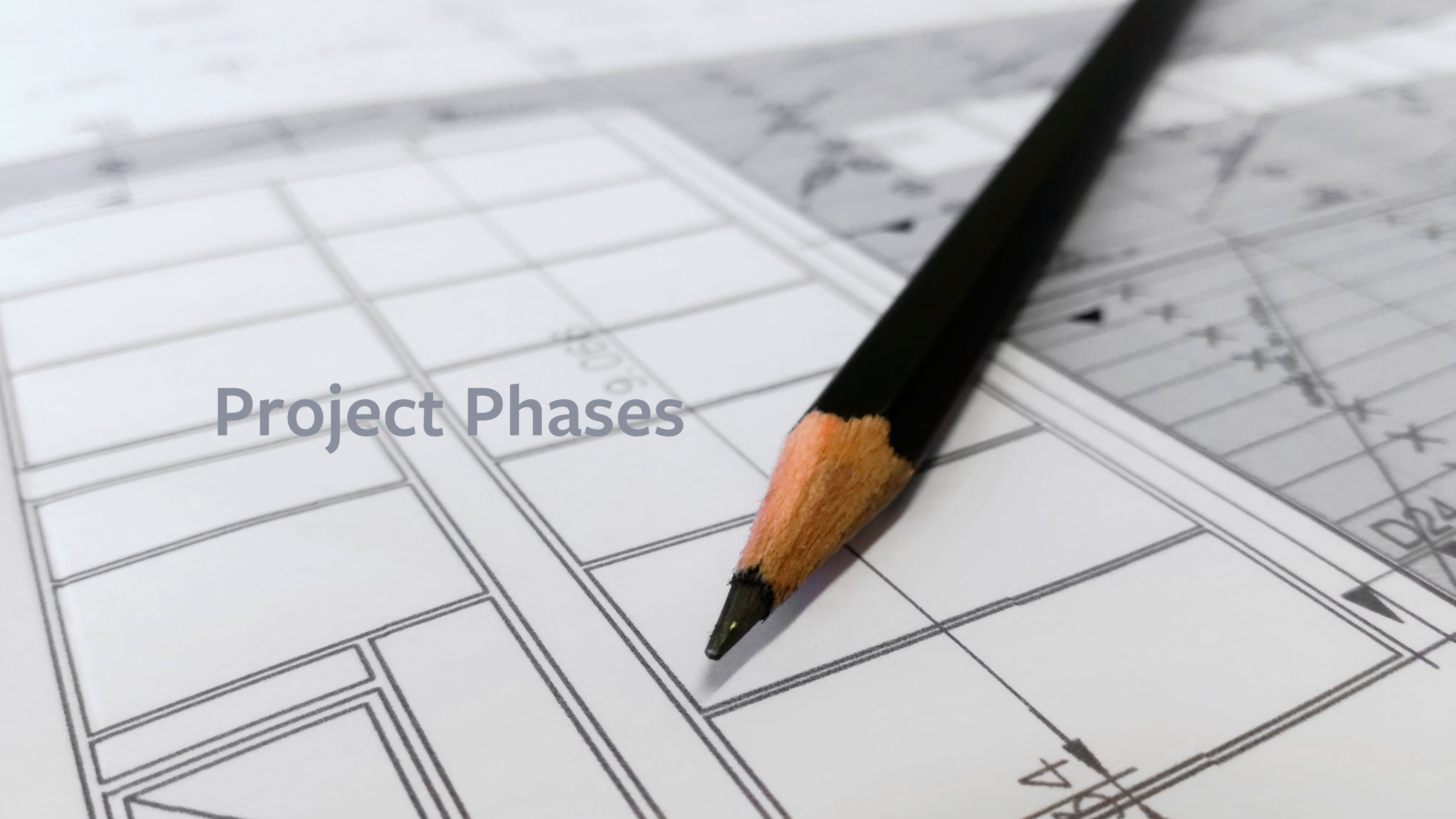


Software architecture

Unit Tests

Every line counts

Architecture changes



Project Phases

Start Up: Metrics are defined and already stored

Scale: Components/Dependencies stay under control

Awesomize: it’s working (for real)

- Identify where you are at: for the team and for your customers
- Know when to switch phase and communicate

Thank you

facebook