

InfoQ Certified Engineering Leadership Program syllabus

Five-week online certification cohort. Four hours a week.

Week-by-week syllabus

Each week pairs a QCon talk with a 4-hour live working session. You apply the week's material to the leadership decisions in your own work, alongside a small cohort of senior engineers and software architects from different companies, each with a minimum of five years' experience.

Week 1: Organizational Foundations

- **Focus:** Being a good technical leader means understanding how organizations work, how to influence them, and how to change people's behavior. This week covers the foundations of working inside an organization, drawing on behavioral economics, industrial-organizational psychology, and a bit of anthropology, so you can assess and work within your own organization.
- **Sample discussion task:** Reflect on your organization in terms of Westrum's model for organizational culture. Would you characterize it as pathological, bureaucratic, or generative? Why? How do you navigate those aspects of your organization?
- **Weekly homework:** Map your part of the organization. Where do decisions actually get made, who influences them, and where does your read differ from the org chart? Bring one place where the informal structure and the formal one disagree

Week 2: Technical Strategy

- **Focus:** This week covers how to identify a business problem that matters, analyze it, and build a (socio)technical strategy for it. You work through systems thinking, extend the previous

week's discussion of behavior change, and cover the product management, software architecture, and systems engineering that a good strategy has to account for.

- **Sample discussion task:** Describe your last large project. What business goals was it addressing? How did you know whether it was successful? What did you have to change as you worked based on feedback from reality?
- **Weekly homework:** Draft a one-page (socio)technical strategy for a current problem in the organization. Name the tradeoff you are choosing and the option you are rejecting, and say what would tell you the strategy is wrong.

Week 3: Technical Execution

- **Focus:** Ideas are free. Execution is everything. This week covers how to execute a technical strategy: risk-driven development, delegation, and the mechanisms that keep the work matched to the strategy. The material draws on planning methodologies and software engineering research.
- **Sample discussion task:** Think about your last major project. What was a risk that was unknown at the start but encountered late? What would you have done differently knowing that risk existed? (Ask the group to contribute ideas for early management of the risk.)
- **Weekly homework:** Write a plan for solving a problem in your organization. The plan should demonstrate a risk-driven approach: work is ordered by risk, risks have a set of mitigation options, and decisions are intentionally delayed to the latest responsible moment.

Week 4: Measurement and Accountability

- **Focus:** The map is not the territory. This week covers how to tell whether your technical strategy is working. It also covers what it means to be accountable for an outcome in an organization, and how to hold that accountability while keeping blamelessness and psychological safety intact.
- **Sample discussion task:** Think about behaviors in your organization that are overlooked or even rewarded as long as nothing bad happens. What would you want to change to mitigate the risk of Outcomes Bias? How would you balance that with accountability?
- **Weekly homework:** Brainstorm a list of potential ways of measuring the success of a business problem you are solving. Classify them as whether they measure participation,

knowledge, behavior, or outcomes. For each measure, describe a way you would also do qualitative sampling to avoid the consequences of Goodhart's Law.

Week 5: Safety, Sustainability, and Resilience

- **Focus:** The software development lifecycle does not end when the code is deployed. Maintenance and operational costs often dwarf the original development cost. This week covers the full picture of running a software system, including capacity planning, support and operational cost, and incident response. Most of the session goes to the capstone presentations.
- **Sample discussion task:** What's a decision that is made by leadership in your organization that would be better made by someone closer to the work? Why do you think the decision was "pulled up" to leadership? How does this affect the team's ability to respond to incidents?
- **Capstone Project:** The capstone is introduced in Week 1 and built throughout the program. The deliverable is a 20-minute group presentation followed by a peer discussion. End on a privacy or security question your group is still working out.