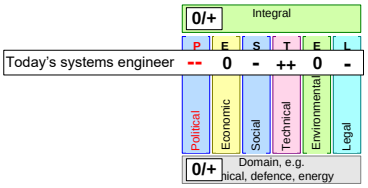
Developing Ecosystem Competences

-



Gerrit Muller
USN-SE
Hasbergsvei 36 P.O. Box 235, NO-3603 Kongsberg Norway
gaudisite@gmail.com

Abstract

This presentation shows what competencies will help across ecosystem organizations.

version: 0.1 status: preliinary draft May 25, 2026

1 Introduction to Ecosystems Competence Development

Working in Ecosystems requires specific capabilities from organizations in the ecosystem. The people forming the organization need specific competences to enable the organization to develop the capabilities. Some of the required competences are:

- Orchestration
- Leadership
- Breadth of perspectives
- Combining agility and rigor

2 Orchestration

Within and between organizations there are many boundaries, see Figure 1. Boundaries are hurdles for communication; they hinder the information flow. They also hinder rational and emotional sharing of understanding. Orchestrators are individuals that are able to work across boundaries. They make connections beyond boundaries. They build and maintain informal networks, By being proactive in contact across layers they make delivery across organizations effective. Orchestrators are the countermeasure for bureaucracy in and between organizations.

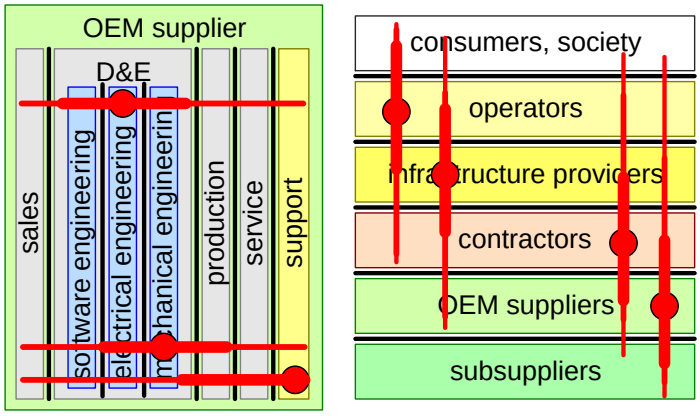


Figure 1: Inter- and Intra-organization boundaries trigger the need for Orchestrators

Figure 2 summarizes the orchestration competency. We can describe orchestrating as the competence to proactively nudge involved parties towards a fitting solution. To do this, orchestration requires the abilities:

- to relate with a wide variety of stakeholders, to understand their interests and concerns, to connect with them such that they can influence them
- to understand the problem and solution space sufficiently
- coping with ecosystem complexity, uncertainties and unknowns, and helping stakeholders to navigate them

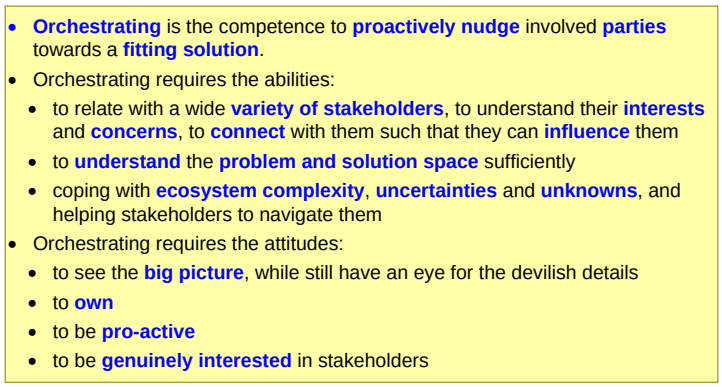


Figure 2: Orchestrating Content and Stakeholders Across Organizations

Orchestrating requires the attitudes:

- to see the big picture, while still have an eye for the devilish details
- to own
- to be proactive
- to be genuinely interested in stakeholders

3 Leadership

Since 2017, TNO-ESI has been offering an extensive competence development program, named Systems and Leadership. This program has 5 modules of in average 2 days, spread over 9 months. We strive to develop the 10 to 18 participants in architecting as well as in leadership. To do this, we have 2 teachers present continuously: one leadership teacher and one architecting teacher. Our experience is that the combination of architecting and leadership is powerful in improving systems engineers effectiveness.

The “eye of leadership” forms the core of the leadership training. Figure 3 shows the eye of leadership at the left-hand side. The vertical axis is self on the

top and other(s) at the bottom. The horizontal axis has feelings left and ratio right. This creates four quadrants:

Own and shape denotes the leadership competence to focus by proactively owning and shaping the problem, goal, and solution.

Deliver is the competence to work with others and deliver a fitting solution together

Manage self is the competence to know yourself to such degree that you manage emotional triggers, ensuring to stay in a learning mindset.

Connect helps to build relations with a wide variety of stakeholders to effectively interact with them.

Core to the success of this program is the combination of emotional and rational perspectives. For most systems engineers, development of social skills is an essential first step for leadership skills. A major challenge for systems engineers is to open up for the emotional and social side.

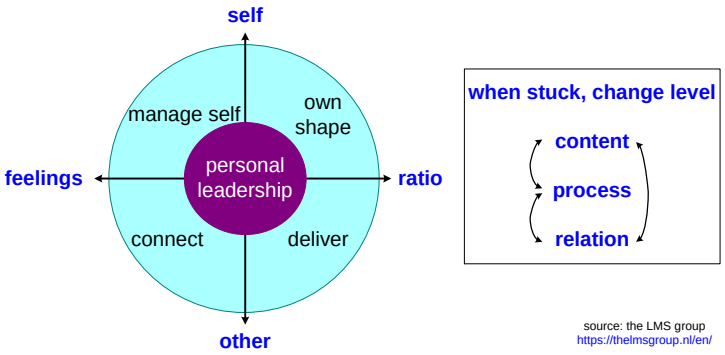


Figure 3: Orchestration builds on leadership, as we teach it in the TNO-ESI Systems and Leadership Program

The right-hand side of Figure 3 shows one of the many techniques from the program. It explains that we communicate at various levels: content, process, and relation. As systems engineers, we tend to have a major content drive. We may move to the process level. However, taking the (emotional) relation level is more challenging. The simple recommendation is that when a conversation is stuck at one level, to move to another level. The rationale is that the cause of being stuck is probably at another level as well. Changing level often helps to get unstuck.

4 Breadth of perspectives

A common framework for perspectives is PESTEL (Political, Economic, Social, technical, Environmental, and Legal). Figure 4 shows this framework with the

addition of a domain specific perspective, and an integral perspective. Example domains are clinical, defense, energy, maritime, manufacturing, information technology, transportation, food, and finance. The integral perspective is the integration of all these perspectives, and especially the tensions between them.

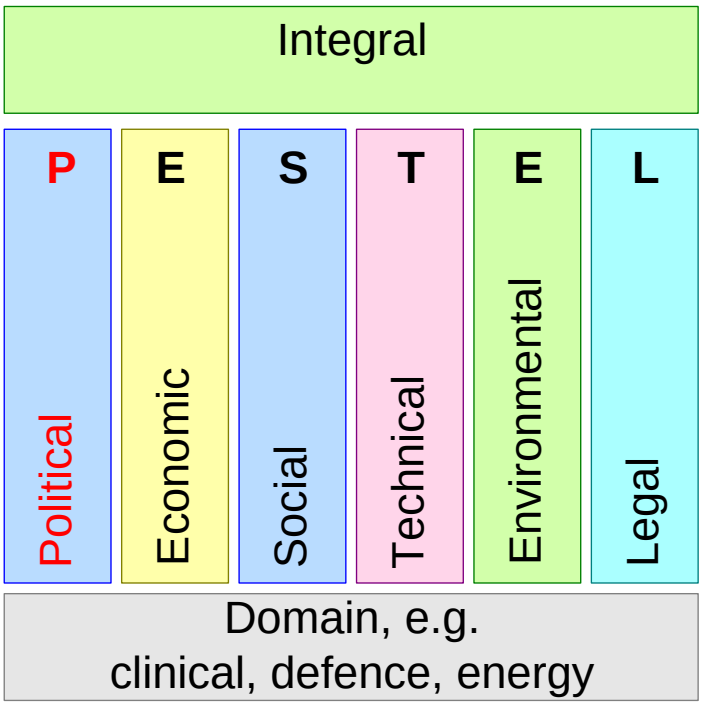


Figure 4: Breadth of Perspectives

I observe that the current generation of systems engineers are broad, however, not that broad. Figure 5 shows my assessment of today's systems engineers:

Technical very competent

Domain specific varies widely

Integral is the core of systems engineering. However, the limited coverage of non-technical perspectives reduces the score.

Economic is neutral. Systems engineers know that economic considerations dominate in decision making. At the same time, the majority of systems engineers that I have seen dislikes the money dimension.

Environmental seems to more appreciated, however, the lack of political priority degrades the attention and hence the score.

Legal is boring to systems engineers; a boundary condition to comply with

Social is poor in general, although it may improve with age. The nerd roots of systems engineers need complementing. This perspective will greatly benefit from social competence development of systems engineers.

Political is very dominant, certainly in the higher ecosystem layers. The power and emotion in politics is in direct conflict with the rational decision-making mindset of systems engineers. Most systems engineers I know severely dislike politics. Unfortunately, given its dominance, we have to find ways to cope with this perspective.

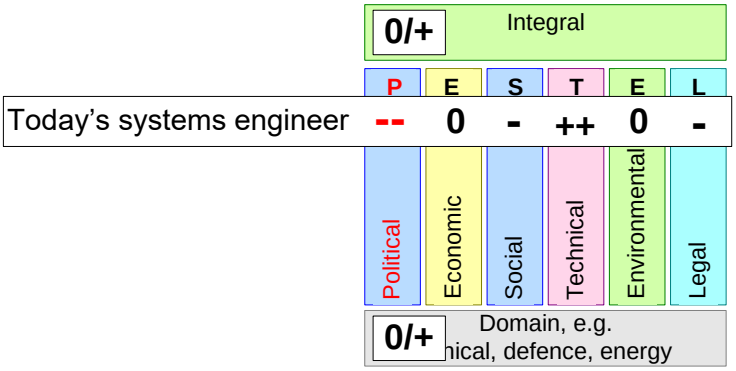


Figure 5: Today's systems engineers need signficant growth

5 Combining agility and rigor

When disruptive events hit an organization or society, then the organization needs to respond fast to survive. Large and mature organization typically have lost their ability to respond fast; bureaucracy slows down the response tremendously. The institutionalization that caused the bureaucracy is there for good reasons:

- it captures past knowledge, and in that way prevents repeating the same mistakes
- it protects against fashion grills and similar emotional upheavals

In that way, it forces organizations to work carefully, and to take time for major decisions. Hence, there is an inherent tension between institutionalization and the need for responsiveness when disruptive events occur.

The Cynefin model, see Figure 6 shows that the way an organization should behave depends on the kind of challenge they are facing. Cynefin defines several classes of challenges:

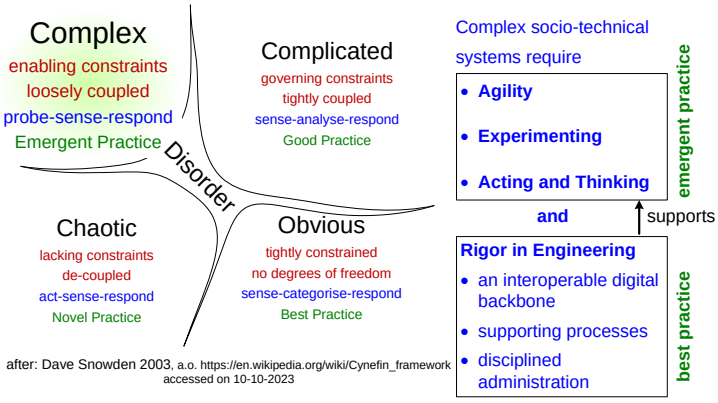


Figure 6: The need is responding fast to disruptive changes: Agility!

Obvious or Clear where challenges are tightly constrained without degrees of freedom. Best practices work well for obvious challenges; past knowledge will do the job.

Complicated where there are governing constraints and problems are tightly coupled. These challenges require competent systems engineers that apply good practice.

Complex where there are fewer constraints, and there is less coupling. Often humans cause complexity, with their emotions, beliefs and other human properties. Complex challenges require a more open-minded approach.

Chaotic or Wicked where we humans have problems to achieve sufficient understanding to reason about the challenge. We have to discover and learn how to address these challenges.

Figure 6 shows that in the complex sociotechnical problems that arise from disruptive events we need agility, experimenting, and an attitude of acting and thinking. However, once we create solutions, we still need the rigor in engineering. Rigor exists in best practices. Effective rigor in engineering enables agility in the probe-sense phases. Examples of elements that help achieve desired rigor are:

- an interoperable digital backbone
- supporting processes
- disciplined administration

References

[1] Gerrit Muller. The system architecture homepage. `http://www.gaudisite.nl/index.html`, 1999.

History

Version: 0.1, date: May 24, 2026 changed by: Gerrit Muller

- Created, article version

Version: 0, date: February 8, 2026 changed by: Gerrit Muller

- Created, as subset of VisionFutureSystemsCapability