

FIGURE 2.1

Typical development of a systems architect.

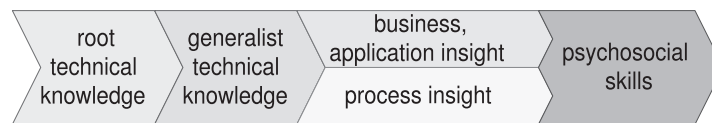


FIGURE 2.2

Generalist versus specialist; depth versus breadth.

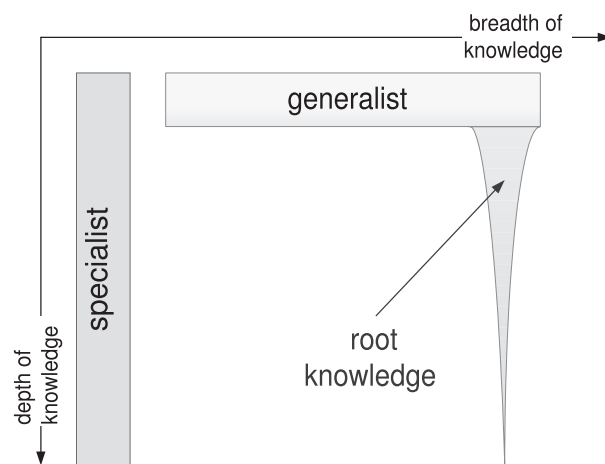


FIGURE 2.3

Generalists and specialists are both needed in complex products; they have complementary expertise.

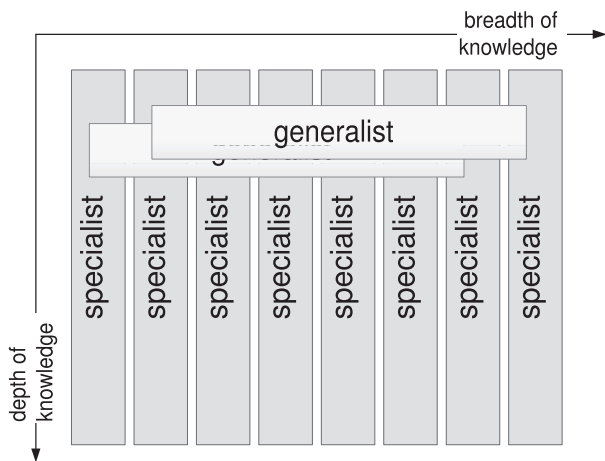


FIGURE 2.4

Growth in technical breadth; intermediate functions from specialist to systems architect.

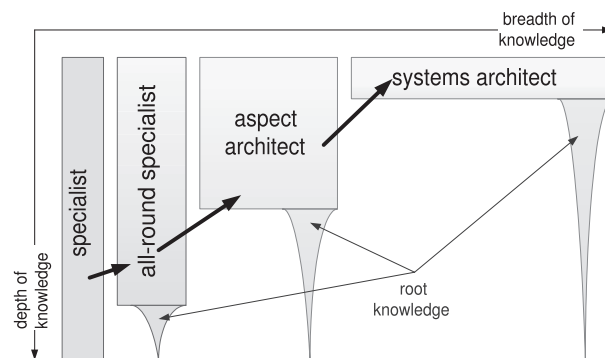


FIGURE 2.5

Classification of architect related titles.

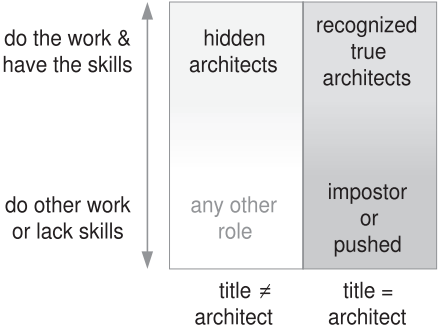


FIGURE 2.6

System roles mapped on the development life cycle.

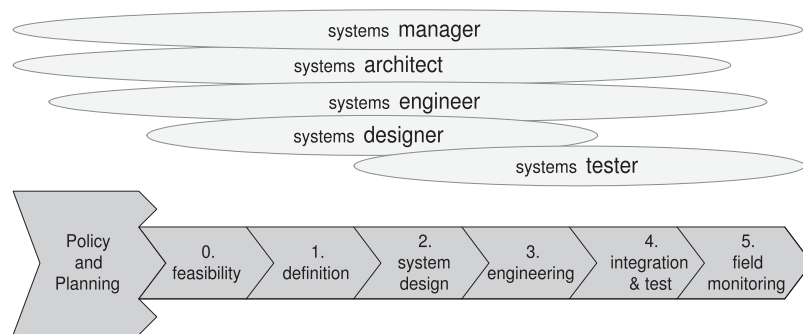


FIGURE 2.7

Deliverables of Systems Architects consists of artifacts forming a stack of paper when printed.

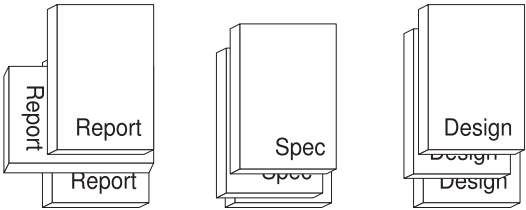


FIGURE 2.8

More specific list of deliverables of Systems Architects.

Customer and Life-Cycle Needs <i>(what is needed)</i>
System Specification <i>(what will be realized)</i>
Design Specification <i>(how the system will be realized)</i>
Verification Specification <i>(how the system will be verified)</i>
Verification Report <i>(the result of the verification)</i>
Feasibility Report <i>(the results of a feasibility study)</i>
Roadmap

FIGURE 2.9

The primary responsibilities of systems architects are not tangible, and measurement is difficult.

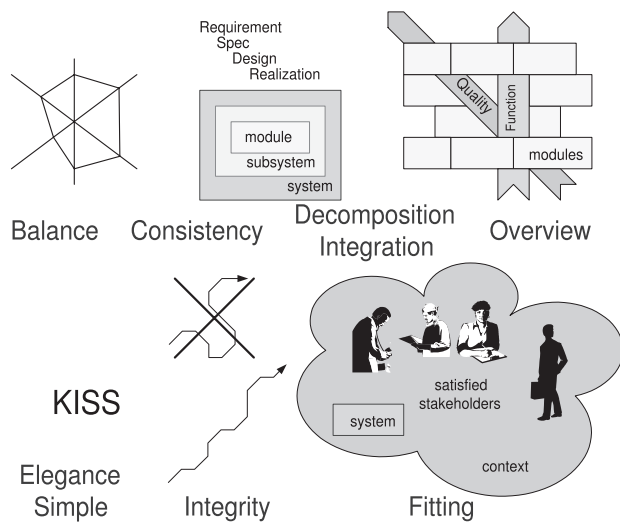


FIGURE 2.10

(Incomplete) list of secondary responsibilities of the systems architect and the related primary owner.

responsibility	primary owner
business plan, profit	business manager
schedule, resources	project leader
market, salability	marketing manager
technology	technology manager
process, people	line manager
detailed designs	engineers

FIGURE 2.11

The systems architect performs a large amount of activities; where most of the activities are barely visible for the environment, they are crucial to the functioning of architects.

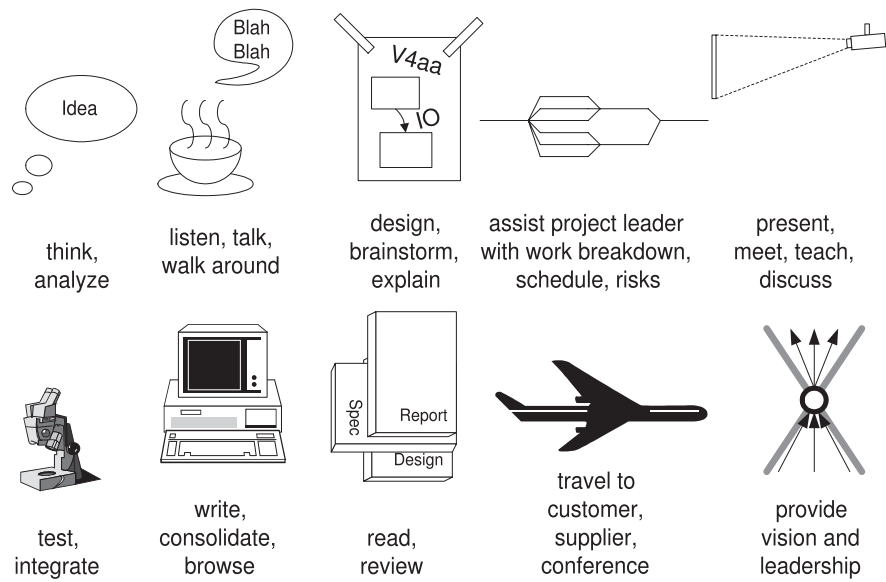


FIGURE 2.12

Bottom-up elicitation of high-level views.

		Quantity per year (order-of- magnitude)	architect time per item
consolidation in deliverables meetings informal contacts sampling scanning	→ driving views	10	100 h
	→ shared issues	10^2	1 h
	→ touched details	10^4	0.5 – 10 min
	→ seen details	$10^5 - 10^6$	0.1 – 1 sec
	→ product details	$10^7 - 10^{10}$	
	real-world facts	infinite	

FIGURE 2.13

The visible outputs versus the (nearly) invisible work at the bottom.

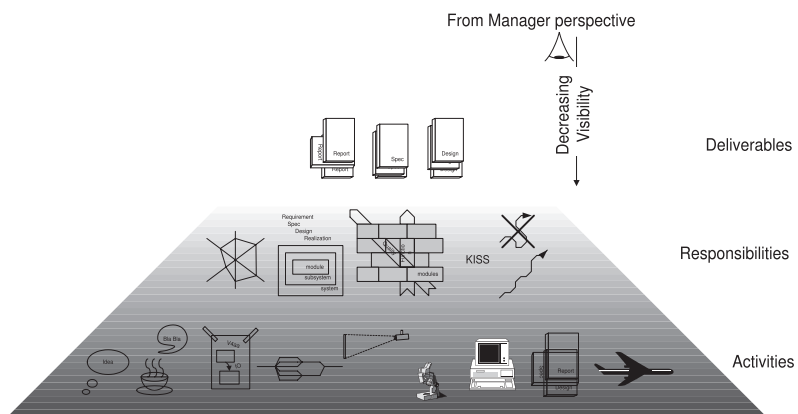


FIGURE 2.14

Connecting system specification to detailed design.

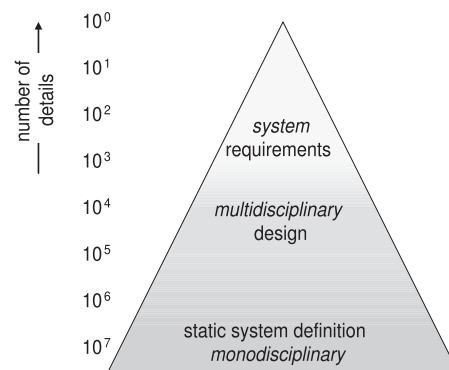


FIGURE 2.15

From system to product family or portfolio.

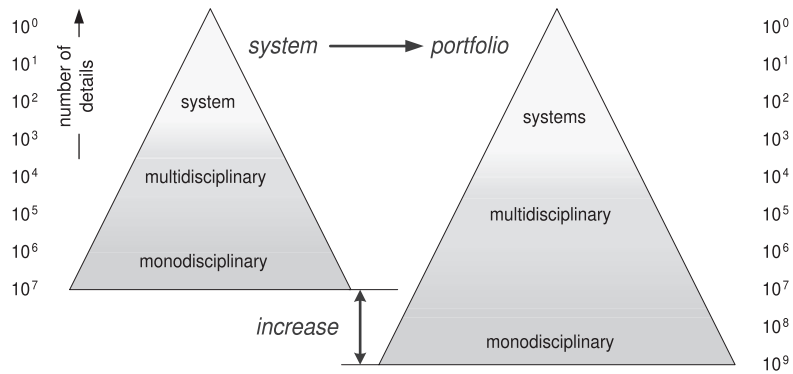


FIGURE 2.16

Product family in context.

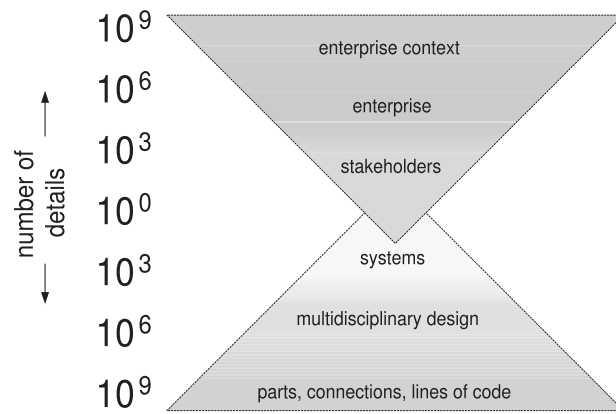


FIGURE 2.17

Architecture: the essence of system and context.

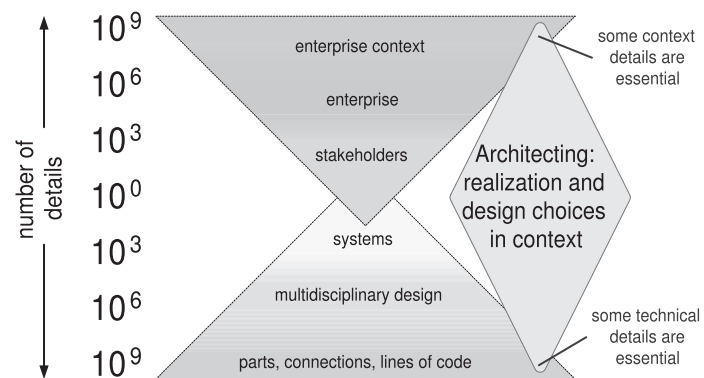


FIGURE 2.18

Positioning design and engineering in the dynamic range of abstraction levels.

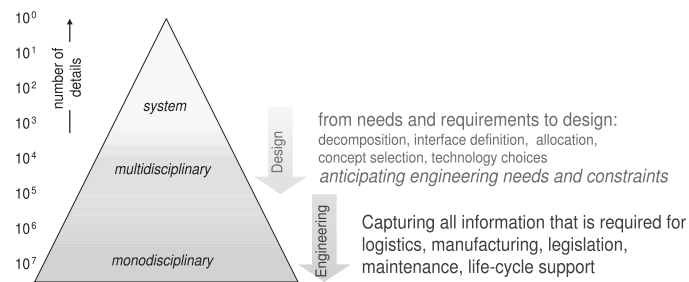


FIGURE 2.19

Frequently observed gaps in practice.

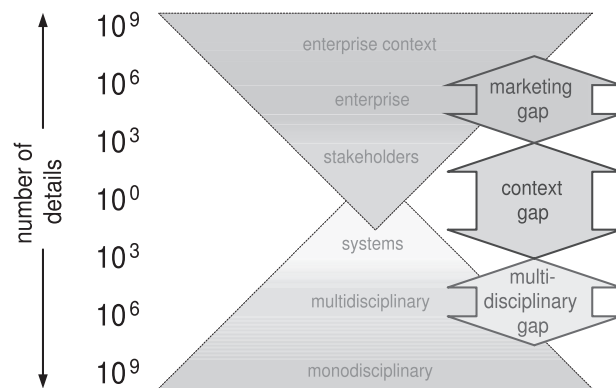


FIGURE 2.20

Interaction styles for architects.

provocation	when in an impasse: provoke effective when used sparsely
facilitation	especially recommended when new in a field: contribute to the team, while absorbing new knowledge
leading	provide vision and direction, make choices risk: followers stop to give the needed feedback
empathic	take the viewpoint of the stakeholder acknowledge the stakeholder's feelings, needs, concerns
interviewing	investigate by asking questions
whiteboard simulation	invite a few engineers and walk through the system operation step by step
judo tactics	first listen to the stakeholder and then explain cost and alternative opportunities