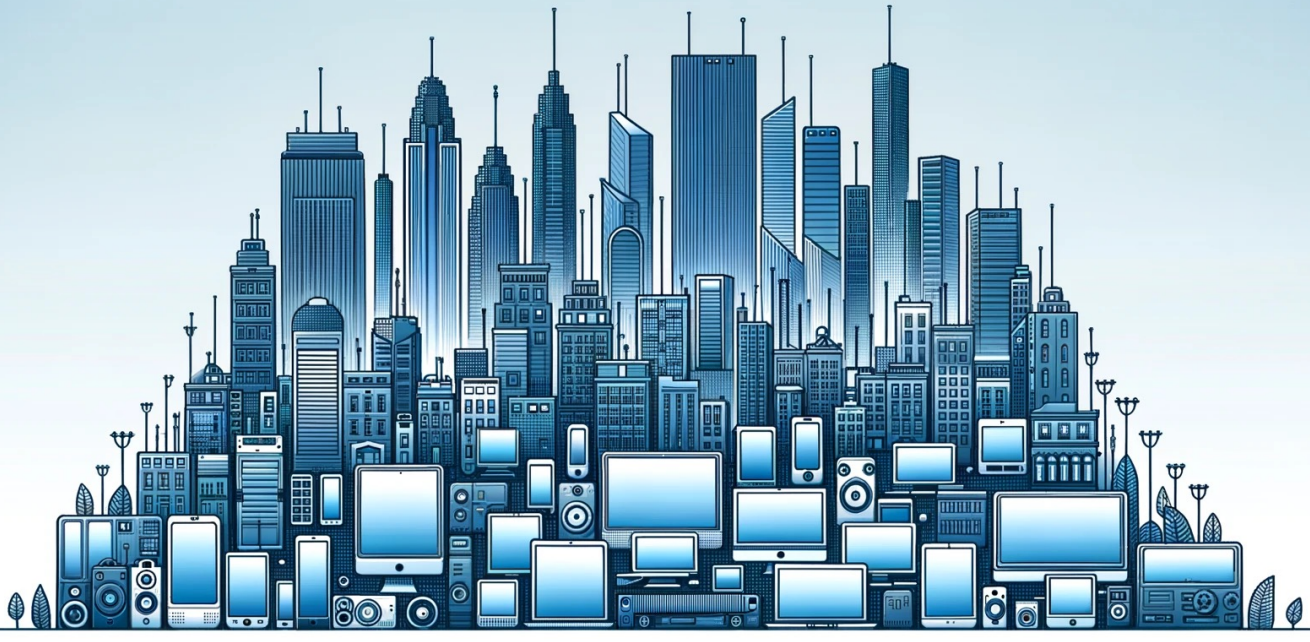
Engineering Clarity: Transforming Ideas into Solutions

according to the proven systems engineering approach



1. Business needs

Identifying business needs that the solution must fulfill, usually aligned with organizational goals and market demands

2. System requirements

Defining the overall capabilities, functionalities, and constraints of the system based on the identified business needs

3. System architecture

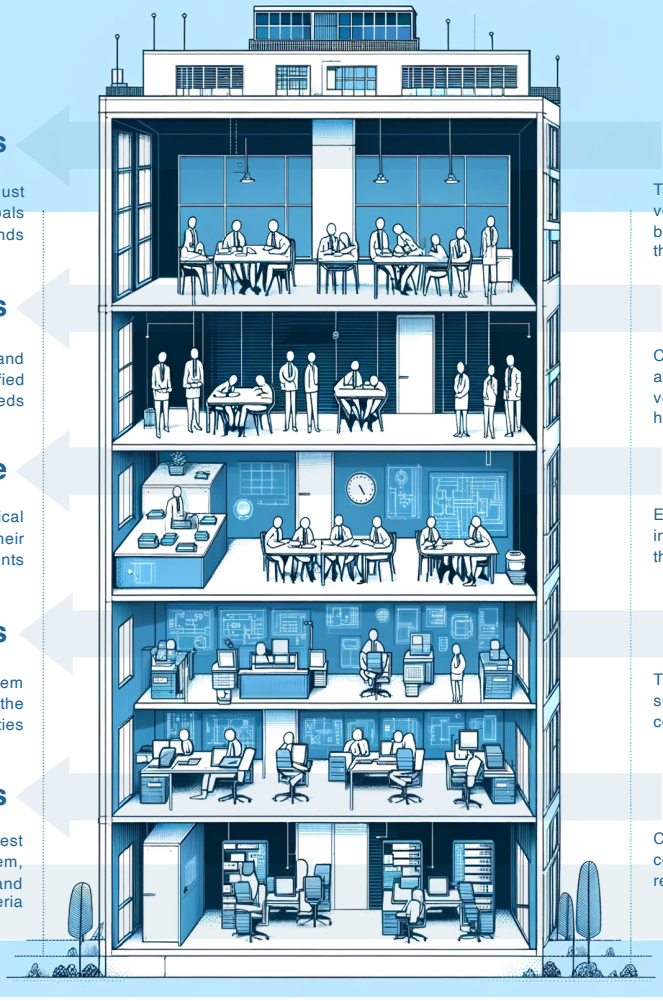
Outlining the system's conceptual and technical framework, including major components and their interactions, to achieve the system requirements

4. Subsystem requirements

Specifying the requirements for each subsystem (like software, hardware, mechanics) within the overall system, detailing their functionalities

5. Unit specifications

Documenting the specifications for the smallest units (components or modules) of the system, detailing their functionality, interfaces, and performance criteria



6. Implementation

The process of implementing (like coding, hardware designing, constructing) the system's components based on the defined specifications and architecture

7. Unit tests

Conducting tests on individual units or components to verify that each meets its specified requirements and performs as intended

8. Subsystem tests

Testing combined units or components as subsystems to ensure they work together correctly and meet subsystem requirements

9. Integration tests

Evaluating the system as a whole by testing the integration of different subsystems, ensuring they work together seamlessly

10. System tests

Comprehensive testing of the complete system against the overall system requirements, verifying that all components function harmoniously within the entire system

11. Acceptance tests

The final testing phase where the system is verified and validated against the original business needs and requirements, often involving the end-user, to ensure it is ready for deployment