JUPITER PRIME

Consulting • Training • Advisory

جوبيتر برايم للاستشارات والتدريب

COURSE BROCHURE

Systems Engineering Fundamentals Training Course

jupiterprime.uk

info@jupiterprime.uk

Registered in the United Kingdom and Saudi Arabia

Systems Engineering Fundamentals Training Course

Systems Engineering Fundamentals is a 1-week training course for engineers, technical project managers and system designers who need to take a complex product from stakeholder needs to a verified design, ending with a traceable requirements and verification matrix. Products now combine hardware, software, connectivity and suppliers, so gaps between disciplines surface late as rework, security flaws or failed acceptance tests. Jupiter Prime Training delivers this course on systems engineering fundamentals.

Why Systems Engineering Fundamentals Matter Now

Three pressures are raising the stakes for engineering teams: software-defined products whose behaviour changes after delivery, supply chains that spread design authority across many firms, and customers who expect security to be designed in rather than patched on. Organisations that apply disciplined life cycle processes catch conflicts in requirements and interfaces while they are cheap to fix; those that skip them discover them during integration, at far greater cost and delay. The course follows the life cycle processes of ISO/IEC/IEEE 15288:2023, the international standard for system life cycle processes, and draws security requirements from the control themes of ISO/IEC 27001:2022.

Who Gains Most from Systems Engineering Fundamentals

- Engineers who elicit stakeholder needs and turn them into verifiable requirements
- Designers who partition functions into subsystems and define interfaces
- Technical project managers who balance scope, schedule and technical risk
- Test engineers who plan verification and validation activities
- Security engineers who embed protective requirements in early design

A multidisciplinary product group spanning mechanical, electrical, software and test specialists gains most, since the course gives them one shared life cycle vocabulary.

Systems Engineering Fundamentals Capabilities

Participants practise moving from vague operational needs to architecture decisions they can trace and test.

- Build a concept of operations and stakeholder needs register — project sponsors and end users
- Specify requirements that are necessary, unambiguous and verifiable — design and test leads
- Design a functional and physical architecture with defined interfaces — subsystem owners
- Weigh design alternatives through structured trade-off studies — technical review boards
-

Apply threat-informed security requirements from concept onwards — information security owners
- Test compliance through a verification and validation strategy — customer acceptance

Key Systems Engineering Fundamentals Terms

- Concept of operations — a description of how users will operate the system in its intended environment, written before design begins.
- Requirement traceability — the linking of each requirement to its source need, design element and verification evidence.
- Interface control document — a record defining how two subsystems exchange data, power, signals or physical loads.
- Verification — confirming through test, analysis, inspection or demonstration that the system meets its specified requirements.
- Validation — confirming that the delivered system satisfies stakeholder needs in its real operating environment.
- Trade-off study — a structured comparison of design alternatives against weighted criteria such as cost, performance and risk.

Systems Engineering Fundamentals Day by Day

Day 1: Life Cycle Thinking and Stakeholder Needs

- ISO/IEC/IEEE 15288:2023 Process Groups — Structuring Technical and Management Activities
- Life Cycle Models — Choosing Sequential, Incremental or Evolutionary Development
- Stakeholder Identification and Mapping — Who Shapes and Accepts the System
- Concept of Operations — Capturing Operational Scenarios Before Design
- Stakeholder Needs Register — Agreeing Priorities and Constraints Early

Day 2: Requirements Engineering

- Requirement Quality Attributes — Writing Necessary, Clear and Verifiable Statements
- Functional and Non-Functional Requirements — Covering Performance, Safety and Reliability
- Security Requirements From Threat Scenarios — Protecting Data and Control Paths
- Requirement Traceability Links — Connecting Needs to Design and Tests
- Requirements Baseline and Change Control — Managing Evolving Customer Demands

Day 3: Architecture and Interfaces

- Functional Decomposition — Allocating Functions to Subsystems and Components
- Physical Architecture Options — Hardware, Software and Supplier Boundaries
- Interface Control Documents — Defining Data, Power and Mechanical Exchanges
- Security Architecture Patterns — Segmentation, Least Privilege and Secure Boot
- Model-Based Systems Engineering — Diagrams That Keep Views Consistent

Day 4: Trade-Offs, Verification and Validation

- Trade-Off Study Method — Weighted Criteria and Sensitivity Checks
- Technical Risk Register — Tracking Design Uncertainty and Mitigations
- Verification Methods — Choosing Test, Analysis, Inspection or Demonstration
- Integration Sequencing — Building Up Subsystems to Expose Interface Faults
- Validation Planning — Proving Fitness in Real Operating Conditions

Day 5: Practising the Requirements and Verification Matrix

- Medical Device Case — Tracing Patient Safety Needs to Design Requirements
- Rail Signalling Case — Trade-Off Between Availability and Security Controls
- Technical Review Simulation — Preliminary and Critical Design Review Gates
- Peer Requirement Inspection — Finding Ambiguity and Missing Verification Methods
- Requirements and Verification Matrix — Linked Needs, Design Elements and Evidence

Applying Systems Engineering Fundamentals After the Course

- Inspect the top-level requirements of a live project against the quality attributes and rewrite any that cannot be verified.
- Run one structured trade-off study on an open design decision and present the weighted results at the next technical review.

Systems Engineering Fundamentals Questions FAQ

What is systems engineering fundamentals and why does it matter now?

It is the core discipline of guiding a complex product from stakeholder needs to a verified system. It matters now because products mix hardware, software and suppliers, so interface and security gaps cause costly late rework.

Which roles gain most from systems engineering fundamentals?

Engineers, technical project managers and system designers gain most. Test and security engineers also benefit, because they join the life cycle at the requirements stage.

How does systems engineering fundamentals differ from a project management course?

This subject focuses on technical processes such as requirements, architecture and verification. A project management course centres on schedule, budget and resources, whereas here you decide what the system must do and how to prove it.

What do you bring back from systems engineering fundamentals?

You bring back a requirements and verification matrix built on a case product. It links each need to design elements and planned evidence, ready to adapt to your own project.

Systems Engineering Fundamentals: The Way Forward

The requirements and verification matrix first informs the scope agreed at the next design review gate. Systems engineers maintain it, test leads plan against it, and project managers use it to judge readiness for integration. After its first use, revisit how requirement changes ripple through the matrix and tighten change control where traceability broke.