BLACKBIRD FOR TRAINING

Systems Engineering for Technology-Based Projects

26 – 30 April 2027
Lyon (France)



Systems Engineering for Technology-Based Projects

Ref: 3255960_357222 **Date:** 26 – 30 April 2027 **Location:** Lyon (France) **Fees:** 5200 GBP

Systems Engineering for Technology-Based Projects

Course Introduction

Technology-based projects often involve complex interactions among hardware, software, data, people, processes, infrastructure, suppliers, and regulatory requirements. Their success depends on more than completing individual technical tasks. Organizations must ensure that all elements work together as one integrated system, meet stakeholder expectations, operate reliably, and remain supportable throughout the intended lifecycle.

Systems engineering provides a structured approach for managing this complexity. It connects business needs, operational objectives, technical requirements, architecture, design, integration, verification, validation, deployment, and lifecycle support. By maintaining a complete view of the system, it helps project teams avoid isolated decisions that may solve one problem while creating difficulties elsewhere.

Many technology projects encounter challenges because requirements are incomplete, interfaces are poorly understood, responsibilities are unclear, or technical decisions are made without considering operational consequences. These issues may result in scope changes, integration failures, schedule delays, cost overruns, security weaknesses, quality problems, and solutions that do not fully satisfy user needs. The risk becomes greater when several disciplines, vendors, technologies, and locations are involved.

The Systems Engineering for Technology-Based Projects course introduces the principles, processes, and practical tools required to plan and deliver integrated technical solutions. It covers stakeholder needs analysis, requirements engineering, functional analysis, system architecture, interface management, technical decision-making, risk control, configuration management, integration, verification, validation, and transition into operation.

Participants will learn how to translate broad organizational needs into clear, measurable, and traceable system requirements. They will examine how requirements flow from the overall system to subsystems and components, and how acceptance criteria can be defined early enough to guide design and testing. This reduces ambiguity and creates a stronger foundation for managing scope, quality, and technical performance.

The course also addresses architecture development and system decomposition. Participants will explore how to define system boundaries, functions, components, interfaces, information flows, and external dependencies. They will learn to compare alternative solutions using structured trade-off analysis and select options that balance performance, cost, risk, security, maintainability, scalability, and lifecycle value.



A significant part of the program focuses on integration and assurance. Technology-based projects frequently experience problems when components that perform successfully in isolation fail when combined. Participants will learn how to plan progressive integration, manage interface risks, define verification and validation activities, and maintain traceability between needs, requirements, design decisions, and test evidence.

The course is relevant to systems engineers, project managers, technical managers, software and hardware engineers, solution architects, business analysts, integration specialists, quality professionals, cybersecurity teams, operations personnel, procurement teams, and decision-makers in public and private organizations. It supports projects involving digital platforms, communication systems, industrial technology, infrastructure, automation, defense, aerospace, transportation, healthcare technology, energy systems, and other complex technical environments.

Course Objectives

By the end of this course, participants will be able to:



- Explain the principles and lifecycle role of systems engineering.
- Define system boundaries, context, stakeholders, and operational needs.
- Translate stakeholder expectations into measurable system requirements.
- Develop clear functional and nonfunctional requirements.
- Establish traceability between needs, requirements, design, and testing.
- Decompose complex systems into manageable functions and components.
- Develop logical and physical system architectures.
- Identify and manage internal and external system interfaces.
- Compare technical alternatives using structured trade-off analysis.
- Integrate risk management into systems engineering activities.
- Coordinate systems engineering with project and technical management.
- Plan system integration, verification, validation, and acceptance.
- Apply configuration and change control to technical baselines.
- Evaluate system performance, quality, reliability, and supportability.
- Manage technical information and engineering decision records.
- Prepare a practical systems engineering plan for a technology project.



Course Modules

Day 1: Systems Engineering Foundations and Project Context

- Definition and purpose of systems engineering.
- Characteristics of complex technology-based projects.
- System, subsystem, component, and enabling system concepts.
- Systems thinking and lifecycle perspectives.
- Relationship with project management.
- Technical and management processes.
- System lifecycle models.
- Stakeholder identification and analysis.
- Operational needs and mission objectives.
- System boundaries and external environment.
- System context development.
- Systems engineering planning.



Day 2: Requirements Engineering and Traceability

- Stakeholder needs elicitation.
- Operational scenarios and use cases.
- Business, user, and system requirements.
- Functional and nonfunctional requirements.
- Performance and quality requirements.
- Security, safety, and regulatory requirements.
- Requirement quality characteristics.
- Requirement prioritization.
- Requirements allocation and flow-down.
- Acceptance criteria development.
- Requirements traceability.
- Requirements review and baseline approval.



Day 3: Functional Analysis and System Architecture

- Functional analysis principles.
- Functional decomposition.
- Inputs, outputs, controls, and constraints.
- Functional flow development.
- Logical architecture.
- Physical architecture.
- System component allocation.
- Architecture viewpoints.
- Technology and platform selection.
- Interface identification.
- Architecture evaluation criteria.
- Alternative solution trade studies.



Day 4: Integration, Risk, and Technical Control

- System integration strategy.
- Incremental and progressive integration.
- Interface management and control.
- Technical risk identification.
- Risk probability and impact assessment.
- Failure modes and dependency analysis.
- Technical performance measures.
- Configuration identification and baselines.
- Engineering change control.
- Technical review processes.
- Decision analysis and documentation.
- Supplier and subcontractor technical coordination.



Day 5: Verification, Validation, and Practical Application

- Verification and validation principles.
- Verification methods and evidence.
- Requirement-based test planning.
- Validation against stakeholder needs.
- System qualification and acceptance.
- Test environments and readiness reviews.
- Defect and nonconformity management.
- Operational transition and deployment.
- System support and lifecycle considerations.
- Practical systems engineering case study.
- Development of a systems engineering plan.
- Technology project improvement roadmap.



Why Should You Attend This Course?

- Develop a structured approach to managing complex technology projects.
- Improve the quality and clarity of technical requirements.
- Reduce design ambiguity and uncontrolled scope changes.
- Strengthen alignment between business needs and technical solutions.
- Identify system dependencies and interfaces before integration.
- Improve communication among technical and nontechnical stakeholders.
- Evaluate design alternatives using transparent decision criteria.
- Reduce integration failures, rework, and late-stage defects.
- Strengthen verification, validation, and acceptance planning.
- Improve technical risk and configuration management.
- Support reliable, scalable, secure, and maintainable solutions.
- Apply systems engineering tools directly to real projects.

Course Conclusion

Systems engineering provides the structure required to manage the technical complexity of modern projects. It ensures that stakeholder needs, requirements, architecture, design, integration, testing, and operational support are treated as connected parts of one lifecycle rather than separate activities.

This course equips participants with practical methods for defining system needs, developing requirements, designing architectures, managing interfaces, evaluating alternatives, controlling technical risks, and planning verification and validation. These capabilities support clearer decisions and reduce the likelihood of costly problems emerging during integration or deployment.



Participants can apply the course outcomes by introducing stronger requirement practices, improving traceability, establishing technical baselines, planning integration earlier, and using formal reviews to assess project readiness. They will also be able to coordinate more effectively with project managers, suppliers, users, engineers, and operational teams.

Over the long term, effective systems engineering improves project predictability, solution quality, lifecycle performance, and organizational learning. It enables institutions to deliver technology-based systems that are not only technically functional, but also aligned with user needs, operationally sustainable, and capable of delivering lasting value.

