Funded by the European Union

Engineering Real-World Competence: Case Studies for Modern IT Education

(DITEC Case Study Book – IT Education Methodology: Engaging with the Business Sector)

January 2026

Table of Contents

INTRODUCTION

The rapid evolution of digital technologies, artificial intelligence, cybersecurity demands, and socio-technical systems is transforming the role of IT professionals. Universities are no longer expected to deliver only technical knowledge; they must prepare engineers who can operate in complex, interdisciplinary, and high-stakes environments where technical decisions carry organizational, societal, and ethical consequences. Despite strong theoretical foundations, many graduates face difficulties when transitioning into professional practice, where communication gaps, system complexity, security risks, and real-world constraints shape everyday engineering work.

The Didactic Technologies for IT Education (DITEC) initiative was created to address this challenge by strengthening the alignment between academic education and industry expectations. Through collaboration between higher education institutions and industry partners in Estonia, Germany, and Cyprus, the consortium works to identify the competencies that modern IT engineers need and to translate these requirements into practical educational approaches. Industry round tables, stakeholder consultations, and feedback from system users provide insights into real workplace challenges, while academic partners transform these insights into innovative teaching and learning methods.

This case study book represents one of the central outcomes of that effort. It provides a collection of realistic, industry-inspired scenarios that reflect the kinds of problems engineers encounter in practice. Rather than focusing solely on technical correctness, the cases emphasize decision-making under uncertainty, interdisciplinary collaboration, communication with non-technical stakeholders, security and safety awareness, system thinking, and professional responsibility. Students are required to analyze complex situations, structure problems, evaluate trade-offs, and propose actionable solutions that balance technical feasibility with organizational and societal needs.

The cases span a broad spectrum of domains, including AI systems, cybersecurity incidents, safety-critical engineering, robotics, software quality, DevOps integration, knowledge management, customer communication, and professional skills development. Each case follows a consistent structure, presenting a real-world challenge, its organizational context, and a problem-solving task designed to foster both technical and transversal competences. The approach supports role-based teamwork, encouraging learners to experience how different professional perspectives—engineering, AI development, management, operations, and stakeholder representation—interact in practical settings.

By working through these cases, students develop more than technical knowledge; they cultivate the ability to think systemically, communicate effectively, manage risks, and act responsibly in complex environments. In this way, the DITEC Case Study Book contributes to narrowing the gap between academic preparation and professional reality, equipping future IT engineers with the competencies needed to succeed in rapidly evolving technological landscapes.

CASE STUDY I: ENSURING RELIABLE LEGAL REASONING IN AN AI DRAFTING ASSISTANT

Description of Company and Challenge

LexoraTech Solutions is a small startup operating in the LegalTech sector, developing an AI assistant designed to help users draft legal documents, interpret regulations, and navigate administrative procedures. While the tool shows strong potential, the team is facing a critical challenge: the AI frequently produces inconsistent legal interpretations, mixes unrelated legal sources, and fails to respect the hierarchy of legal authority. In practice, this means that constitutional rules may be treated at the same level as local guidelines, outdated regulations may appear in responses, or the AI may provide confidently worded yet incorrect interpretations-leading to hallucinations and significant risks for users. The company currently lacks an internal framework for structuring legal sources within the AI system, and needs a solution that combines legal logic, stakeholder awareness, and practical implementation guidance.

What Is Expected From Students?

Students are tasked with proposing a clear and realistic solution that enables an AI legal assistant to consistently apply the hierarchy of legal sources and provide reliable, legally grounded outputs. They should outline how legal authority should be organized-from higher-level constitutional and statutory laws down to regulations, case law, contracts, and internal policies-and describe how the AI can operationalize this structure in practice. The task includes creating a Role Awareness Map that identifies key stakeholders who would guide, validate, or supervise the AI's legal reasoning, as well as developing a generalizable model that organizations or universities can use to teach hierarchical legal interpretation to AI systems. Students must also connect their proposal to real-world practice by illustrating, through examples or workflows, how their approach reduces hallucinations, increases consistency, and improves trust in the AI's support for legal drafting and analysis.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study II: Encouraging Timely Security Collaboration in Software Development

Description of Company and Challenge

FinoraBank Systems is a large enterprise operating in the finance sector, responsible for building and maintaining high-security digital products used by millions of customers. Despite its strong engineering capacity, the company faces a persistent organizational challenge: the security tester is often consulted too late in the development process-or sometimes not at all. This leads to security vulnerabilities being discovered only after major development stages are completed, causing delays, rework, and unnecessary financial risk. Project managers hesitate to involve the security tester early because they fear the process may slow down delivery or create a bottleneck in an already complex workflow. As a result, the organization lacks a balanced approach that encourages timely security involvement without disrupting productivity, creating friction between teams and exposing the company to preventable security threats.

What Is Expected From Students?

Students are tasked with designing practical strategies that encourage project managers and development teams to involve the security tester at the right time, without making the process burdensome or slowing down development cycles. Their proposal should reflect strong communication and collaboration skills, offering concrete methods the security tester can use to build trust, demonstrate value, and integrate seamlessly into project workflows. Students should explore approaches such as role-clarity mapping, communication frameworks, lightweight check-in processes, or collaboration rituals that improve cross-team interaction. They are expected to provide a list of actionable strategies, supported by realistic examples or scenarios, showing how these recommendations can strengthen security practices, reduce misunderstandings, and enhance overall product quality.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study III: Improving Technical Communication Between Product Owners and Developers

Description of Company and Challenge

TransoRoute Digital is a large company operating in the transportation sector, responsible for building complex digital systems that support logistics, route optimization, fleet monitoring, and customer-facing services. Product Owners play a central role: they communicate with clients, decide which features to prioritize, and define what the development teams should build. However, the organization faces a recurring challenge—Product Owners often struggle to translate client requirements into sufficiently detailed technical specifications. As a result, many architectural and implementation decisions fall to developers who make the best choices they can, but without a holistic technical overview. Over time, this leads to inconsistent architecture, reduced system reliability, and a growing gap between business expectations and technical execution. The company seeks a sustainable way to improve communication and collaboration between Product Owners and development teams without requiring Product Owners to become deeply technical experts.

What Is Expected From Students?

Students are tasked with designing a practical solution that improves how technical information flows between Product Owners and developers, ensuring that development teams receive clear, structured, and actionable specifications. They should propose an innovative Agile-inspired workflow, communication model, or role structure that bridges the gap between business requirements and technical implementation. This may involve new collaborative rituals, lightweight documentation frameworks, shared modeling techniques, or even new organizational roles that specialize in translating features into technical domains. Students should demonstrate how their solution enhances collaboration, prevents architectural inconsistencies, and supports long-term system reliability—while avoiding excessive burden on Product Owners. The final output should clearly show how the proposed approach could function in real Agile environments within large transportation-focused organizations.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study IV: Closing the Communication Gap Between Product Owners and Developers in a Mental Health App Startup

Description of Company and Challenge

MindWave Studio is a small startup in the mental health sector developing a mobile application designed to support emotional well-being and provide guided self-help tools. The founder and Product Owner has a clear vision of the app's purpose, functionality, and user experience. However, a persistent communication gap exists between the Product Owner and the developer responsible for implementation. Even when the Product Owner believes the explanation is detailed and sufficient, the developer often responds with "just make a prototype and I will build from there," indicating a lack of clarity, shared understanding, or structured communication. As a result, development progress slows down, misunderstandings accumulate, and the final features often do not fully reflect the original intent. This challenge creates frustration on both sides and affects the quality and pace of app development. MindWave Studio seeks a way to bridge this communication gap to ensure smoother, more collaborative product development.

What Is Expected From Students?

Students are tasked with identifying the underlying causes of the communication gap between the Product Owner and the developer, and proposing methods, tools, or communication frameworks that can improve mutual understanding. They should analyze why explanations may feel clear to the Product Owner but still be unclear for the developer, and suggest practical ways to translate vision into actionable development tasks. Solutions may include structured requirement formats, feature breakdown templates, user story refinements, improved feedback loops, or lightweight visual tools that support shared understanding. Students should demonstrate how their proposed methods enhance collaboration, reduce misunderstandings, and lead to more accurate implementation of mental health app features in real-world startup conditions.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study V: Supporting Conflict Resolution to Restore Team Collaboration in an Automotive Engineering Environment

Description of Company and Challenge

AutoMech Dynamics is a medium-sized company in the automotive sector specializing in software and hardware solutions for next-generation vehicle systems. The teams are technically strong, consisting of engineers, developers, and system architects who work on complex, interdependent automotive technologies. Recently, a serious issue emerged: two team members refused to work together due to escalating personal conflict. Despite their solid technical competence, they were unable to collaborate without tension, which began affecting deadlines, project quality, and team morale. The situation escalated to the point where both individuals requested removal from the project, placing the team manager in a difficult position. The conflict disrupted workflow, strained other team members, and threatened the stability of a critical project. AutoMech Dynamics now seeks a structured, fair, and effective way to address personal conflicts and support healthier team collaboration.

What Is Expected From Students?

Students are tasked with designing an HR-oriented conflict resolution tool or method that helps team members address personal disagreements constructively and resume effective collaboration. They should identify the root causes of interpersonal conflicts in technical project teams and propose practical strategies—such as mediated dialogue frameworks, feedback routines, structured communication processes, conflict-mapping tools, or mini-workshops that promote psychological safety and shared understanding. Students must demonstrate how their proposed approach supports early conflict detection, prevents escalation, and helps the team manager act fairly and consistently. The final solution should clearly show how the method improves collaboration, strengthens team communication, and maintains project performance in real-world automotive engineering environments.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

CASE STUDY VI: DESIGNING AN ONBOARDING PROCESS TO IDENTIFY CANDIDATE FIT WITH COMPANY VALUES

Description of Company and Challenge

ValueCore Industries is a medium-sized organization operating in the technology services sector. The company is strongly committed to building high-performing teams rooted in shared values—collaboration, responsibility, transparency, and a growth-oriented mindset. However, recent hiring experiences revealed a recurring challenge: technically skilled employees often struggle to adapt to the company culture, leading to misalignment within teams, tension in communication, and reduced performance. Although applicants pass technical interviews, the company lacks a structured onboarding or assessment process that can reliably evaluate whether a candidate's personality, working style, and values truly match the organizational environment. ValueCore Industries needs an innovative approach that helps identify whether new hires are likely to integrate well, contribute positively to the team atmosphere, and uphold the company's values from day one.

What Is Expected From Students?

Students are tasked with designing an onboarding and pre-selection process that assesses whether a candidate is a strong cultural and value-based fit for the organization. They should propose a structured, fair, and ethical method that evaluates characteristics such as teamwork style, problem-solving approach, communication preferences, responsibility, and alignment with company values. The solution may include interview frameworks, scenario-based assessments, simulation tasks, collaborative challenges, personality-fit indicators, or digital tools that support value-based screening. Students should demonstrate how their proposed onboarding model helps managers make informed decisions, enhances team compatibility, and improves long-term workplace satisfaction and performance. The final solution must be realistic, applicant-friendly, and applicable in a modern technology-driven organization.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

CASE STUDY VII: HUMAN-CENTERED INNOVATION PILOTS FOR A FUTURE-READY BUSINESS PARK

Description of Company and Challenge

FutureUrban Labs is a development partner working with a large, fast-growing business park that aims to transform itself into a real-world testing ground for social well-being, community building, inclusion, and sustainable behavior. Although the business park is technologically advanced and already supports pilots in mobility, innovation, and green solutions, several human-centered challenges remain unsolved. Many knowledge workers experience social isolation despite working in a vibrant environment; underused indoor and outdoor spaces remain inactive; newcomers, international workers, older adults, and people with different abilities often struggle to feel included; and companies want to contribute to climate action but lack simple guidance on where to begin. The business park wants to shift from abstract programs to small, practical, low-tech interventions that can be tested quickly with real people and generate actionable learning. The goal is to design meaningful pilots that support healthier daily habits, build community, enhance inclusion, activate unused spaces, or accelerate climate-positive behavior among companies and employees.

What Is Expected From Students?

Students are tasked with designing a small-scale, people-centered pilot that can be implemented and observed directly inside the business park. They should select one real issue—such as daily well-being, reducing social isolation, creating inclusive participation for a specific community, activating unused spaces, or helping companies take their first steps toward climate action—and propose a practical, testable solution supported by a clear hypothesis. The intervention may take the form of a pop-up activity, a behavioral nudge, a simple service, a participatory process, a new community ritual, a storytelling project, or a lightweight toolkit for sustainability. Students must explain how the pilot would be implemented, which user groups it targets, how success could be measured, and how it would create positive impact in daily life within the business park. The final solution should be realistic, low-tech, easy to test, and grounded in real human needs.

Note: The organization name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study VIII: Raising Software Quality Competence of Junior Developers in the Age of AI-Generated Code

Description of Company and Challenge

CodeForge Technologies is a mid-sized software development company operating in the digital solutions and product engineering sector. The company employs many junior developers who recently graduated from computer science programs. While these graduates possess solid theoretical knowledge and are comfortable using modern development tools, the company has observed a critical trend: the software quality delivered by junior developers is often comparable to code produced by AI coding assistants based on large language models. Although AI systems can generate functional code quickly, they frequently introduce hidden errors, security vulnerabilities, inefficient logic, and reliability issues. Senior developers are typically able to detect these problems due to their experience in debugging, architectural thinking, and risk awareness, whereas junior developers often lack the advanced bug-finding competence and software quality evaluation skills required to critically assess both human-written and AI-generated code. As AI-assisted coding becomes standard practice, the company faces a new challenge: junior developers increasingly rely on AI-generated code but are not yet capable of identifying subtle defects, ensuring security robustness, or validating long-term code reliability. This creates a quality gap between junior and senior developers and increases the risk of deploying unstable or insecure software. CodeForge Technologies therefore seeks a structured approach to help junior developers elevate their competence in software quality assessment and error detection, particularly in the context of AI-generated software artifacts.

What Is Expected From Students?

Students are tasked with designing a practical and educational strategy that helps IT graduates improve their competence in debugging, software quality evaluation, and the safe use of AI coding tools. Their solution should focus on how junior developers can develop skills that allow them to approach the quality assurance capabilities of senior engineers. They should propose a structured learning and evaluation model in which AI and large language models are used as development assistants while emphasizing critical evaluation rather than blind trust. The proposal should include the design of case-based exercises where learners analyze AI-generated code to identify bugs, inefficiencies, security risks, and reliability issues, along with the development of clear evaluation criteria for assessing code quality, such as correctness, maintainability, performance, security, and robustness. Students should also describe methods for strengthening bug-finding competence through systematic debugging workflows, failure scenario analysis, and test-driven reasoning, as well as approaches for teaching secure coding awareness and risk identification in AI-assisted development environments. Finally, they must explain how their model helps transform junior developers from passive users of AI tools into active and critical reviewers of AI-generated software, and illustrate, through examples or practical workflows, how their approach reduces hidden defects, improves software security, and increases overall code reliability in real-world development contexts.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

CASE STUDY IX: DEVELOPING SAFETY AWARENESS IN SOFTWARE ENGINEERING FOR LIFE-CRITICAL SYSTEMS

Description of Company and Challenge

MediControl Systems is a large medical technology company responsible for developing both the hardware and software of advanced infusion pumps used in hospitals. The system relies heavily on precise software control to ensure accurate medication dosing. Even minor deviations in dosage can have life-threatening consequences for patients. Although the company employs highly capable software engineers, a significant challenge has emerged within large development teams: many junior developers lack a clear awareness of how their individual software contributions can directly affect patient safety. The consequences of software errors are not always obvious. Problems may arise not only from incorrect data processing, which represents functional failure, but also from timing issues, delays, or performance instability, which are non-functional aspects that can equally result in dangerous outcomes. Because software modules are often developed in isolation, developers may not fully understand the broader cause-and-effect relationships between code behavior and real-world medical impact. MediControl Systems therefore seeks a way to strengthen safety culture among engineers and ensure that developers understand the direct connection between software quality, system behavior, and risks to human life.

What Is Expected From Students?

Students are tasked with actively engaging in a structured safety and risk analysis process that helps them understand the real-world consequences of software failures in safety-critical systems. They should apply methods such as Failure Mode and Effects Analysis (FMEA) to identify how different types of software faults, including functional errors and timing-related issues, can propagate through the system and affect patient safety. The task requires students to map cause-and-effect chains, identify potential failure modes, analyze their possible impacts, and evaluate the severity of outcomes. They must systematically document risks, including sources of failure, system-level consequences, and possible mitigation strategies. Through this process, students should develop a deeper awareness of safety responsibility, recognize how individual software decisions influence overall system reliability, and understand the importance of quality, testing, and risk prevention in medical and other life-critical technologies. The final outcome should be a structured and well-documented risk list demonstrating clear reasoning about failure scenarios and their safety implications.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study X: Requirements Engineering for a Humanoid Care Robot

Description of Company and Challenge

CareBot Solutions develops humanoid robots for use in elderly care facilities. Humanoid robotics is a new technological field, and both care staff and private users have limited experience working with such systems. The robot is a socio-technical product, with roughly half of its functionality determined by software, operating strictly according to programmed rules and decision models. A key challenge is not only technical implementation, but identifying real operational needs in care settings and translating them into safe and feasible automation solutions. Automation does not necessarily mean imitating human behavior; often, alternative technical approaches are more efficient and safer.

For example, a care home requests that the robot tidy a common room twice per day. This request is passed to the software and AI team as a general goal, but it must first be broken down into detailed processes, technical constraints, hardware limits, safety rules, and interaction requirements, especially because the robot works near elderly residents. An interdisciplinary team, including robotics software engineers, AI engineers, and a product manager representing user needs, must convert this vague request into a structured technical specification.

What Is Expected From Students?

Students are tasked with designing a requirements engineering approach that translates the task “tidying up” into a safe and autonomous robotic solution. Working in teams of three—covering AI engineering, software architecture, and technical product management—they must decompose the task into sub-problems such as perception, object recognition, manipulation, navigation, and human-aware safety. They should define what capabilities the robot needs, how it interprets its environment, and how it avoids safety risks while operating around people.

The final result should be a structured technical specification describing how software and AI components enable the robot to perform the task. Students must show how user needs are transformed into technical requirements, how the task is broken into implementable modules, and how safety is integrated across all solution elements.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

CASE STUDY XI: AI-SUPPORTED ONBOARDING FOR DEVELOPERS WORKING WITH LEGACY CODEBASES

Description of Company and Challenge

SoftBridge Systems is a large software company maintaining long-standing, complex codebases that have evolved over many years. The systems include intertwined modules, legacy components, custom data models, and infrastructure dependencies that are only partially documented. A recurring organizational challenge is that newly hired developers require several months before they fully understand the existing architecture and are able to contribute productively. During this period, they struggle to navigate the code, identify relevant components, understand design decisions, and locate colleagues with historical knowledge of specific modules. As a result, onboarding is slow, productivity is delayed, and knowledge remains concentrated among a small group of experienced engineers.

The company is exploring how artificial intelligence can support knowledge management and accelerate code understanding. AI-based tools could help summarize large code sections, visualize system structures, highlight invariants and architectural constraints, and assist developers in finding relevant documentation or knowledgeable team members. The challenge is to design a structured approach that combines AI-supported code representation, documentation practices, and organizational knowledge-sharing methods to reduce onboarding time while improving developers' independent understanding of complex data models, infrastructure setups, and deployment processes.

What Is Expected From Students?

Students are tasked with designing a knowledge management and onboarding framework that uses AI to support developers in understanding large and complex legacy codebases. They should propose how AI-generated summaries, structured code documentation, and visual representations of code structure and system invariants can help newcomers build a mental model of the system more quickly. Their solution should also include methods for supporting targeted search within the codebase, enabling developers to efficiently locate relevant code sections, related components, and historical design decisions.

In addition, students should address how AI tools can help identify subject-matter experts within the organization by linking code areas to developers who previously worked on them. The framework should support the independent understanding of complex data models, system infrastructure, and deployment pipelines. The final outcome should be a coherent model that combines technical tools and organizational practices to accelerate onboarding, distribute knowledge more effectively, and strengthen long-term maintainability of large software systems.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study XII: Integrating Operational Requirements Early into the Development Process

Description of Company and Challenge

SysCore Technologies is a large organization developing complex software systems that must operate reliably in long-term production environments. Within the company, software development and system operations are organized as separate units, often with different responsibilities, performance indicators, and priorities. Development teams focus on delivering new features, functionality, and innovation, while operations teams prioritize system stability, maintainability, performance, and risk reduction. This organizational separation frequently leads to misunderstandings, delays, and solutions that are technically functional but difficult to operate, maintain, or scale in real-world environments.

A central challenge is the lack of early integration of operational requirements into the development lifecycle. Important aspects such as deployment constraints, monitoring needs, hardware dependencies, update procedures, and long-term maintenance considerations are often addressed too late. As a result, systems may require costly rework, suffer from performance issues, or create operational risks. SysCore Technologies therefore seeks ways to improve mutual understanding between development and operations and to ensure that operational perspectives are embedded from the beginning of the engineering process.

What Is Expected From Students?

Students are tasked with designing a process model that integrates the requirements of both development and operations in a structured and collaborative way. They should analyze how the needs of the two domains differ and identify the key interfaces where coordination is essential. Their solution should address how operational requirements—such as system stability, monitoring, hardware selection, deployment conditions, update strategies, and long-term maintainability—can be incorporated early into design and implementation activities.

Students must also clarify what each side needs to understand about the other, including how development workflows function and what operational constraints influence system performance. The proposal should outline communication mechanisms, shared artifacts, and collaboration practices that support a continuous flow of information from programming and testing through hardware selection, technical setup, system evolution, and live operation. The final result should demonstrate a holistic view of the software lifecycle, highlighting how improved communication and cooperation reduce risks, shorten delays, and lead to more complete and sustainable solutions.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

CASE STUDY XIII: DEVELOPING SELF-ORGANIZATION SKILLS IN STRUCTURED PROJECT ENVIRONMENTS

Description of Company and Challenge

ProActive Systems is a technology company where project processes, milestones, and overall workflows are clearly defined at the organizational level. Despite this structured environment, a recurring challenge arises among junior project members: they often struggle to organize their own work effectively within the broader process. While expectations, deadlines, and deliverables are specified, young professionals frequently lack the skills to break down their responsibilities into manageable tasks, estimate effort realistically, and monitor their own progress. As a result, delays, last-minute stress, and incomplete outputs occur—not due to missing technical competence, but because of insufficient self-organization and planning ability.

The company is exploring frameworks such as the Skills Framework for the Information Age (SFIA) to provide orientation regarding professional responsibilities and competency levels. However, individuals still need practical methods to translate abstract role descriptions into daily planning and structured execution. The challenge is to help project participants develop self-management strategies that allow them to structure their tasks, anticipate risks, and recognize early when objectives may not be met.

What Is Expected From Students?

Students are tasked with designing a self-organization approach that helps individuals break down their assigned goals into concrete, manageable work packages and realistic timelines. They should propose methods for task decomposition, prioritization, and progress tracking that allow early identification of potential non-fulfillment of objectives. The solution should consider how external factors—such as the availability of colleagues, dependencies on other tasks, and personal skill development—affect planning and execution.

Students should also reflect on how structured self-management contributes to professional growth, reliability, and better collaboration within project teams. The final outcome should present a practical model that enables individuals to plan their work transparently, monitor their own progress, and adapt proactively to changing conditions while remaining aligned with the overall project process.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

CASE STUDY XIV: AI-SUPPORTED SOC UNDER ATTACK – DECISION-MAKING, PROTECTION, AND COMMUNICATION UNDER TIME PRESSURE

Description of Company and Challenge

CyberShield Technologies is a mid-sized technology company operating a Security Operations Center (SOC) that is increasingly supported by an AI-driven analytics platform. This platform uses machine learning and generative AI models to prioritize security alerts, enrich incidents with contextual information, and provide response recommendations to human analysts. During a regular operational day, a simulated security incident occurs: multiple critical alerts indicate a possible attack targeting the AI platform itself. Indicators include potential manipulation of training data, suspicious API usage, and unexplained model decisions. At the same time, the volume of incidents rises sharply, putting additional strain on the SOC team.

Management expects a rapid, well-founded decision: whether the AI system should continue to operate, be restricted, or be temporarily shut down. Meanwhile, operational security, regulatory compliance, and customer communication must remain intact. The situation is characterized by incomplete information, high time pressure, and conflicting priorities between maintaining operational efficiency and ensuring system integrity. The organization must stabilize SOC operations while preserving the ability to make reliable decisions, even as the trustworthiness of the AI support system is in question.

What Is Expected From Students?

Students are tasked with designing a structured response approach that maintains SOC stability during a security-relevant incident affecting AI components. They must address how to safeguard the integrity of AI systems while ensuring continued decision-making capability under time pressure. Their work should include a concise decision document that presents a situation overview, at least two possible courses of action, and an evaluation of risks and impacts leading to a clearly justified decision. In addition, students should develop a high-level architectural sketch of the SOC, illustrating AI components, security mechanisms such as isolation and monitoring, and the integration of human-in-the-loop controls. Finally, they should prepare a short management statement that communicates the situation, chosen approach, and next steps in a clear and decision-oriented manner, focusing on safety, stability, and operational continuity rather than technical details. The overall solution should demonstrate structured crisis handling, balanced risk assessment, and effective communication between technical and management levels.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

CASE STUDY XV: CYBERSECURITY IN SMART CITY INFRASTRUCTURE – STREET LIGHTING SYSTEMS

Description of Company and Challenge

UrbanGrid Solutions is a technology provider responsible for implementing intelligent street lighting systems as part of a smart city infrastructure project. The system connects streetlights through a digital network, enabling remote monitoring, automated control, energy optimization, and integration with other city services. While the focus of the project has largely been on functionality, efficiency, and sustainability, a significant challenge has emerged: insufficient awareness of cybersecurity risks during the design and implementation phases.

Street lighting may appear to be a low-risk application, but as a connected infrastructure component, it can serve as an entry point into broader municipal networks. Vulnerabilities in communication channels, device firmware, cloud interfaces, or management platforms could allow attackers to manipulate lighting behavior, disrupt city operations, extract data, or pivot into other critical systems. The organization must therefore reconsider the system not only as an energy solution but as part of a larger cyber-physical ecosystem where security weaknesses may have cascading consequences.

What Is Expected From Students?

Students are tasked with conducting a structured cybersecurity analysis of the given street lighting application context. They should identify potential security risks across the system, including network communication, device-level vulnerabilities, centralized control mechanisms, and interactions with other urban infrastructure. Their work should highlight possible attack vectors, threat scenarios, and critical paths where security failures could lead to significant operational or safety impacts.

In addition, students must propose measures to mitigate these risks by outlining a structured security architecture. This should include principles such as segmentation, authentication, secure communication, monitoring, and update management. The final outcome should demonstrate a clear understanding of how cybersecurity must be integrated into system design, identify particularly critical system paths, and provide practical recommendations for securing connected urban infrastructure.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study XVI: Production Shutdown After a Ransomware Attack – Balancing Security, Time Pressure, and Business Continuity

Description of Company and Challenge

OptiCore Manufacturing is a highly specialized production company supplying optical lenses for the semiconductor industry. Its production environment is heavily automated and depends on the tight integration of IT systems, such as ERP, manufacturing execution systems, and quality management platforms, with operational technology (OT) components including production machinery and control systems. In the early morning hours, a large-scale ransomware attack is detected. Several critical systems are encrypted, production lines have stopped, and delivery deadlines for international key customers are at risk. Initial indications suggest that the compromise may have existed undetected for weeks.

Management is under severe pressure. Every hour of production downtime leads to significant financial losses and potential contractual penalties. At the same time, it is unclear whether backups are complete, clean, and quickly restorable. The organization must also consider external communication, regulatory obligations, and the risk that restoring systems too quickly could reintroduce compromised components. The central challenge is to make structured, high-impact decisions under extreme time pressure, balancing operational recovery with long-term security and risk containment.

What Is Expected From Students?

Students are tasked with designing a controlled and secure recovery strategy following a successful ransomware attack. They must address how production can be restored in a way that minimizes economic damage while preventing further security risks. Their work should include a concise decision document describing both the technical and business situation, prioritized courses of action such as restore strategies, and a risk analysis covering IT, OT, and business impacts. The decision must be clearly justified and based on explicit assumptions.

In addition, students should develop a high-level architectural sketch of the target environment after recovery. This should illustrate the separation of IT and OT domains, backup and recovery zones, and the integration of monitoring and security controls to reduce future risk. Finally, students must prepare a short management statement that clearly explains the situation, the chosen approach, associated risks, and expected timelines in language understandable to non-technical stakeholders. The overall solution should demonstrate structured crisis decision-making, alignment between technical and business priorities, and an emphasis on secure, resilient system restoration.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study XVII: Security by Design for Networked Embedded Systems

Description of Company and Challenge

EmbedSys Engineering is a company developing networked embedded systems used in industrial and technical environments. Traditionally, system development has focused on functionality, performance, and cost efficiency, while security aspects have often been addressed only at later stages or after deployment. This reactive approach has repeatedly led to situations where systems must be retrofitted with security mechanisms, which is technically difficult, costly, and sometimes only partially effective. In some cases, architectural decisions made early in the design phase limit the feasibility of implementing strong security controls later on.

As embedded systems become increasingly connected and integrated into larger digital ecosystems, the absence of security awareness during early development stages poses significant risks. Vulnerabilities at device, firmware, communication, or interface levels can expose entire systems to cyber threats. EmbedSys Engineering therefore seeks a systematic way to embed security considerations throughout the entire development lifecycle rather than treating security as an add-on.

What Is Expected From Students?

Students are tasked with designing a development approach that integrates security considerations continuously across the full engineering process. They should analyze how security requirements can be introduced early during system conception and carried through design, implementation, testing, and maintenance. Their proposal should address both classical and agile development models, identifying where and how security-related activities, roles, and checkpoints can be embedded.

Students should define what kinds of security requirements must be considered, how threat awareness and risk analysis can influence architectural decisions, and how development practices can ensure that security is not overlooked under time or cost pressure. The final outcome should be a structured model that demonstrates how “Security by Design” can be operationalized, reducing the need for costly retrofitting and improving the resilience of networked embedded systems from the outset.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study XVIII: Building Customer Understanding During Product Development

Description of Company and Challenge

InnovaTech Solutions develops technical products for diverse customer groups with varying professional backgrounds. A recurring challenge during product development is that customers often describe their needs in terms of problems rather than clear technical requirements. Their expectations may be incomplete, unclear, or based on assumptions that differ significantly from the technical perspective of project developers. At the same time, developers tend to interpret requests through their own technical lens, which can lead to misunderstandings, incorrect solutions, or products that do not fully address the real user need.

Customers and end users typically do not share the same technical vocabulary as development teams. Specialized terminology, system constraints, and implementation logic are often not well understood outside the engineering context. As a result, communication gaps arise that can affect project scope, timelines, and final product usability. InnovaTech Solutions therefore aims to strengthen developers’ ability to understand customer perspectives, question initial requirements constructively, and anticipate potential misunderstandings early in the development process.

What Is Expected From Students?

Students are tasked with designing a communication and collaboration approach that helps development teams build a deeper understanding of customer needs during product development. They should propose methods for active customer inquiry, structured feedback collection, and systematic clarification of requirements. Their solution should focus on developing problem awareness, enabling developers to interpret customer statements critically and to translate them into technically meaningful requirements while considering the user perspective.

Students should also outline a communication concept that supports dialogue between technical and non-technical stakeholders, including strategies for managing terminology gaps and expectation mismatches. Working in small groups with different roles, students should practice how collaborative questioning, reflection, and role-based interaction can improve mutual understanding. The final outcome should demonstrate how structured communication practices reduce misunderstandings, support better requirement definition, and lead to more user-centered product solutions.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study XIX: Strengthening Presentation and Communication Skills for Non-Technical Audiences

Description of Company and Challenge

CodeBridge Learning is a small organization operating in the information technology education sector. The company delivers introductory programming seminars and workshops to a wide range of audiences, including school students, adult learners, and professionals from non-technical backgrounds. While the organization's instructors and facilitators possess strong technical knowledge, a recurring challenge has emerged in their ability to communicate this knowledge effectively to non-technical audiences of different ages and backgrounds.

Specifically, instructors often struggle to simplify complex programming concepts, adapt their language to varied audience needs, and present information in a clear, engaging, and accessible way. As a result, participants may feel confused or disengaged, limiting the overall impact of the training sessions. This challenge highlights a gap not in technical expertise, but in presentation, communication, and user-centered explanation skills, which are critical for knowledge transfer in educational and outreach contexts.

What Is Expected From Students?

Students are tasked with designing a practical approach to strengthening presentation and communication skills for technically skilled individuals who must engage with non-technical audiences. They should propose a user-centered training or development framework that helps instructors explain technical concepts in simple terms, tailor messaging to different age groups, and improve audience engagement. Solutions may include presentation skill toolkits, instructional design methods, storytelling techniques, or interactive educational materials. Students should demonstrate how their proposal enhances clarity, accessibility, and learning outcomes across diverse target groups.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

CASE STUDY XX: BRIDGING THE AGILE READINESS GAP OF NEW IT GRADUATES

Description of Company and Challenge

AgileWorks Solutions is a medium-sized company operating in the software development sector. The organization frequently hires newly graduated IT professionals who demonstrate strong theoretical and technical knowledge. However, despite their academic preparation, many new hires struggle to apply these skills effectively within real-world agile development environments.

Challenges commonly arise in areas such as sprint-based work, cross-functional collaboration, prioritization, iterative delivery, and adapting to fast-paced team dynamics. New graduates often find it difficult to translate individual technical tasks into collective team outcomes, leading to delays, misalignment, and reduced productivity during onboarding periods. This gap between academic knowledge and agile practice affects both team performance and the confidence of new employees as they integrate into development teams.

What Is Expected From Students?

Students are asked to design a realistic solution that helps new IT graduates apply their technical knowledge more effectively within agile projects. Their proposal should focus on improving adaptability, critical thinking, and practical collaboration skills. Suggested outputs may include agile training modules, simulated sprint projects, onboarding playbooks, or experiential learning tools that mirror real development environments. Students should clearly demonstrate how their solution supports smoother onboarding, stronger team contribution, and improved delivery outcomes in agile settings.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

CASE STUDY XXI: ENHANCING CYBERSECURITY RISK AWARENESS AND STAKEHOLDER COMMUNICATION

Description of Company and Challenge

SecureData Tech is a small company specializing in cybersecurity and sensitive data management solutions. The organization frequently collaborates with newly graduated IT professionals and interns who possess foundational technical knowledge in system security and data protection. However, a recurring challenge lies in their limited understanding of the broader business, regulatory, and risk-management context surrounding cybersecurity decisions.

While new professionals can identify technical vulnerabilities, they often struggle to assess risks holistically, prioritize threats, or communicate security-related decisions to non-technical stakeholders. This gap can lead to misaligned priorities, insufficient compliance awareness, and communication breakdowns between technical teams, management, and external partners. Strengthening business awareness and stakeholder-oriented thinking is essential to improving overall security practices.

What Is Expected From Students?

Students are tasked with proposing methods to enhance the ability of new cybersecurity professionals to evaluate risks and communicate informed decisions. They should design simple yet practical frameworks for cybersecurity risk analysis and incident response, taking into account business impact, regulatory considerations, and stakeholder needs. Expected deliverables may include basic risk assessment models, incident management scenarios, or communication guidelines that bridge technical and non-technical perspectives.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study XXII: Upskilling New Hires in Technical Writing and Project Management

Description of Company and Challenge

NextStep Consulting is a small technology consulting firm operating in a highly competitive and rapidly evolving market. Due to a limited hiring pool and market distortions, the company often recruits new hires with varied educational backgrounds and limited professional experience. Although technically capable, many new employees struggle with two critical areas: concise technical writing and effective project and time management.

These challenges affect documentation quality, internal communication, and the ability to meet project deadlines in fast-paced consulting environments. New hires often find it difficult to structure written outputs clearly, manage competing priorities, and adapt to the intensity of project-based work. This results in increased supervision needs, inefficiencies, and slower integration into consulting teams.

What Is Expected From Students?

Students are asked to design a structured upskilling approach that supports new hires in developing succinct technical writing skills and stronger project and time management capabilities. Their proposal should emphasize communication, collaboration, and decision-making skills, and may include formal training modules, early involvement in research or applied projects, internship-based learning models, or integrated onboarding programs. Students should demonstrate how their solution helps new hires adapt more quickly and contribute effectively to consulting projects.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study XXIII: Building a Culture of Ownership and End-to-End Responsibility

Description of Company and Challenge

EndVision Systems is a small company developing ICT and web-based decision support systems. While employees generally perform their assigned tasks competently, the organization faces a persistent challenge related to ownership and end-to-end responsibility. Tasks are often completed in isolation, without clear accountability for the overall outcome, delivery timelines, or the resolution of unexpected obstacles.

This lack of ownership leads to increased micromanagement, implementation delays, and reduced team effectiveness. Employees may complete individual responsibilities but hesitate to take initiative beyond their immediate scope. The situation highlights a need to strengthen soft skills such as responsibility, initiative, holistic thinking, and outcome orientation, enabling teams to operate more autonomously and with greater focus on results.

What Is Expected From Students?

Students are tasked with designing a strategic approach to strengthening ownership and end-to-end responsibility within teams. They should propose methods that enhance autonomy, accountability, and decision-making, drawing on design thinking, project management, and stakeholder awareness principles. Expected outputs include a strategic recommendation report outlining practical interventions, cultural practices, or skill-development initiatives that help employees take greater responsibility for outcomes and improve the overall quality of deliverables.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study XXIV: Aligning Technical Talent With Business Awareness and Stakeholder Understanding

Description of Company and Challenge

DataSense Innovations is a small technology company delivering analytical and decision-support solutions to external clients. While its technical teams are highly skilled, management has observed that employees often lack business awareness and stakeholder sensitivity. Developers and analysts may optimize technical solutions without fully understanding client priorities, constraints, or organizational dynamics.

This disconnect results in solutions that are technically sound but misaligned with user needs or strategic goals. The risk is reduced client satisfaction and inefficient use of resources. The company seeks to strengthen employees' ability to think beyond technical execution and consider the broader business and stakeholder context of their work.

What Is Expected From Students?

Students are asked to design learning or organizational mechanisms that improve business awareness and stakeholder understanding among technical professionals. They should propose methods that help employees interpret client needs, balance trade-offs, and communicate effectively with non-technical stakeholders. The solution should demonstrate how improved awareness leads to better decision-making and stronger client relationships.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study XXV: Designing Practical Learning Deliverables for Student–Industry Collaboration

Description of Company and Challenge

InnovateBridge Network is a small organization coordinating collaborations between companies and students through project-based learning initiatives. While industry partners value fresh perspectives, they often find that student outputs lack structure, practical relevance, or clear applicability to real organizational challenges.

The issue is not creativity, but alignment: students may produce abstract concepts or overly theoretical analyses that are difficult to implement. For InnovateBridge Network, the challenge is to guide students toward producing deliverables that balance analytical depth with real-world usefulness, ensuring value for both learners and partner organizations.

What Is Expected From Students?

Students are expected to reflect on what makes a professional, actionable deliverable and design output formats that meet industry expectations. They should propose guidelines, templates, or evaluation criteria that help translate ideas into usable artifacts. The solution should demonstrate how clearer expectations and structured outputs improve collaboration outcomes and learning effectiveness.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study XXVI: Too Many Security Findings – A Pragmatic Approach to an Overloaded Vulnerability List

Description of Company and Challenge

SecureSoft Solutions is a large software organization maintaining a historically grown and complex codebase that has evolved over many years. The system includes legacy components, custom-developed modules, and numerous third-party dependencies. As part of an internal improvement initiative, the development team activates a GitLab-based security scanning tool to gain better visibility into potential vulnerabilities. The result is overwhelming: hundreds of security findings appear almost immediately, ranging from known CVEs in external libraries to potential weaknesses in proprietary code and a significant number of unclear or suspected false positives.

At the same time, the organizational context is highly constrained. There is no dedicated security team, no additional budget, and no spare capacity for large-scale remediation efforts. Development resources are already fully allocated to feature delivery and maintenance tasks. Despite these limitations, management expects measurable risk reduction and accountable decision-making. The team faces a critical challenge: addressing security responsibly without attempting to fix everything at once, avoiding disruption of operations, and making defensible prioritization decisions under uncertainty.

This situation reflects a common real-world dilemma in software engineering, where security tooling reveals more issues than an organization can realistically handle, forcing teams to balance technical risk, business impact, and available resources.

What Is Expected From Students?

Students are tasked with designing a pragmatic, risk-based approach for managing a large volume of security findings under severe resource constraints. They must propose a structured method for organizing and prioritizing vulnerabilities in a way that leads to meaningful security improvement rather than superficial activity. Their solution should explain how findings can be categorized, for example by distinguishing between third-party dependency issues, proprietary code vulnerabilities, and likely false positives.

Students should define a prioritization strategy that goes beyond raw severity scores and incorporates contextual factors such as system exposure, exploitability, business impact, and operational risk. They are expected to identify “quick wins” that can significantly reduce risk with minimal effort and to propose lightweight processes that help prevent future floods of unmanageable findings. The solution should be realistic for small or resource-constrained teams and compatible with ongoing development work.

As a reflection exercise, students should assume they are part of the development team facing 500 findings with the capacity to address only 10. They must select which findings to handle, justify their prioritization method, explicitly state which risks they consciously accept, and explain how they would communicate and defend these decisions to management. The final outcome should demonstrate clear risk awareness, structured reasoning, and the ability to align technical decisions with organizational realities.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

CASE STUDY XXVII: AI-ASSISTED REQUIREMENTS ENGINEERING UNDER STRICT REGULATORY CONDITIONS

Description of Company and Challenge

RegulaSys Engineering is a safety-critical company preparing the development of a new generation of systems subject to strict regulatory oversight. The organization operates in an environment where precise and complete requirements documentation is mandatory due to safety, compliance, and certification obligations. However, many requirements from previous system generations were never fully captured or consistently documented. Instead, they exist across fragmented sources such as legacy documents, emails, meeting notes, and the tacit knowledge of senior experts.

At the same time, the company faces significant constraints. There is limited time, a shortage of experienced requirements engineers, and high pressure to progress quickly with the new system generation. To address the backlog, the organization considers the use of artificial intelligence to support requirements engineering tasks. Due to data protection laws, export control restrictions, and critical infrastructure regulations, only regulated AI tools operating entirely within the company's secure enterprise environment are permitted. Public or cloud-based AI services are strictly prohibited. The challenge is to determine how AI can be used responsibly and effectively to support requirements identification, clarification, and documentation, while ensuring full compliance and recognizing that ultimate responsibility remains with human experts.

What Is Expected From Students?

Students are tasked with designing an approach for applying regulated AI tools to support requirements engineering in a safety-critical and highly constrained environment. They should propose how AI can be used to analyze and organize unstructured internal materials, extract implicit domain knowledge from secure data sources, and assist in reformulating requirements in a precise and consistent manner. The solution should address how AI can help identify ambiguities, inconsistencies, missing information, and common errors in existing requirements without compromising regulatory compliance.

Students must also explain how strict constraints—such as data protection, safety certification requirements, and export control—shape the permissible use of AI tools. Their proposal should clearly distinguish between tasks that can be supported by AI and those that must remain under direct human responsibility, particularly where judgment, accountability, and safety assurance are required. As a reflection exercise, students should imagine working with an unsorted collection of legacy requirement materials and outline three concrete AI-assisted steps they would take within a secure enterprise environment, while identifying at least one stage where human expertise is indispensable. The final outcome should demonstrate a balanced, compliant, and realistic use of AI as a support tool for high-quality requirements engineering in safety-critical systems.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study XXVIII: Between Legacy and Modernity – Becoming Productive in a Long-Established Software Landscape

Description of Company and Challenge

HeritageSoft Systems is a company maintaining a long-established software landscape that has evolved continuously over several decades. Large parts of the codebase were written 10 to 20 years ago and have proven to be highly reliable, operating in safety-critical or high-availability environments. The systems are business-critical, stable, and subject to strict internal standards and operational constraints. Development processes are well established, though not always aligned with modern agile practices or contemporary technology trends.

A newly graduated software developer joins the organization, bringing up-to-date academic knowledge of modern programming paradigms, web-based user interfaces, architectural patterns, and AI-assisted development tools. While motivated to contribute and improve the system, the newcomer faces a dual challenge: becoming productive quickly in a complex legacy environment while respecting the deep domain knowledge embedded in the existing system and the experience of senior developers. At the same time, there is interest in evolving parts of the system over time—for example, introducing web-based user interfaces alongside or instead of traditional native UI frameworks such as Qt—without destabilizing the system or violating internal standards. The core challenge is to bridge two worlds: preserving the strengths of a mature legacy system while enabling controlled, sustainable modernization.

What Is Expected From Students?

Students are tasked with designing a pragmatic approach that enables newcomers to become productive in a long-established software landscape while supporting gradual system evolution. They should focus on how domain knowledge can be systematically acquired from experienced colleagues through methods such as pair programming, shadowing, architecture walkthroughs, targeted interviews, or AI-assisted code analysis. The solution should emphasize knowledge management as a central success factor, recognizing that much critical understanding is implicit and not fully documented.

In addition, students should propose how modern UI technologies, such as web-based interfaces, can be introduced incrementally alongside existing native frameworks. This includes defining architectural principles that decouple user interfaces from business logic, for example through API layers, modular services, or presenter/view-model patterns, allowing different UI technologies to coexist or migrate gradually. Students should also reflect on how modern development practices and agile elements can be integrated respectfully into established workflows, and how discussions around coding standards and modern techniques can be conducted constructively without causing friction. As a reflection exercise, students should formulate questions for senior developers to elicit domain knowledge, outline a high-level architecture supporting parallel UI technologies, and describe how they would raise modernization topics in a professional and respectful manner. The final outcome should demonstrate balanced reasoning, strong communication skills, and an understanding of how technical evolution must align with organizational context and long-term system stability.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

Case Study XXIX: From Studies to Structured Software Development – Embedding Security Requirements in Practice

Description of Company and Challenge

StructuraSoft Solutions is a medium-sized software company developing complex, long-lived systems within a structured organizational environment. Development follows defined workflows, architectural guidelines, change-management processes, and clearly assigned roles. Shortly after joining the company, a newly graduated software engineer identifies several security weaknesses in the existing system. From an academic perspective, these issues appear obvious and solvable through well-known best practices. However, practical implementation quickly proves far more complex.

Security improvements cannot be implemented ad hoc. Development capacity is limited, colleagues have competing priorities, and responsibilities for security are distributed across multiple roles, including development teams, architects, process owners, and management. Changes must align with existing policies, release cycles, and architectural decisions. Without integration into official processes and governance structures, security improvements remain isolated and unsustainable. The core challenge is not recognizing security issues, but embedding security requirements into the organization's structured way of working so that improvements are durable, accepted, and repeatable.

What Is Expected From Students?

Students are tasked with designing a systematic approach for integrating security requirements into structured software development processes. They should propose how security can be embedded into existing workflows, architectural practices, and organizational responsibilities rather than treated as a one-off improvement. The solution should address how security aspects can become a recurring element of development activities, for example by integrating security considerations into design reviews, change requests, or release planning.

Students must also consider the organizational dimension of security work. This includes identifying relevant stakeholders such as technical leadership, architecture boards, process owners, and management, and describing how collaboration with these groups enables sustainable change. The proposal should explain how security improvements can be aligned with business priorities, resource constraints, and internal standards. As a reflection exercise, students should select one concrete security improvement, identify the organizational levers required to implement it, and explain how they would argue for the change to gain long-term support. The final outcome should demonstrate an understanding that effective security in industry is not only a technical challenge, but a structured, collaborative, and continuous organizational effort.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

CASE STUDY XXX: LEGACY IT MEETS NEW INTERFACES – FROM LEGACY SYSTEMS TO AN API ARCHITECTURE

Description of Company and Challenge

InterLink Systems is a company operating a complex IT and technology landscape that has evolved over several decades. Hundreds of developers have contributed to the systems, which support critical business operations and rely on long-standing interfaces and data formats. A newly hired junior software engineer joins the organization at a time when external requirements are changing rapidly. External partners now exchange larger volumes of data, use different formats, deliver data of varying quality, and require stricter security measures for data transfer.

To address these changes, the organization plans to introduce a new solution architecture consisting of a unified external API layer for all external partners and an internal adapter mechanism that connects this layer to existing systems. The adapters are intended to translate, convert, and enrich incoming data so that legacy systems can continue operating without major internal changes. However, this target architecture cannot be implemented uniformly across all systems. Some legacy components are difficult or impossible to adapt due to technical, contractual, or financial constraints. In addition, parts of the organization are resistant to moving away from familiar legacy interfaces. The central challenge is to introduce the new architecture gradually while maintaining stable operations and aligning both technical and human factors.

What Is Expected From Students?

Students are tasked with designing a pragmatic approach for introducing an external API and internal adapter architecture in a historically grown IT environment. They should propose a target architecture that provides a unified interface for external partners while shielding existing systems from disruptive changes through adapter mechanisms. The solution should explain how data conversion, validation, and enrichment are handled to cope with inconsistent formats and varying data quality.

Students must also address the transition aspect. This includes identifying which systems should be adapted first, which can temporarily remain outside the new standard, and how mixed environments can be managed safely. Appropriate use of design patterns, such as Adapter or Facade, should be explained as practical tools for migration rather than abstract theory. Since resistance to change is a key challenge, students should also propose communication and collaboration strategies for engaging skeptical colleagues, building trust, and demonstrating the value of the new architecture. As a reflection exercise, students should outline a simple architecture sketch, identify key transformation points, and formulate a concise explanation of the benefits of the API/adapter approach aimed at an experienced colleague. The final outcome should demonstrate balanced technical reasoning, awareness of organizational dynamics, and an ability to manage gradual system evolution responsibly.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.

CASE STUDY XXXI: FINDING THE RIGHT SERVICE BOUNDARY IN CLOUD AND KUBERNETES ARCHITECTURES

Description of Company and Challenge

CloudCore Solutions is a company undergoing a large-scale transformation toward cloud-native technologies. Several security-critical applications are being migrated and distributed across three cloud pillars that share a common technical infrastructure but are operated in separate geographic regions. A master plan for data center provisioning and architectural guidelines for the future application landscape already exist, defining high-level goals for scalability, security, and compliance.

Within this framework, teams developing Kubernetes-based solutions face a critical architectural decision: how to define meaningful service boundaries. Applications can be structured according to business domains or use cases, technical layers, or data domains. Each option has implications for deployment, operations, monitoring, access control, compliance audits, and responsibility assignment. Development teams may favor domain-driven structures for flexibility, while operations and compliance teams focus on stability, traceability, and regulatory clarity. In addition, several legacy applications cannot yet adopt the new service model but must still be integrated into the evolving cloud environment. Incorrect or misaligned decisions risk creating operational complexity, compliance gaps, and costly redesigns later in the project.

What Is Expected From Students?

Students are tasked with designing a structured decision-making approach for defining service boundaries in Kubernetes and cloud architectures. They should analyze different architectural partitioning models, such as business-driven, technical-layer–based, or data-domain–oriented splits, and evaluate their suitability in a multi-region, security-critical environment. The solution should propose clear decision principles that guide the selection of an appropriate partitioning strategy and explain how these principles support development efficiency, operational reliability, and compliance requirements.

Students must also identify the key stakeholders involved in such architectural decisions, including development, operations, and regulatory or compliance teams, and describe how their perspectives can be aligned early in the process. The proposal should address potential risks arising from unsuitable service boundaries or mismatched assumptions between teams and suggest ways to detect and mitigate these risks proactively. Finally, students should consider how legacy applications that cannot yet be modularized can be integrated into the target architecture without undermining the overall design. The final outcome should demonstrate sound architectural reasoning, strong coordination skills, and an understanding of how technical decisions in cloud environments are deeply intertwined with organizational and regulatory constraints.

Note: The company name used in this case study is fictional and serves only to protect the anonymity of the real organization.