Case Study 1:

A team of software engineers had incorporated a database containing user's details into the CMS project and was planning to release the software in the following week.

During the final software testing, the team realised that some security measures were not implemented properly and users' private details could potentially be compromised if the users used CMS Software in the future.

The team leader Alex was very concerned about the security risk to the public, did not approve the software and raised this with the CEO. He proposed to the CEO that the release date should be delayed until this problem was resolved.

However, due to business pressure from the client and for not damaging his personal reputation, the CEO decided to go ahead with releasing the product as planned but asked Alex and his team to resolve the issue by providing patches in the following weeks after the release.

Case Study 2:

Alex, Barry and Charlie were employed by Monash Pty Ltd as software engineers. The company had recently been given the contract to work on the CMS project.

The company CEO decided to assign Alex, Barry and Charlie to work on the project.

During requirements elicitation through client interview, the team found out that in order for CMS software to be compatible with the client company's operating system, CMS software must be implemented in Java, and not in any other programming languages.

Alex and Barry had no problem with Java, but Charlie had never programmed in Java ever, even though he had experience in other object-oriented languages. Realising their team mate's lack of Java programming experience, Alex and Barry mentioned that they were quite happy to teach Charlie the Java syntax, and said to Charlie that as long as he understood the concepts of OOP, he should be able to learn a new object-oriented language syntax quite easily.

However, Charlie insisted that he did not want to learn Java, stating that he can still work on the project without programming. He further stated that he will focus only on creating the UML diagrams and let Alex and Barry be the programmers for CMS.

Case Study 3:

A team working for Monash Pty Ltd has been working on the security system of CMS software for a few months.

One of the team members Alex decided to resign from the team and take up a job in the IT department of Seek.com, where he worked with a team of software engineers.

After a few years of hard work, Alex was recognised by the CEO for his effort and he was promoted to team leader.

Seek.com then decided to expand their business to add some further features of their website. Alex and his team were tasked to extend the existing security system to include the new functionalities.

After three months, the team got stuck at one of the design issues that could not be resolved. Alex then recalled that his previous team mate Barry from Monash Pty Ltd had produced something similar for a similar project a few years ago.

Alex informed his CEO and the CEO encouraged Alex to contact Barry, his old friend for help. Barry was happy to hear from Alex again and as a favour, explained to Alex how to resolve the security issues and pass him some of the Java classes to the rest of the team.

Alex appreciated Barry's help and immediately reused the Java classes and solved the problem for Seek.com. Hearing the good news from Alex, the CEO was very pleased with Alex's ingenuity.

Case Study 4:

A software team comprising Alex, Barry and Charlie, was tasked by the CEO of CMS to develop the prototype for their database. Given the requirements documents and armed with a number of use case scenarios, the team were developing the conceptual model for the system in terms of a class diagram.

Alex and Barry were familiar with object-oriented concepts but Charlie has no experience with object-orientation but has a lot of experience with database design in particular Entity-Relationship (ER) modelling.

Charlie realised his shortcomings and decided that he should not be involved in developing the class diagram and preferred to take on other duties.

However, Alex and Barry tried to encourage Charlie to stay with the team stating, “ER modelling is the same as class modelling. The only difference is the syntax. You can produce the Entity types, change them to Entity classes and join them up with different lines to produce the class diagram.”

Case Study 5:

Alex, Barry and Charlie were recently new software engineers employed by the CEO of Monash Pty Ltd. Alex was trained formally as system analyst while Barry and Charlie trained formally as computer programmers.

None of them were trained formally as software engineers. The CEO realised their unique backgrounds and decided to ask them to form a team to complement each other in software engineering projects. Their first project assigned to them was CMS.

Barry and Charlie knew a bit about software engineering, and therefore knew that the first task of any software project is requirements elicitation and analysis.

They suggested that since Alex was trained as system analysts, Alex should be involved in the first stage, while Barry and Charlie can work on another project. After the CMS software requirements were analysed and modelled, Barry and Charlie promised that they would return to program the system during the implementation stage.

Alex was happy to oblige because he planned to take some annual leave when Barry and Charlie were implementing the system. The team informed their CEO who promptly endorsed their plan.

Case Study 6:

Alex, Barry and Charlie employed as software engineers by Monash Pty Ltd were working on the CMS project.

The process model used in the project was the Unified Process. The project was in the final stage and the delivery deadline was getting close.

The head of the IT department was concerned that the project may not be completed on time and the software may not be ready for deployment in the store on the day specified in the contract.

To hasten the process, the head decided to employ a new person David to join the team so that the progress can be quicken and the project can be completed on time. Alex, Barry and Charlie were very concerned, stating that for someone to join the team so late in the project will surely cause a lot of problems, potentially may cause further delay and increase the project cost.

The head assured the team that it should be fine because David was an expert in testing. He rationalised that since the Test Workflow is the final workflow in the Unified Process, David will have no problem in contributing and helping the team in the final stage of the project focusing on testing the software.

The team decided to submit a formal complaint to the CEO, but the head refuted their claim and stopped the complaint in its tracks. He insisted that the team should accept David as the new member and should focus more on completing the CMS project.

Case Study 7:

Kim is working as a software developer for a software company that plans to develop the application for the inventory management module of CMS software.

The company provides Inventory Management Solutions for small-time shop owners.

However, Kim noticed that one of the features that was requested by their clients is not working as intended. This is because the feature is incomplete.

However, the front end of the system will still show a “Successful” message even though nothing really happens at all. This feature was the ”Add Product feature”. This feature was supposed to be completed by Kim’s co-worker Sam. Kim decided not to tell anyone regarding this matter even though the feature is supposed to be one of the more important features requested by the client.

Case Study 8:

Blake works as an intern software developer for a Monash Pty Ltd.
Blake was promised a position as a full time software developer by the CEO of the company, however, it was a verbal promise.
Blake did a great job in her team working together on the CMS project. After Blake's internship ended, she was not given a full time position as the CEO promised.
Blake was advised if she continues to intern for the company for another month, a full time position will be given. However, this is once again a verbal promise. Blake is undecided of what should be done.

Case Study 9:

Spencer works as a junior software developer at Monash Pty Ltd.

Spencer was assigned Quinn as a mentor. However during this period, Quinn as the senior developer told Spencer the solution to all the problems is Google and provided no help at all once requested. Spencer was tasked to work on the design of the CMS software and whenever he asked Quinn for help, Quinn would say, “Follow your heart”.

Over the course of 3 months, Quinn not only did not provide any form of mentoring but also said that Spencer is a weak developer to the upper management. Due to the lack of support, Spencer was late in submitting his designs.

Case Study 10:

Hailey is a newly hired intern at Monash Pty Ltd. On her first day, she noticed that the company is very secretive regarding how everything is done.

Hailey is not provided with the tools and equipment to do her job. She is assigned to work on gathering requirements from the client for the CMS software. She is not even provided with a computer and asked to bring her own equipment that is required for the job.

After the interview with the client, she handed over her work to the analysis team. The team reported back to her supervisor that she has done her job inefficiently due to the lack of experience and knowledge.

Case Study 11:

Tanya works as a Software Engineer at Monash Pty Ltd. Tanya is a part of the team developing the CMS software.

She reports to her Product Manager, Rowan who does not have a background in Information Technology.

Wheneven, she is required to discuss or seek advice, Rowan seems uninterested in communicating with her. Even the other team members in Tanya's team have troubles communicating with Rowan.

After a period of time, the senior developers decided to exclude Rowan from any form of communication as it was considered unproductive. It was later discovered that Rowan was hired for that position because of a close friendship with the CEO of the company.

Case Study 12:

James was employed as a software engineer to work on the CMS software. The CMS software will integrate data from the various existing systems, such as CRM, finance and human resource systems.

Due to family issues, James was given permission to work at home and access the company server remotely. James filled in the off-site working application and sent it to Peter, the project manager who was in charge of granting off-site working permission.

However, Peter went on leave for a month before signing the paper. James was not sure what to do and assumed that Peter had given the permission as he was aware of his family issue at a BBQ party.

James started working on the system and copied all of the data from the integrating system to his personal laptop. Several weeks later, when James was testing on newly developed features of the system, he found out that the data used for testing was hacked due to ransomware downloaded in his system. His laptop showed a picture of a computer screen with a message that the system is locked until the ransom is paid. James believed that he didn't have to pay ransom to get data, so he got data from the company again, reinstalled his operating system and kept on testing new features.

Case Study 13:

Haichi was employed as a software engineer by Monash Pty Ltd to develop a software application as a part of the research project led by Professor Elena. For this project, Prof. Elena was collaborating with a research student Aish from some other university. The team was supposed to publish the conducted work as an artefact (a tool paper).

The main role of Haichi was to develop a software application in Java by extending a research tool already developed by Aish during her research. Prof Elena was acting as a client to Haichi to let him understand the requirements of the new project and at the same time he was writing on the artefact as a tool paper. Aish was assisting Haichi to understand the functionalities of her previous tool and the underlying concepts.

In the new application, Haichi was consulting Aish to use some of the functionalities of Aish's tool. At some places, he changed the source code in Aish's application to implement some of the functionalities of the new application, without consulting and informing Aish about the change. Prof Elena wrote the technique (algorithm) developed to implement the new application where the first step was the work done by Aish in her previous work without citing the reference, assuming that Aish is the co-author of the new paper.