

Competências Comunicacionais I

1º semestre
(P1)

Effective Feedback and Feedforward in Software Engineering

Prof. Cláudia Silva

Nesta atividade, os alunos terão a oportunidade de aperfeiçoar as suas habilidades em dar e receber feedback e feedforward eficaz num contexto relacionado com o mundo da tecnologia. Serão divididos em pares e terão de interpretar e ser criativos em cenários relacionados com o campo das TI, nos quais terão de dar e receber feedback e feedforward. O objetivo é aplicar conhecimentos teóricos sobre feedback e feedforward eficaz na prática, desenvolver habilidades em análise crítica e comunicação clara e concisa e demonstrar pensamento reflexivo. Também serão incentivados a manter uma atitude respeitosa e abertura para receber feedback e feedforward, usando-o como uma oportunidade para crescimento pessoal e profissional. O principal objetivo é praticar o uso destas técnicas, desenvolver habilidades de comunicação e promover a troca de ideias construtivas.

Objetivos de Aprendizagem

Até ao final desta tarefa, o aluno deverá ser capaz de:

- Aplicar conhecimentos teóricos sobre feedback e feedforward eficaz de forma prática.
- Desenvolver habilidades em análise crítica.
- Fornecer feedback eficaz, identificando pontos fortes, positivos e negativos de forma concisa, clara e objetiva.
- Utilizar feedforward, oferecendo sugestões construtivas para melhorar esses pontos no futuro.
- Desenvolver habilidades de comunicação, expressando ideias e sugestões de forma clara e concisa.
- Demonstrar pensamento reflexivo, analisando o feedback e o feedforward que recebem.
- Participar em discussões construtivas e trocar ideias com colegas.
- Objetivo-chave da Tarefa: Compreender a diferença e importância de feedback (focando em ações passadas) e feedforward (focando em melhorias futuras) nas práticas de engenharia de software.

Instruções:

- Encontre um colega com quem trabalhar. Um aluno desempenhará o papel de 'Desenvolvedor' e o outro de 'Revisor'.
- Escolha dois cenários da lista abaixo.
- O 'Revisor' deve fornecer feedback e feedforward com base no cenário, aplicando as técnicas aprendidas durante a aula teórica.
- Depois de discutir cada cenário, troque de papéis e passe para o próximo cenário.
- Encene e apresente o seu cenário para a turma.
- Após a realização do cenário, os alunos devem ser capazes de explicar quais técnicas utilizaram.

Cenários:**1. Correção de *Bugs*:**

Situação Problemática: **Sara**, uma programadora sénior meticulosa, notou que o recém-contratado desenvolvedor, João, conseguiu resolver um *bug* crítico que deixara a equipa perplexa. No entanto, ela não encontrou nada quando procurou detalhes sobre a correção no sistema de documentação deles. Ela percebeu a importância de orientar **Jake** sobre o valor da documentação.

2. Qualidade do Código:

Situação Problemática: Na agitada startup 'TechDream', **Pedro** era conhecido por produzir resultados rapidamente. No entanto, **Maria**, a líder da sua equipa, achava difícil decifrar o seu código. Embora fosse funcional, a falta de comentários e métodos pouco convencionais tornavam-no um enigma para todos os outros.

3. Entregas Atrasadas:

Situação Problemática: Todos na 'NexaSoft' conheciam **Alex** pela sua criatividade incomparável. Mas a sua gerente, **Rachel**, também o conhecia por outra razão: ele falhava constantemente nos prazos. As suas entregas tardias, embora repletas de inovação, perturbavam o ritmo da equipa.

4. Colaboração em Equipa:

Situação Problemática: Na 'BlueWave Technologies', **Joana**, uma programadora apaixonada, frequentemente mergulhava profundamente no seu mundo de codificação, esquecendo o mundo exterior. Os seus colegas de equipa, embora respeitassem a sua dedicação, frequentemente sentiam-se desconectados, uma vez que **Joana** raramente partilhava o seu progresso ou desafios.

5. Cobertura de Testes:

Situação Problemática: **Bernardo** era um desenvolvedor estrela no ciclo de desenvolvimento rápido da 'QuickSoft'. Mas **Amy**, a responsável pela QA, notou um problema recorrente. As funcionalidades de Bernardo, embora excelentes, vinham sem quaisquer testes, tornando o processo de QA num jogo de adivinhação.

6. Revisões de Código:

Situação Problemática: **Paulo**, o líder na 'FutureTech', sentia-se inquieto. Observou que sempre que **Rita**, uma desenvolvedora júnior, revisava código, o seu feedback era genérico. A equipa precisava de mais do que apenas um "Parece bom" para garantir a robustez do software.

7. Aprendizagem Contínua:

Situação Problemática: 'SolarTech' era uma empresa na vanguarda, sempre a evoluir. No entanto, **Mike**, um dos desenvolvedores mais experientes, parecia preso aos seus métodos antigos. **Luciana**, uma colega desenvolvedora, notou que os métodos de Mike, embora testados e comprovados, estavam a ficar ultrapassados.

8. Otimização de Performance:

Situação: Na 'Zeta Systems', o código do **Carlos** era sempre funcional. Nunca falhava. Mas quando era submetido a uma auditoria de performance pelo analista de sistemas, Nuno, frequentemente mostrava ineficiências, exigindo mais recursos do que o necessário.

9. Recebendo Feedback:

Situação Problemática: No plano aberto da 'NeuroTech', as discussões eram frequentes. Mas quando **Samuel**, o líder do projeto, oferecia feedback ao **Miguel**, a atmosfera da sala ficava tensa. Miguel, embora brilhante, tinha dificuldade em ver o feedback como uma ferramenta em vez de uma crítica.

10. Preocupações com Escalabilidade:

Situação: 'DataDynasty' era uma startup em ascensão com uma base de utilizadores crescente. **Emília**, a CTO, percebeu que, embora as soluções do **Ronaldo** fossem perfeitas para o presente, faltava-lhes a previsão de escalabilidade, levantando preocupações para o crescimento futuro.

Debrief:

Depois de tu e o teu colega terem passado pelos cenários, reagrupem-se e discutam a importância do feedback e do feedforward no processo de desenvolvimento de software com a turma e instrutores.