**Práticas:**

1. Experiência como critério para determinar o custo do ICP. **Fala:** “Vamos supor que a pessoa tem experiência em tudo, a pessoas vai ter o critério onde pode ser todos 1 de ICP, ou nem coloca, por exemplo, branches e loops”. Contexto do projeto como critério para adicionar itens na Tabela de ICPs do Time. **Fala:** “Depende muito do que o projeto precisa. Sempre vai depender de cada projeto, do que cada projeto vai precisar. Por exemplo, se fosse um projeto somente de partes visuais, não precisaria de gerenciamento de estados e muito menos de função assíncrona”.

1. Raras consultas na Tabela de ICPs do Time. **Fala:** “Ela era um guia que eu olhava de vez em quando. Principalmente na hora do desenvolvimento, porque eu sempre verificava após final aquela parte que eu estava desenvolvendo”.

1. Atividades de identificação, apontamento e cálculo de ICP das classes somente no final do processo de codificação.
2. Componentização e acoplamento como estratégia de refatoração para redução de complexidade. **Fala:** “Como eu estava trabalhando na parte visual, eu via que eu tava repetindo muitas linhas que poderiam ser componentizadas e chamadas em apenas uma linha”.

**Benefícios:**

1. Melhorias na complexidade e legibilidade do código. **Fala:** “A gente tinha o cuidado se a pessoa ia entender o nosso código… Não adianta só a pessoa que tá programando o código entender, e outra pessoa bater o olho no código e não entender”.
2. CDD como agente de atenção ao código. **Fala:** “O CDD ele te ajuda a parar pra pensar se quando tu entregar o projeto, quando ele estiver finalizado, qualquer outro revisor de código ou a pessoa responsável pela manutenção futura vai conseguir fazer aquele de uma maneira mais fácil, de uma maneira mais eficaz… É um agente que ficar te perturbando pra querer melhorar”.
3. Constante refinamento da Tabela de ICPs do Time. **Fala:** “Acaba te fazendo perguntar a si mesmo se aquele item faz sentido. Por exemplo, ele também te mostra se você aprendeu algo mais ou ganhou mais experiência naquela etapa durante uma tabela e outra. Até mesmo para compreender se o projeto melhorou ou piorou durante essa etapa de desenvolvimento”.

**Desafios:**

1. Processo gradual de aprendizado. Não compreendeu completamente o processo e as atividades na parte inicial do projeto, mas ganhou entendimento ao longo do projeto. **Fala:** “Bem no início do projeto, eu não tinha essa visão do porquê usar o CDD. Por que ele não tava fazendo muita diferença no projeto, pra mim… No início ele não foi tão relevante pra mim, mas ele começou ficar importante a partir da metade do projeto”.
2. Baixo impacto no começo do projeto. **Fala:** “No início do projeto, na minha opinião, o CDD não faz tanta diferença assim. Acho que é até um pouco lógico dele não fazer diferença porque o grupo poderia pensar que está desenvolvendo certinho e que não vai dar problema nenhum”.
3. Redução da complexidade do projeto (ao invés do código). **Fala:** “O arquivo Main, depois da refatoração ele ficou com bem menos linhas de código… Mesmo assim se uma pessoa de fora desse projeto, ela pegar esse arquivo e for ler, acho que ela não conseguiria entender, a não ser que ela visitasse cada um desses arquivos que estão sendo chamados nessa classe. Então nesse ponto, acho que o CDD não facilitou a leitura mesmo não estourando a carga máxima cognitiva”.

**Lições:**

1. Utilizar em projetos longos e com trabalho em equipe.
2. Realizar as atividades durante todo o processo de codificação. **Fala:** “Eu não senti tanta eficácia assim quanto eu senti em um período depois, quando fui testar quando tava desenvolvendo e consultando a tabela ao mesmo tempo”.