**Práticas:**

1. Baixo grau de experiência como critério para adicionar itens na Tabela de ICPs do Time. **Fala:** “Eu determinei a experiência que eu tinha de usar aquela ferramenta da linguagem. Por exemplo, Provider, flutter\_bloc, FutureBuilder e StreamBuilder. Então isso interferiu na forma como eu escolhi esses itens da tabela”.
2. Raras consultas na Tabela de ICPs do Time. **Fala:** “Eu usava a tabela toda vez quando eu terminava uma feature”.
3. Atividades de identificação, apontamento e cálculo de ICP das classes somente no final do processo de codificação. **Fala:** “Eu fazia sempre no final. Terminava de codar, aí depois eu pegava a tabela analisava cada categoria em cada widget diferente, e depois no final eu ia somando”.
4. Componentização e acoplamento como estratégia de refatoração para redução de complexidade. **Fala:** “Eu refatorava. Tirava isso da classe, transformava em uma só e fazia o acoplamento disso. Fazia dessa forma para diminuir o código… É, eu fazia isso, uma componentização”.

**Benefícios:**

1. Ajudou a diminuir classes com muitas linhas de código. **Fala:** “Eu acho que em alguns códigos foi útil, porque realmente estavam muito grandes”.
2. Constante refinamento da Tabela de ICPs do Time. **Fala:** “Acho que ajudou. Porque na primeira tabela a gente tinha alguns pontos a mais de CDD e isso depois foi corrigido”.
3. Classes mais granuladas e componentizadas. **Fala:** ”Querendo ou não a gente conseguiu componentizar mais. Em alguns casos, na minha opinião, foi uma componentização boa”.
4. Legibilidade. **Fala:** “Um indicador seria na organização dos arquivos e na facilitação da leitura dos arquivos”.
5. Boa aceitação do CDD pela Equipe.

**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.
2. Representar corretamente a complexidade da classe percebida pelo desenvolvedor. **Fala:** “Apesar de os pontos não corresponderem ao que estava combinado, mesmo assim dava pra ler bem o código”.
3. Aprofundamento de custos da Tabela de ICPs do Time. **Fala:** “Quando eu precisava fazer alguns filtros do banco de dados, era muito mais difícil entender o for ou while do que if ou else. Acho que não faz sentido eles terem o mesmo ponto”.
4. Dificuldade em manter as atividades de CDD (identificar, apontar e calcular) em classes com muitas linhas de código. **Fala:** “Eu acho que foi a classe de visualizar projetos que ela era muito grande. E a classe de visualizar demandar que ela também era bem grande”.
5. Impacto no início do projeto. **Fala:** “Mais para o final que a gente conseguiu aplicar mais o CDD, já que durante o projeto a gente praticamente não estourava ele. Então foi mais para o final que o limite caiu bastante e a gente teve que cortar bastante coisa”.

**Lições:**

1. Adicionar outras abordagens. **Fala:** “Eu acho que antes de usar o CDD, eu tentaria ver uma forma melhor de organizar os arquivos”.
2. Investir mais tempo na definição da Tabela de ICPs do Time. **Fala:** “Decidir bem o limite do CDD… Discutiria mais essa questão do if, else e for”.