**Práticas:**

1. Baixo grau de entendimento como critério para adicionar itens na Tabela de ICPs do Time. **Fala:** “Era mais coisas que eu não entendia como funcionava direito”.
2. Raras consultas na Tabela de ICPs do Time.
3. Atividades de identificação, apontamento e cálculo de ICP das classes somente no final do processo de codificação. Fala: “Quando tava pronto eu abria a tabela e fazia a contagem dos pontos”.
4. Componentização como estratégia para redução da complexidade das classes. Outra forma, funções separadas. **Fala:** “Eu componentizava algum widget que estivesse muito complexo. Dividia ele em partes. Quando era uma questão de lógica básica de programação tipo um for… Eu tentava colocar dentro de uma função separada”.

**Benefícios:**

1. Classes mais simples e mais fáceis de entender. **Fala:** “Depois que a gente fez uma boa determinação de qual era o limite, quais eram os pontos de cada coisa… a gente conseguiu deixar elas mais simples. O CDD avisou a gente sobre isso”.
2. Constante refinamento da Tabela de ICPs do Time. Além de melhoria gradual da percepção da complexidade. **Fala:** “Acho que ter essa reunião toda semana para verificar a tabela foi bem útil. Já que o nosso nível de entendimento do Flutter foi mudando a cada semana”.
3. Aprofundar a definição dos custos dos ICPs de forma abrangente. Tomando como exemplo funções assíncronas e os diferentes custos para criação e manipulação. **Fala:** “Acho que tem casos que realmente é diferente a complexidade de usar ou criar”.
4. Organização e separação das classes do projeto. **Fala:** “Mesmo que a gente não estivesse usando algum tipo de arquitetura para o projeto. O CDD por limitar a complexidade das classes, acabou deixando um pouco mais organizada”
5. Melhor legibilidade e complexidade das classes.
6. Boa aceitação do CDD pela Equipe

**Desafios:**

1. Entendimento dos itens da Tabela de ICPs do Time no início do projeto. **Fala:** “No início tive mais dificuldade de entender o que deveria e não deveria ser considerado. Os pontos de complexidade de cada coisa”.
2. Manter o CDD em tarefas mais difíceis. **Fala:** “Se tivesse algum card, alguma tarefa pra fazer que estava muito difícil. Acabava me focando muito nela e deixando só pra hora de entregar a contagem de pontos”.
3. Conciliar as atividades de CDD com as atividades de desenvolvimento. Fala: “Porque era uma etapa a mais no trabalho, então, às vezes a pessoa acabava esquecendo ou ainda tava aprendendo a fazer. Inclusive, no início, algumas vezes eu não contabilizei direito, mas era porque eu tava me acostumando com essa metodologia”
4. Calcular manualmente os itens da tabela.

**Lições:**

1. Primeiro “codar”, depois criar a tabela do time. **Fala:** “Achava melhor primeiro a gente fazer o código e depois a tabela que provavelmente a gente ia ter os problemas maiores que a gente ia ver quais são os itens que mais impactam para poder adicionar na tabela”.
2. Adicionar mais metodologias para apoiar a refatoração. Fala: “Analisar e limitar a complexidade das classes, isso realmente aconteceu. Foi feito! Acho que ele deveria ter outra metodologia de desenvolvimento para ajudar na qualidade do código. E seria muito bom se tivesse uma ferramenta para fazer a contagem automaticamente”.
3. Investir mais tempo para definir a Tabela de ICPs do Time e capturar melhor as experiências dos desenvolvedores. Fala: “Acho que depois que faz uma boa definição da tabela fica bem mais fluido esse processo”.