UNIVERSIDADE FEDERAL DO CEARÁ

Code smells em frameworks front-end

Disciplina: Qualidade de Software
Equipe: José Iago da Silva Lima
Projeto: igniteui-angular
Fork: https://github.com/IagoLSJ/igniteui-angular

Eu estou atualmente trabalhando na refatoração do seguinte code smell: Too Many Inputs (TMI), caracterizado por componentes que possuem mais de 8 (parâmetro retirado do código da ferramenta de detecção) @Input(), dificultando a manutenção, reutilização e leitura do código e, em alguns casos, deixando o arquivo grande.
Minhas principais dificuldades na remoção do code smell são: Identificar quais Input realmente pertencem ao mesmo contexto funcional. não quebre bindings existentes no template.
Eu estou usando os seguintes métodos de refatoração para remover o code smell: Agrupar inputs que têm um certo contexto entre si em interfaces tipadas.

Eu estou atualmente trabalhando na refatoração do seguinte code smell: DOM Manipulation, caracterizado pelo acesso direto ao DOM usando nativeElement, document, window ou remove(), o que quebra o padrão Angular e dificulta testes e manutenção.
Minhas principais dificuldades na remoção do code smell são: Encontrar equivalentes funcionais para ações como remover elementos e alterar classes sem usar Renderer2.
Eu estou usando os seguintes métodos de refatoração para remover o code smell: Mover lógica visual para bindings e diretivas. delegar ações visuais para o próprio componente IgniteUI. Eliminação de efeitos colaterais: remover dependência de nativeElement e acesso global ao DOM.

Eu estou atualmente trabalhando na refatoração do seguinte code smell: Large File (LF), caracterizado por arquivos muito grandes (240 linhas), que concentram muitas responsabilidades em um único componente ou serviço, dificultando a leitura, manutenção e evolução do código.
Minhas principais dificuldades na remoção do code smell são: Identificar corretamente os limites de responsabilidade dentro dos arquivos e

quebrar dependências entre métodos e propriedades ao dividir o código.
Eu estou usando os seguintes métodos de refatoração para remover o code smell: criação de novos serviços, helpers e utilitários, divisão do componente em partes menores e mais coesas e organização do código em arquivos menores e com responsabilidade única.

Eu estou atualmente trabalhando na refatoração do seguinte code smell: Any Type, caracterizado pelo uso excessivo do tipo any em aplicações Angular, o que reduz a segurança de tipos e aumenta a chance de erros em tempo de execução.
Minhas principais dificuldades na remoção do code smell são: Entender a estrutura real dos dados quando não existe tipagem prévia.
Eu estou usando os seguintes métodos de refatoração para remover o code smell: substituição de any por interfaces tipadas,aplicação sistemática de tipagem forte em serviços, componentes e modelos e criação de tipos reutilizáveis.