

Paulo Meirelles

Anomalia vs cheiro de código

Problemas ortográficos (vírgulas)

Reduzir a quantidade de citações diretas

Paulo falou da 1a frase do abstract – time to market e que faz parte de métodos ágeis desenvolver dessa maneira

Discutir em algum momento a camada de apresentação em plataforma nativa vs cross

Definir código limpo a partir do livro do Clean Code

Problema na referencia do Aniche

Problema de encadeamento na introducao – demora a entender o que vai fazer

Em alguns momentos demanda que seja lido um determinado trabalho (ex: Aniche)

Seguir o trabalho do Aniche

Problema de testabilidade – está tratando isso na camada de apresentação?

Deixar claro que nao vai olhar XML, JavaScript

Deixar claro que Propensão a Mudanças = commits -> explicar melhor esta parte

Fazer a mineração vai ser melhor do que fazer um experimento controlado

Usar uma análise qualitativa baseada em Grounded Theory para produzir um mapa mental (produzir categorias e seus relacionamentos) – inspirar no trabalho do Igor

Paulo recomenda se apoiar fortemente na metodologia do Igor (na parte da qualitativa) e do Aniche

Paulo sugere não separar desenvolvimento e manutenção

Depois que achar os maus cheiros – as atuais ferramentas os encontrariam?

Em português é melhor falar software livre do que código aberto

Android – a maior parte dos componentes é livre

Dar menos ênfase a trabalhos de segurança e aumentar a comparação com os trabalhos fortemente relacionados – mostrando o que é igual e o que é diferente

Traz detalhes dos trabalhos que não vai usar depois (resultados, métricas, etc.)

Incluir mais especialistas – Paulo citou o número mágico de Nielsen de 5 sujeitos

Olhar o projeto da calculadora oficial do Google e ver se tem mal cheiro. Se não tiver, inserir mal cheiros.

Vai deixar pedaço em branco para o participante completar?

Alfredo

O texto deveria ser mais autocontido

Esperava que nessa qualificação já tivesse encontrado alguns padrões – poderia entrar como indícios de que o problema é interessante

Tem que trabalhar mais na motivação – Alfredo não viu o porquê é interessante estudar a camada de apresentação de Android

Alfredo falou que não está escrito porque a camada de apresentação e não o resto

Talvez não seja verdade que nas outras camadas os outros smells são suficientes – faltam indícios

A motivação focou muito em Android, mas não na camada de apresentação

A camada de apresentação é quantos por cento do código? Isso pode ajudar a justificar

Poderia colocar alguns exemplos de sua experiência para ajudar o texto

Referencia da tese de 2008 – checar

Deveria ter muito mais destaque o que está no slide 17 (isso fortalece o trabalho)

Poderia já ter colocado a análise das 44 respostas do questionário

Não está escrito que fez um piloto do questionário

Faltou uma discussão sobre termos: smells, anti-patterns, anomalias, dívida técnica – Alfredo recomenda até adotar anti-patterns {a propósito, vai tratar code smells ou são anomalias mais relacionadas a design)

Erros de crase

Falar um pouco mais dos problemas em geral de Android (multiplicidade de devices e versões etc.)

Um mal cheiro não leva a comportamento errôneo -> Não dá para afirmar tão categoricamente isso. Pode resultar sob certas situações

Para falar em refatoração sem contextualização – tem que ter teste automatizado

O cap de método de pesquisa está repetitivo em relação à introdução

Alfredo acha que precisa ter mais foco.

Alfredo não achou claro o questionário