

Decisões

Para evitar que os dados fossem inconsistentes, tomei a decisão de trabalhar somente com os commits que causaram alguma alteração no código, os chamados commits de mudança. Para isso selecionei os autores e seus respectivos commits e adicionei na tabela 'COMMIT_MUDANÇAS_2', tabela que contém os campos "nome do projeto, nome do autor, commitHash de mudança e o commitHash original", os campos referente aos commitHash são para a confirmação de que o commit que está sendo contabilizado é o de mudança, por isso ambos os campos devem ter os valores iguais.

Através da tabela de COMMIT_MUDANÇAS_2 foi possível obter a quantidade de linhas editadas de cada desenvolvedor. Relacionando os campos de commit da atual tabela com os campos de commit da tabela GIT_COMMIT_CHANGES e somando para cada commit a quantidade de linhas adicionadas ou removidas.

Para os code smells foram selecionados somente as anomalias detectadas pela ferramenta Ptidej, que informa o tipo de smell detectado.

Com a tabela de commit de mudança pronta, foi feita a busca pela quantidade de code smell que cada desenvolvedor implementou. Relacionando os campos de *commitHash* da tabela COMMIT_MUDANÇAS_2 com o campo de commit de criação da tabela SONAR_ISSUES e somando a quantidade de smells relacionada a cada commit.

Logo após a obtenção do total de commits implementado por cada desenvolvedor, o próximo passo foi separar os tipos de cada code smell detectados. Para cada tipo de code smell detectado existe uma tabela que contém a quantidade daquele tipo de smell que cada desenvolvedor implementou, o nome do desenvolvedor e o nome do projeto.

Esses dados obtidos foram inseridos na tabela BASE_ULTIMATE que contém o nome do autor, projeto, quantidade de linhas editadas (code churn), o total de smells de cada autor, e os tipos de smells que cada autor implementou.

Inicialmente estava sendo considerando também os autores que não tinham feito commits de alteração de código, o que estava causando inconsistência nos dados na hora de contar a quantidade dos tipos de smells, pois alguns autores não apresentavam na lista de desenvolvedores que haviam feito mudanças. Alguns scripts de busca de dados também não estavam confiáveis o que foi preciso fazer algumas alterações.

Tabelas

COMMIT_MUDANCAS_2

Esta tabela contém todos os desenvolvedores que fizeram alguma alteração no código. *(Não está sendo considerado o primeiro commit realizado)*. Para evitar principalmente a contabilização de commits que não são de alteração.

Para a criação dessa tabela foi feito uma junção entre a tabela GIT_COMMIT e GIT_COMMIT_CHANGES com a condição de *GIT_COMMIT.commitHash = GIT_COMMIT_CHANGES.commitHash* onde buscamos o nome do autor, nome do projeto e o commit que foi relacionado.

BASE_ULTIMATE

Esta tabela é considerada a principal, onde contém o nome do projeto, nome do desenvolvedor, quantidade total de linhas editadas (code churn) de cada desenvolvedor, quantidade de code smell implementado por cada desenvolvedor e a quantidade de cada tipo de smells que cada desenvolvedor implementou.

Tabelas dos tipos de Smells

É um conjunto de 12 tabelas onde cada tabela representa um tipo de smell detectado pela ferramenta Ptidej. Nela contém os campos nome do projeto, nome do autor, quantidade de smell implementado por cada desenvolvedor.

Obtenção dos dados

Primeiramente foi criada a tabela *COMMIT_MUDANCAS_2* e dela deriva quase todas as operações.

Code Churn

O code churn é a quantidade total de linhas que foram editadas pelo desenvolvedor.

Para a obtenção desse dado foi feita a junção entre as tabelas *COMMIT_MUDANCAS_2* e *GIT_COMMITS_CHANGES*, onde relacionamos como condição os campos *commitHash* de ambas para somarmos os campos *linesAdded* e *linesRemoved* da tabela *GIT_COMMITS_CHANGES*.

Nesta busca, foi obtido o nome do projeto, nome do autor e a quantidade total de linhas editadas (*code churn*) de cada desenvolvedor.

Total Code Smells

O objetivo dessa busca é obter a quantidade total de smells que um desenvolvedor implementou.

Para esses dados foi feita a junção da tabela *COMMIT_MUDANCAS_2* e *SONAR_ISSUES* onde o critério de relacionamento foi o campo *commitHash* da tabela *COMMIT_MUDANCAS_2* ser igual ao campo *creationCommitHash* da tabela *SONAR_ISSUES* e o campo *squid* de *SONAR_ISSUES* for igual a “*code_smells: %*”, com isso garantimos que estamos somando os smells do commit certo e ser da ferramenta *Ptidej*. Por fim somamos a quantidade de *creationCommitHash* que tem relação direta com os code smells.

Tipos de smells detectados pelo Ptidej

O objetivo dessa busca é encontrar os tipos de code smells detectados pelo Ptidej e a sua quantidade

Primeiramente foi necessário descobrir quais eram os tipos de code smells que existiam no *Data Set* trabalhado, para isso foi feita a busca na tabela *SONAR_ISSUES* no campo *squid* e usando o *Distinct* do SQL foi possível saber quantos tipos de code smells estavam sendo trabalhados no *Data Set*.

Sabendo os tipos de smells existentes fez-se a contagem de cada smell relacionando com o autor e o projeto. Essa contagem foi feita relacionando as tabelas *COMMIT_MUDANCAS_2* com a tabela *SONAR_ISSUES*.

Cálculo das correlações

Considerando todos os autores

Aqui está sendo considerado todos os autores, os que tem smells detectados pela ferramenta Ptidej e os que não foram detectados pela ferramenta

Variável	Correlação (rho)	Significância (p-value)
Code_churn	0.4373372	2.334284e-114

Considerando somente os autores com smells detectados

Aqui está sendo considerado somente os autores que tiveram smells detectados pela ferramenta Ptidej

Variável	Correlação (rho)	Significância (p-value)
Code_churn	0.6825855	1.209077e-48

Imagens Tabelas - Exemplos

BASE_ULTIMATE

```
+ :: CREATE TABLE "BASE_ULTIMATE" (  
  "projectID" TEXT,  
  "author" TEXT,  
  "linesEdited" REAL,  
  "code_smells" INTEGER,  
  "smell_antisingleton" INTEGER,  
  "smell_baseclass_abstract" INTEGER,  
  "smell_class_data_private" INTEGER,  
  "smell_complex_class" INTEGER,  
  "smell_large_class" INTEGER,  
  "smell_lazy_class" INTEGER,  
  "smell_long_method" INTEGER,  
  "smell_long_parameter_list" INTEGER,  
  "smell_many_field_attributes_not_complex" INTEGER,  
  "smell_refused_parent_bequest" INTEGER,  
  "smell_spaghetti_code" INTEGER,  
  "smell_speculative_generality" INTEGER  
)
```

Tabelas Smells

- ▶ Baseclass_abstract
- ▶ Class_data_private
- ▶ Complex_class
- ▶ Large_class
- ▶ Lazy_class
- ▶ Long_method
- ▶ Long_parameter_list
- ▶ Many field attributes not com...

Commit_Mudancas_2

projectID	author	commit_hash	commit_hash_changes
Filtro	Filtro	Filtro	Filtro
accumulo	Keith Turner	e0880e263...	e0880e263e4bf8662ba...
accumulo	Billie Rinaldi	e8774c5ec3...	e8774c5ec3a35e042f3...
accumulo	Keith Turner	2032ebbd0...	2032ebbd0ed90734da...
accumulo	Billie Rinaldi	de297d493...	de297d4932e08625a5...
accumulo	Billie Rinaldi	34efaae876...	34efaae87639a83b60f...
accumulo	Keith Turner	7e9bba037...	7e9bba037c4fbd70c7e...