

Software Systems as Cities: A Controlled Experiment

Renan de Melo Oliveira

13 de maio de 2013

Conteúdo

Introdução

Método de pesquisa

Execução

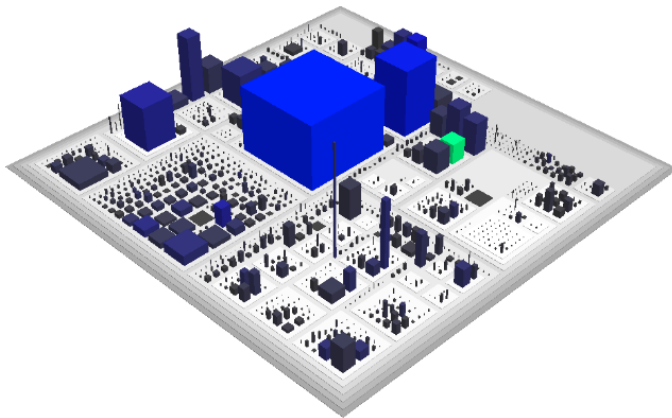
Resultado

Conclusões

Introdução

- ▶ CodeCity: Software para visualização de código;
- ▶ Representa o código de sistemas como cidades.

CodeCity



Perguntas de pesquisa

- ▶ Pacotes - Bairros;
- ▶ Classes - Prédios;
- ▶ Métodos - Altura dos prédios;

Perguntas de pesquisa

1. CodeCity auxilia na correção?
2. CodeCity reduz o tempo em tarefas de compreensão?
3. Quais são os tipos de tarefas em que ocorre diferença em ambos?
4. Quais são os benefícios potenciais em se usar o CodeCity destes aspectos em relação ao background do participante?
5. Quais são os benefícios potenciais em se usar o CodeCity destes aspectos em relação ao nível de experiência do participante?

Experimento controlado

Survey

Mais de 50 artigos e livros!

1. Escolha uma base de comparação justa;
2. Envolver participantes da indústria;
3. Leve em consideração o nível de experiência dos participantes;
4. Promova um tutorial da ferramenta experimental para os participantes;

Experimento controlado

5. Encontre um conjunto de tarefas relevantes;
6. Inclua tarefas que possam não dar vantagem à ferramenta sendo avaliada;
7. Limite o tempo de cada tarefa;
8. Escolha sistemas da vida real;
9. Escolha mais de um sistema para aplicar o experimento;
10. Promova os mesmos dados a todos os participantes;
11. Reporte resultados nas tarefas individuais;
12. Proveja todos os detalhes necessários para que o experimento seja replicável;

Hipóteses

- ▶ Nulas:
 - ▶ A ferramenta não impacta em: 1. Correção; 2. Tempo para completar tarefas;
- ▶ Alternativas:
 - ▶ A ferramenta impacta em: 1. Correção; 2. Tempo para completar tarefas;

Variáveis

- ▶ Independentes:
 - ▶ Ferramenta;
 - ▶ Tamanho do sistema.
- ▶ Dependentes:
 - ▶ Correção;
 - ▶ Tempo para conclusão de tarefa;

Variáveis

- ▶ Independentes:
 - ▶ Ferramenta;
 - ▶ Tamanho do sistema.
- ▶ Dependentes:
 - ▶ Correção;
 - ▶ Tempo para conclusão de tarefa;

Baseline

- ▶ Ferramenta: Construída (Eclipse + Excel)! Não haviam disponíveis com todas as features necessárias.
- ▶ Variáveis controladas:
 - ▶ Background: Profissional, Acadêmico;
 - ▶ Experiência: Iniciante, Avançado;
- ▶ Design de blocos aleatórios: Variáveis controladas x Variáveis independentes.

Tarefas

- ▶ Tarefas inspiradas em Pacione et al e Sillito et al.

Operação

- ▶ Duas fases:
 - ▶ Estudo piloto: alunos da Universidade de Lugano.
 - ▶ Experimento: 6 execuções em Quatro locais diferentes;

Coleta de Dados

- ▶ Informações pessoais: Questionário online;
- ▶ Dados de tempo: Aplicação Smalltalk;
- ▶ Dados de correção: Avaliação cega por 2 de 3 experimentadores e depois consenso;
- ▶ Feedback dos participantes: Questionário ao final;

Análise dos dados

- ▶ Remoção de outliers: Indivíduos que fogem muito do padrão e dificultam a análise (Ex: indivíduo sem exp. no Eclipse);

Hipóteses

- ▶ CodeCity proveu, em relação ao (Eclipse + Excel) melhoria de:
 - ▶ 24,26% na correção;
 - ▶ 12.01% no tempo para completar a tarefa;
- ▶ Estatisticamente relevante! (?)

Tipos de tarefas

- ▶ CodeCity foi melhor que (Eclipse + Excel) em quase todas as tarefas, com exceção das tarefas “focadas” (A4.1, B1.1);
- ▶ Excel é bom em tarefas precisas (ex. Encontrar o maior N). Porém mesmo nessas tarefas, o CodeCity conseguiu se equiparar a ele.

Ameaças à validade

- ▶ Validade Interna: Indivíduos, Tarefas escolhidas, Ferramenta composta, Diferenças de dados, Diferenças nas sessões, Treinamento;
- ▶ Validade Externa: Indivíduos, Tarefas escolhidas, Sistemas, Efeito do experimentador.

Conclusões

- ▶ Aparentemente, CodeCity é melhor nas variáveis dependentes escolhidas;
- ▶ Mesmo para tarefas precisas, CodeCity se equiparou a Eclipse + Excel.