Measuring Programming Experience

Janet Feigenspan
Christian Kastner
Jorg Liebig and Sven Apel
Stefan Hanenberg

IEEE 20th International Conference on Program Comprehension (ICPC)

Introdução

- Compreensão de programas é frequentemente avaliado;
- Processo cognitivo que não é observado diretamente;
- Mas frequentemente conduzimos experimentos, observamos e tiramos conclusões;
- Experiência de programação é um dos parâmetros mais confundidos;
- Maior entendimento pode aumentar a validade dos experimentos e ajudar na interpretação dos resultados, assim como a ignorância pode ser uma ameaça a validade;

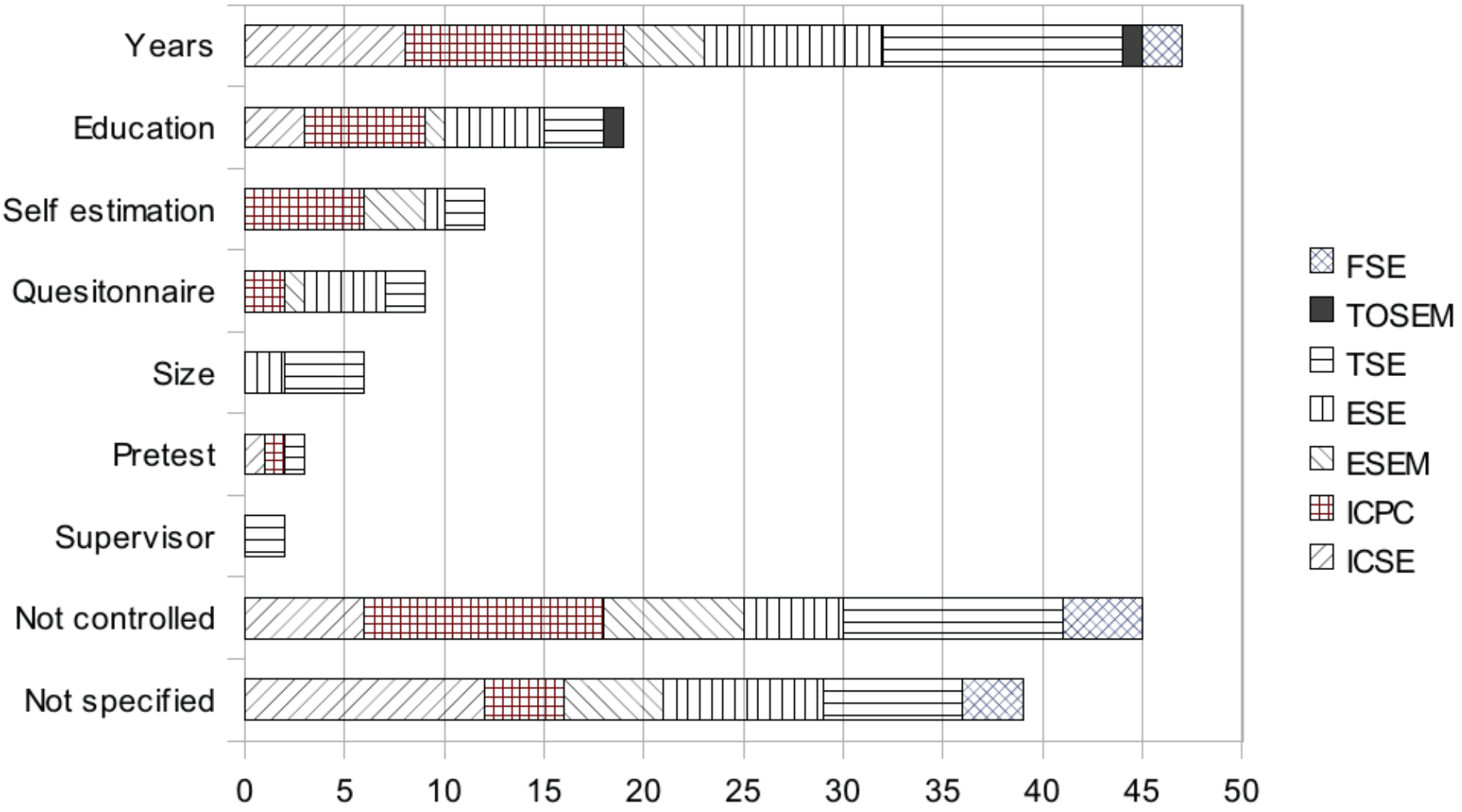
Objetivo

Avaliar quão confiáveis são diferentes formas de medir a experiência de programação?

Revisão da Literatura

- Baseada nas diretrizes providas por Kitchenham and Chartes;
- Restrições:
  - Artigos publicados entre 2001 e 2010;
  - Somente conferências e jornais “top” na área de engenharia de software empírica e compreensão de programas (ICSE, ICPC, ESEM, entre outras);
- 2.161 resultados no qual 288 foram selecionados (pré)selecionados fazendo a leitura do título, resumo e palavras-chave;
- Dos 288 artigos da etapa anterior restaram 161 após da leitura completa de todo artigo;

Revisão da Literatura



Compondo um Questionário

- Das categorias identificadas foram selecionadas:
  - *Years*, *Education*, *Self-estimation* e *Size;*
- E excluídas:
  - *Pretest*, *Supervisor* e *Unspecified Questionnaire*;
- Para cada categoria foram elaboradas algumas questões;
- O objetivo era de descobrir quais questões de quais categorias tinham maior poder de predição;

Compondo um Questionário

Source	Question	Scale
Self estimation	On a scale from 1 to 10, how do you estimate your programming experience?	1: very inexperienced to 10: very experienced
	How do you estimate your programming experience compared to experts with 20 years of practical experience?	1: very inexperienced to 5: very experienced
	How do you estimate your programming experience compared to your class mates?	1: very inexperienced to 5: very experienced
	How experienced are you with the following languages: Java/C/Haskell/Prolog	1: very inexperienced to 5: very experienced
	How many additional languages do you know (medium experience or better)?	Integer
	How experienced are you with the following programming paradigms: functional/imperative/logical/object-oriented programming?	1: very inexperienced to 5: very experienced
Years	For how many years have you been programming?	Integer
	For how many years have you been programming for larger software projects, e.g., in a company?	Integer
Education	What year did you enroll at university?	Integer
	How many courses did you take in which you had to implement source code?	Integer
Size	How large were the professional projects typically?	NA, <900, 900-40000, >40000
Other	How old are you?	Integer

Questionário - Auto-avaliação

Source	Scale
Self estimation	1: very inexperienced to 10: very experienced

Question
On a scale from 1 to 10, how do you estimate your programming experience?
How do you estimate your programming experience compared to experts with 20 years of practical experience?
How do you estimate your programming experience compared to your class mates?

Avaliação Empírica

- Como avaliar um questionário de avaliação?
  - Recrutando programadores experientes e novatos e avaliar as respostas;
  - Ou …
  - Comparar as respostas dos questionários com tarefas que estão relacionadas à experiência em programação;

Avaliação Empírica

- Como avaliar um questionário de avaliação?
  - Recrutando programadores experientes e novatos e avaliar as respostas;
  - Ou …
  - **Comparar as respostas dos questionários com tarefas que estão relacionadas à experiência em programação;**
- Não é necessário diferentes grupos, somente um é suficiente;

O Experimento – Objetivo

- Como as questões podem relatar a experiência de programação;
- Medir quais participantes resolveram as tarefas corretamente e quanto tempo precisaram para completar cada tarefa;

O Experimento – Material

- 10 tarefas de compreensão de programas;
- Seleção de algoritmos presentes em livros introdutórios;
- Somente as duas últimas tarefas exigiam alto conhecimento;

```
1  public class Class1 {
2    public static void main(String[] args) {
3      int array[] = {14,5,7};
4      for (int counter1 = 0; counter1 < array.length;
           counter1++) {
5        for (int counter2 = counter1; counter2 > 0;
             counter2--) {
6          if (array[counter2 - 1] > array[counter2]) {
7            int variable1 = array[counter2];
8            array[counter2] = array[counter2 - 1];
9            array[counter2 - 1] = variable1;
10         }
11       }
12     }
13     for (int counter3 = 0; counter3 < array.length;
           counter3++)
14       System.out.println(array[counter3]);
15   }
16 }
```

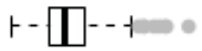
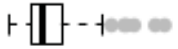
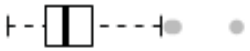
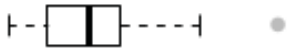
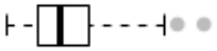
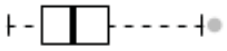
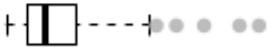
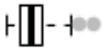
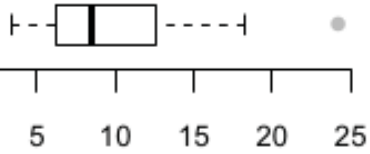
O Experimento – Participantes

- 128 participantes de 3 universidades da Alemanha;
- A todos foram oferecidos algum tipo de bônus (ex: pontos);
- Os dados foram armazenados anonimamente;

O Experimento – Execução

- Realização entre Janeiro e Abril de 2011;
- Tempo limite de 40 minutos para realização de todas tarefas;
- Quando o prazo acabou, foi permitido que os participantes terminassem a tarefa que estavam fazendo;

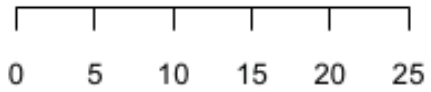
Resultados – Média e Desvio padrão

Variable	Response time Distribution	Mean	N	Correct
Task 1		4.44	124	70
Task 2		3.65	123	90
Task 3		5.02	121	97
Task 4		6.17	117	22
Task 5		4.06	118	46
Task 6		4.72	111	40
Task 7		2.34	92	31
Task 8		4.1	82	69
Task 9		1.94	78	11
Task 10		9.64	30	22

N: number of subjects who completed this task;
Correct: number of subjects with correct solution.

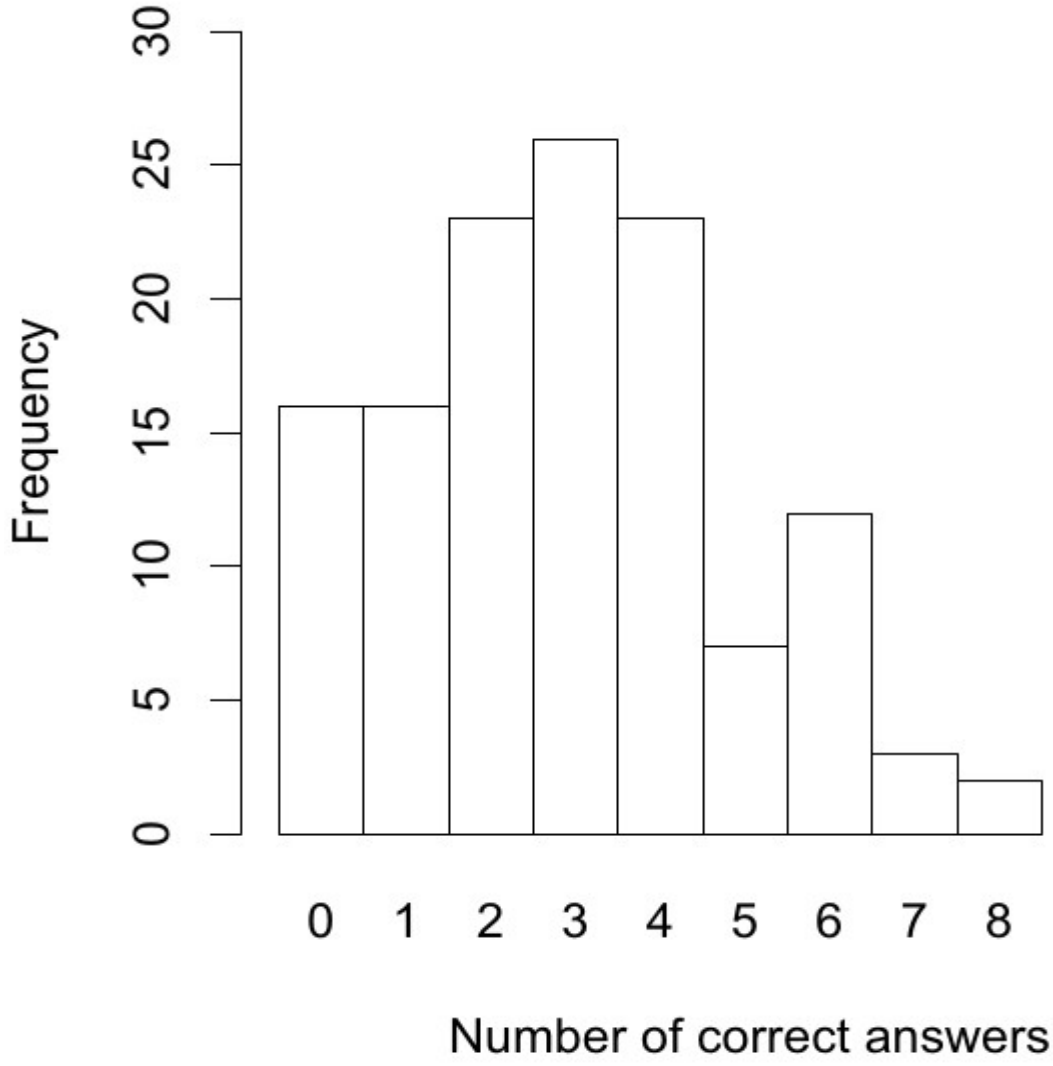
Resultados – Média e Desvio padrão

Variable	Response time Distribution	Mean	N	Correct
Task 1		4.44	124	70
Task 2		3.65	123	90
Task 3		5.02	121	97
Task 4		6.17	117	22
Task 5		4.06	118	46
Task 6		4.72	111	40
Task 7		2.34	92	31
Task 8		4.1	82	69
Task 9		1.94	78	11
Task 10		9.64	30	22

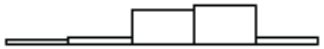
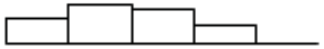
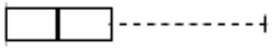
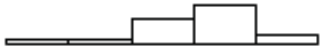


N: number of subjects who completed this task;
Correct: number of subjects with correct solution.

Resultados – Média e Desvio padrão



Resultados – Média e Desvio padrão

No.	Question	Distribution	N
1	s.PE		70
2	s.Experts		126
3	s.ClassMates		127
4	s.Java		124
5	s.C		127
6	s.Haskell		128
7	s.Prolog		128
8	s.NumLanguages	0  6	118
9	s.Functional		127
10	s.Imperative		128
11	s.Logical		126
12	s.ObjectOriented		127
13	y.Prog	0 25	123
14	y.ProgProf	0 7	127
15	e.Years	0 9	126
16	e.Courses	0 20	123
17	o.Size		128
18	o.Age	19 40	128

Análise Exploratória

Variable	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
s.C	.723				
s.ObjectOriented	.700			.403	
s.Imperative	.673	.333		.303	
s.Experts	.600	.326			
s.Java	.540		.427		
y.ProgProf		.859			
z.Size		.764			
s.NumLanguages	.335	.489		.403	
s.ClassMates		.449	.403	.424	
s.Functional			.880		
s.Haskell			.879		
e.Courses				.795	
e.Years			-.460	.573	
y.Prog		.493		.554	
s.Logical					.905
s.Prolog					.883

Ameaças à validade

- Tarefas representativas para alunos de graduação;
- Amostra dos participantes (alunos de graduação);
- Resultados da analise exploratória não podem ser generalizadas;

Recomendações

1. Combinar diferentes questões de diferentes categorias;

2. Utilizar questões de auto-avaliação em alunos de graduação;

3. Os cinco fatores apresentados podem servir de ponto de partida para o desenvolvimento de teorias em experiência de programação;

Conclusões

- Revisão da literatura sobre o estado da arte de medir e controlar a influência da experiência de programação;
- Um questionário que contém as questões mais comuns para medir a experiência de programação;
- Projeto experimental replicável para avaliar questionários;
- Proposta de cinco fatores modelos de experiência de programação.

Obrigado

hsborges@dcc.ufmg.br