

Zwei empirische Studien im Bereich des PSP

Sebastian Otto

Institut für Informatik

FU Berlin

10.5.2007

Teil1: Der Personal Software Process

Teil2: Vorstellung der Studien

Studie 1: Wesslén, 2000

Studie 2: Wohlin, 2004

Teil3: Zusammenfassung

Teil1: Der Personal Software Process (PSP)

“The Personal Software Process (PSP) is a self improvement process designed to help you control, manage, and improve the way you work.” (Humphrey 1995, S. 1)

Der Personal Software Process

- 7 Schritte (Humphrey, 1995):

PSP0: Wasserfallmodell, Messung von Aufwand und Defekten

PSP0.1: Einführung: Programmgrößenmessung, Zählstandards, Code Standards, Vorschläge zur Prozessverbesserung (PIP)

PSP1: Methoden zur Abschätzung der erwarteten Programmgröße, des Aufwandes

PSP1.1: Ressourcen- und Zeitabschätzung und –planung

PSP2: Design und Code-Reviews, Fehlerkontrolle

PSP2.1: Checklisten für Dokumentation und Design

PSP3: inkrementelle Entwicklung

Fragen?

Teil 2: Vorstellung der Studien

Studie 1: Wesslén 2000

1. Anliegen:

- Replikation einer Studie von Hayes und Over (1997), Überprüfung der Ergebnisse

2. Ort/Teilnehmer:

- Lund und Linköping Universität, Schweden
- Datenmaterial von 113 – 131 Teilnehmern

Table I. Summary of the results in Hayes and Over (1997). X indicates a significant result ($\alpha = 0.05$).

	PSP0 vs. PSP1	PSP1 vs. PSP2	Median improvement from PSP0 to PSP2
Size Estimation Accuracy	X	-	A factor of 2.5
Time Estimation Accuracy	X	-	A factor of 1.75
Overall Defect Density	X	-	A factor of 1.5
Compile Defect Density	X	X	A factor of 3.7
Test Defect Density	X	X	A factor of 2.5
Pre-Compile Defect Yield	-	X	50%
Productivity	-	X	No gain or loss

3. Durchführung:

→ 2 Graduiertenkurse an der Lund Universität

→ 1 Kurs für Doktoranden an der Linköping Universität

Datensätze: 131-113

4. Hypothesen:

- dreiteiliges Grundmuster:
 1. Aufstellen der Hypothese (Gesamtvergleich)
 2. Zusatzhypothese, die den Fortschritt zwischen PSP-Level berücksichtigt (paarweiser Vergleich)
 3. Nullhypothese: Negierung der beiden vorangegangenen Thesen

Beispiel:

5. Productivity

Hypothesis: The productivity gets higher for each PSP-Level

Additional hypothesis: The dispersion on the productivity gets smaller for each PSP level.

Null hypothesis: There is no difference in productivity between the PSP levels.

5. Auswertung der Ergebnisse:

„[T]his replication of the study in Hayes and Over (1997), has shown almost the same results as the original study[...]“ (Wesslén 2000, S.119)

	Results from Hayes and Over (1997)		Results from this study			
	Difference in mean		Difference in mean		Difference in dispersion	
	PSP0 vs. PSP1	PSP1 vs. PSP2	PSP0 vs. PSP1	PSP1 vs. PSP2	PSP0 vs. PSP1	PSP1 vs. PSP2
Size Estimation Accuracy	X	-	X	-	X	-
Effort Estimation Accuracy	X	-	-	-	-	X
Overall Defect Density	X	-	X	-	X	-
Compile Defect Density	X	X	X	X	X	-
Test Defect Density	X	X	X	X	X	-
Pre-Compile Defect Yield	-	X	-	X		
Productivity	-	X	-	X	X	X

Table IX. Median improvement from PSP0 to PSP2.

		Results from Hayes and Over (1997)	Results from this study
Size Estimation Accuracy	A factor of	2.5	2.1
Effort Estimation Accuracy	A factor of	1.75	3.0
Overall Defect Density	A factor of	1.5	1.4
Compile Defect Density	A factor of	3.7	2.9
Test Defect Density	A factor of	2.5	2.0
Pre-Compile Defect Yield		50%	39%
Productivity	A factor of	No gain or loss	0.9 (0.86)

- Parallelität bei Kursbegleiterscheinungen
 - Lerneffekte
 - Einführung von Checklisten und Formularen
 - „process overhead“
 - Homogenisierung der Leistungen der Studenten.

6. Bewertung der Ergebnisse

Fehlerquellen:

- 60 Prozent aller Fehler durch fehlerhafte Datensätze
1. zufällig
 2. lückenhafte Datensätze
 3. absichtlich

Maßnahmen:

1. Ausschluss aller lückenhaften Datensätze
2. Werteeinträge durch Studenten, Berechnung durch Mitarbeiter
3. Exel-Formulare für eine bessere Handhabung der Daten

Frage: Was ist mit den restlichen 40 Prozent?

7. Vergleichbarkeit:

Einschränkungen

- Teilnehmer 131 vs. 298
- Kursstärke 30-70 vs. 4-21
- Programmiersprachenwahl eingeschränkt

Fragen?

Teil 2: Vorstellung der Studien

Studie 2: Wohlin 2004

Teil2: Wohlin, 2004

1. Anliegen:

„Two questions are (1) whether prior knowledge of a specific language actually does improve software quality and (2) whether it is possible to capture performance using simple quantitative measures?“ (Wohlin 2004, S.211)

2. Ort/Teilnehmer:

- Blekinge Institute of Technologie, Ronneby, Schweden
- 65 Testpersonen, 6 ausgeschieden

Teil2: Wohlin, 2004

Unabhängige Variablen (Fragebogen):

- Programmiererfahrung

Abhängige Variablen (Messung):

- Zeit
- Größe
- Defekte
- Defekte/KLOC
- Produktivität
- Voraussagbarkeit der Größe
- Voraussagbarkeit der Zeit

Teil2: Wohlin, 2004

Table 1. Student characterization.

Area	Description
Study program (denoted program)	Answer: Computer Science and Engineering or Electrical Engineering
General knowledge in computer science and software engineering (denoted SE general)	<ol style="list-style-type: none">1. Little, but curious about the new course2. Not my speciality (focus on other subjects)3. Rather good, but not my main focus (one of a couple of areas)4. Main focus of my studies
General knowledge in programming (denoted Programming)	<ol style="list-style-type: none">1. Only 1–2 courses2. Three or more courses, no industrial experience3. A few courses and some industrial experience4. More than three courses and more than one year industrial experience
Knowledge about the PSP (denoted PSP)	<ol style="list-style-type: none">1. What is it?2. I have heard about it3. A general understanding of what it is4. I have read some material
Knowledge in C (denoted C)	<ol style="list-style-type: none">1. No prior knowledge2. Read a book or followed a course3. Some industrial experience (less than six months)4. Industrial experience
Knowledge in C++ (denoted C++)	<ol style="list-style-type: none">1. No prior knowledge2. Read a book or followed a course3. Some industrial experience (less than six months)4. Industrial experience
Number of courses (denoted Courses)	A list of courses was provided and the students were asked to put down a yes or no whether they had taken the course or not. Moreover, they were asked to complement the list of courses if they had read something else they thought was a particularly relevant course.

Table 2. Number of individuals giving a certain answer.

Independent variable	1	2	3	4
Study program	CSE: 32 and EE: 27			
SE general	1	14	22	22
Programming	7	28	23	1
PSP	18	30	10	1
C	32	19	6	2
C++	37	16	4	2
Courses	2: 9; 3: 6; 4: 6; 5: 7; 6: 6; 7: 11; 8: 7; 9: 3; and 10: 4. ^a			

^aInterpretation: 2: 9 means that nine students had read two courses and so forth.

Table 4. Relations between individuals.

Dependent variables	Max/min	P90/P10	Q3/Q1
Size	3.06	1.78	1.38
Time	4.62	2.57	1.65
Defects	19.15	3.46	1.81
Defects/KLOC	17.09	2.78	1.69
Productivity	5.90	3.10	2.00
Predictability size	5.49	2.79	1.83
Predictability time	5.42	2.76	1.75

Table 5. Factor analysis of the performance measures.

Dependent variable	Factor 1	Factor 2	Factor 3
Time	0.676	-0.527	-0.131
Defects	0.963	0.139	-0.024
Defects/KLOC	0.919	-0.068	0.008
Size	0.382	0.736	-0.142
Productivity	-0.301	0.925	0.037
Predictability size	-0.098	0.060	0.776
Predictability time	0.045	-0.095	0.774

Table 6. The p -values for independent variables when evaluated versus the dependent variables.

p -value	Program	SE	Courses	Prog	C	C++	PSP
Time	0.098	0.034	0.394	0.173	0.101	0.318	0.022
Defects	0.602	0.335	0.260	0.701	0.768	0.961	0.928
Defects/KLOC	0.832	0.458	0.325	0.845	0.724	0.899	0.840
Size	0.047	0.032	0.365	0.429	0.903	0.947	0.932
Prod.	0.002	0.001	0.020	0.035	0.111	0.110	0.465
Pred. size	0.015	0.138	0.243	0.243	0.166	0.183	0.507
Pred. time	0.256	0.750	0.695	0.756	0.271	0.272	0.807

Teil2: Wohlin, 2004

3. Validität:

1. Beschränkung auf C, C++
2. Keine Zufallsstichprobe
3. Ausschluss eines Teilnehmers
4. Wirkung des PSP-Kurses
5. Signifikanztest, aber keine Effektstärken
6. „fishing for significance“

Teil2: Wohlin, 2004

4. Schlussfolgerung:

„Two questions are (1) whether prior knowledge of a specific language actually does improve software quality and (2) whether it is possible to capture performance using simple quantitative measures?“ (Wohlin 2004, S.211)

Antwort 1:

- Paradigmen der Programmierung wichtiger als Beherrschung einer Programmiersprache

Antwort 2:

- Messung möglich, aber da ist noch mehr!

Teil2: Wohlin, 2004

5. Bewertung der Ergebnisse:

- unerwartete, interessante Ergebnisse
- zahlreiche Mängel: keine Zufallsstichprobe, Beschränkung der Programmiersprache, Fehlen von Hypothesen, Ausschluss eines Teilnehmers
- Andererseits wäre zu fragen: Wie steht es um den Ausbildungsstand der Teilnehmer beziehungsweise um die Qualität der Lehre?
- Adäquatheit des Fragebogens

Fragen?

Teil 3: Zusammenfassung

1. Wesslén:

- Replikation innerhalb des vorgegebenen Rahmens
- positiv: gesonderter Abschnitt zur Validität

2. Wohlin:

- ambitioniert, jedoch hohe Intransparenz
 - zahlreiche Mängel
- Ergebnisse wenig vertrauenserweckend

Vielen Dank!