

PSP

Experiment über den Nutzen von Kursen über den
“persönlichen Softwareprozess”

Inhalt

- Über PSP?
- Experiment
 - Motivation
 - Beschreibung
 - Auswertung
- Bewertung des Experiments

Über PSP

Über den
“persönlichen Softwareprozess”

Was ist PSP?

- Verfahren, um die eigenen “Fähigkeiten” zu verbessern
- Terminplanung
- Fehlervermeidung



Watts Humphrey

Arbeitsweise ohne PSP

- Termine werden nicht eingehalten
 - Übliche Schätzungen zu optimistisch
 - Fehler werden bei Planung nicht berücksichtigt
 - Fastregeln helfen auch nicht weiter ("Schätzen, Zeit verdoppeln")

Arbeitsweise ohne PSP

- Fehler werden erst sehr spät erkannt
 - Je später Fehler erkannt, desto schwieriger zu korrigieren
- Gut zu wissen: Wo macht man häufig Fehler?
- Fehler häufig erst bei Auslieferung erkannt
- Zeitdruck: geringere Sorgfalt

Arbeitsweise ohne PSP

- Fazit
 - Intuitiv sammelt man zu wenige Informationen, um Dauer künftiger Projekte gut schätzen zu können
 - Man versäumt, sich häufige Fehlerquellen bewusst zu machen

Arbeitsweise mit PSP

- Idee von PSP
 - Das geht aber besser!
 - Daten über seinen Arbeitsstil sammeln
 - Daten regelmäßig auswerten
 - historische Daten

Arbeitsweise mit PSP

- “Logbuch” erstellen:
 - Erfasse minutengenau, was du wann gemacht hast
 - Arbeitsphasen angeben
 - Beginn und Ende einer Programmeinheit (mit LOC)
 - Nebenbei ausgeführte Tätigkeiten
 - Fehler gefunden
 - Fehler beseitigt
- Individuelle Schwerpunkte

1998-08-11 09:49:53 ecd

1998-08-11 09:49:54 bcd

1998-08-11 09:59:04 ecd

Kaffee holen gewesen

1998-08-11 10:14:32 bte

Beispiel für “Logbuch”


```
1998-08-11 10:17:05    be
```

```
1998-08-11 10:22:19    ee cd 1 F
```

```
read_input.c:112
```

Zweiter Index von substring()
zeigt hinter letzten Buchstaben!

Beispiel für “Logbuch”

Arbeitsweise mit PSP

- Auswerten des “Logbuchs”
 - Programmeinheiten nach Typ kategorisieren
 - Dauer für Programmeinheit nach Typ und Größe bestimmen
 - Häufige Fehler bestimmen
 - Fehler mit zeitaufwendigem Debugging bestimmen
 - Arbeitsphasen der Fehler bestimmen

Klasse	Zahl	%	Dauer	Ø
A	21	81	132	6,3
F	4	15	71	17,8
W	1	4	18	18,1
	26	100	221	8,5

Beispiel für Auswertung

Arbeitsweise mit PSP

- Werkzeuge
 - Editor mit Zeitstempelfunktionalität
 - z.B. UltraEdit (F7), Emacs
- Skripte zur Auswertung

Arbeitsweise mit PSP

- Terminplanung für künftige Projekte
 - Größe des Projektes schätzen und grob einteilen
 - Größe der Programmeinheiten schätzen
 - Benötigte Zeit aus der Auswertung des “Logbuchs” folgern

Arbeitsweise mit PSP

- Fehlervermeidung
 - Durch Einsichten aus der Analyse
 - Intensivere Reflexion der eigenen Tätigkeit
 - Checklisten für künftige Codedurchsicht

Fragen?

Experiment

Experiment über den Nutzen von Kursen über den
“persönlichen Softwareprozess”

Motivation

- Was will man untersuchen?
 - Werden Erkenntnisse aus den PSP-Kursen in der Praxis genutzt?
- Arbeitsweise der Versuchspersonen:
 - Zuverlässigkeit
 - Zeitschätzung
 - Produktivität (Randbetrachtung)

Motivation

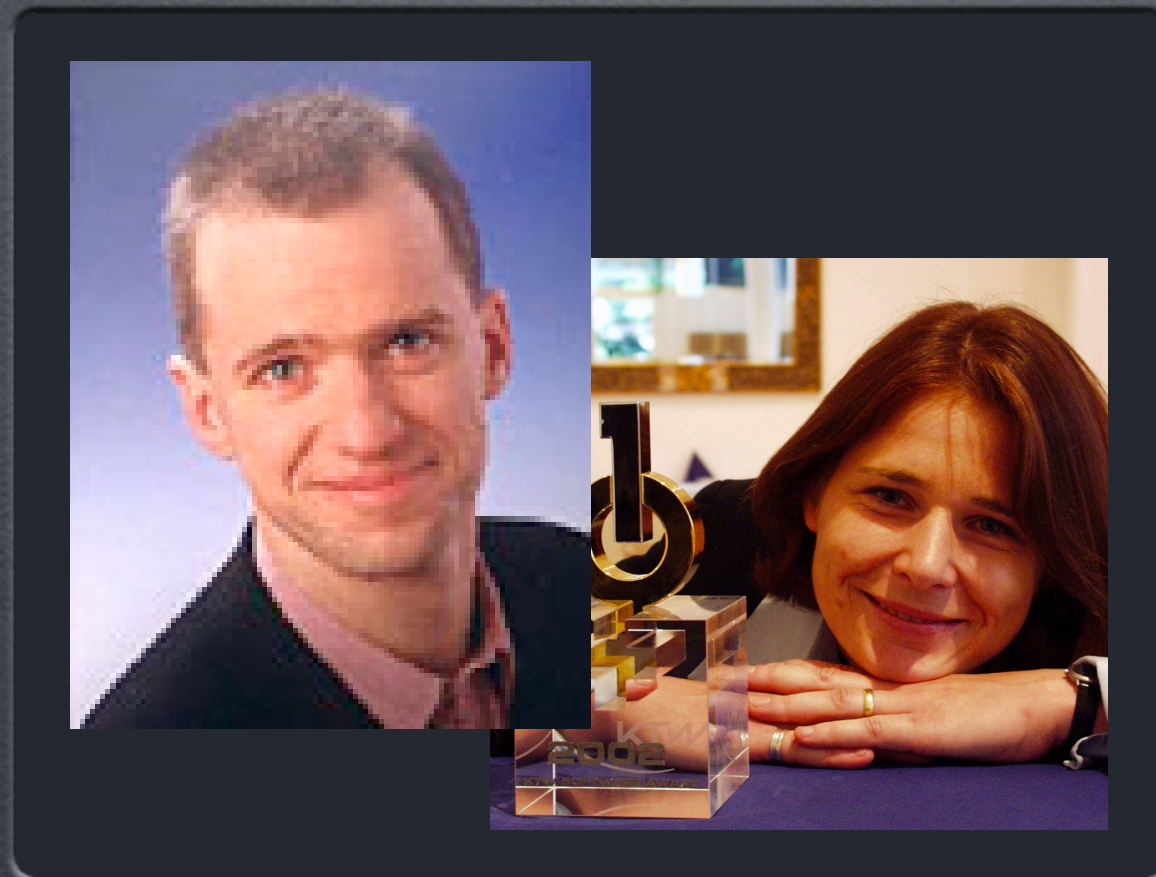
- Frühere Untersuchungen:
 - Watts Humphrey
 - mehrere Untersuchungen (pro PSP)
- Industrie
 - Ansätze, die für PSP sprechen

Motivation

- Nachteile früherer Untersuchungen:
 - Hawthorne-Effekt
 - einfache Aufgaben (Humphrey)
 - unüberschaubares Umfeld (Industrie)
 - keine Kontrollgruppen

Beschreibung

- Überblick
 - Durchführung zwischen Februar 1997 und Oktober 1998
 - Experimententwurf
 - Zwei Gruppen
 - Extra-Subjekt-Design
 - Programm entwickeln
(mit und ohne PSP-Vorkenntnisse)
 - Ist die Zuverlässigkeit, Zeitschätzung und Produktivität der Gruppe mit PSP besser?



Lutz Prechelt
Barbara Unger

Beschreibung

- Aufgabe
 - Programm entwickeln: “phoneword”
 - Telefonnummern zu einer Menge von Wortfolgen zuordnen
 - Beliebige Programmiersprache
 - Hauptkriterium: Zuverlässigkeit

Beschreibung

- Aufgabe
 - Keine einfache Zahl, sondern /, -, 0, ... 9
 - Maximale Länge der Telefonnummern: 50
 - Wörterbuch: maximal 75.000 Einträge
 - Wörterbuch mit 73.113 Einträgen stand als Textdatei zur Verfügung

3586-75: Dali um
3586-75: Sao 6 um
3586-75: da Pik 5

0	E
1	J N Q
2	R W X
3	D S Y
4	F T
5	A M
6	C I V
7	B K U
8	L O P
9	G H Z

Beschreibung

- Versuchspersonen
 - 48 männliche Informatik-Studenten
 - 600h Programmiererfahrung
 - 20.000 Zeilen Quellcode (außer Uni)

Beschreibung

- Versuchgruppe
 - 29 Personen zu Beginn
 - 5 Personen haben abgebrochen
 - 15-wöchiger PSP-Kurs als Vorbereitung
- Kontrollgruppe
 - 19 Personen zu Beginn
 - 3 Personen abgebrochen
 - 6-wöchiger kompakter KOJAK-Kurs

Beschreibung

- Durchführung
 - Beide Gruppen wurden genau gleich behandelt
 - keine spezielle Aufforderung PSP zu nutzen
 - Unterlagen in Papierform

Beschreibung

- Durchführung
 - Persönlicher Fragebogen
 - Aufgabenbeschreibung
 - Befragung nach Aufwandsschätzung
 - Lösen der Aufgabe an speziellen UNIX-Rechnern mit präparierter Arbeitsumgebung

Beschreibung

- Durchführung
 - Versuchspersonen wurden im Hintergrund “ausspioniert”
 - Protokollierung aller kompilierten Versionen mit Zeitstempel sowie Gesamtarbeitszeit

Beschreibung

- Durchführung
 - Versuchspersonen konnten Akzeptanztest beantragen

Beschreibung

- Akzeptanztest
 - Vergleich einer zufälligen Auswahl von 500 Nummern mit Referenzimplementierung
 - Anderes Wörterbuch

Beschreibung

- Akzeptanztest
 - Definition der Zuverlässigkeit:
 - Anzahl der richtigen Lösungen durch Anzahl aller Lösungen (richtige und falsche)
 - Bestanden: 95%
 - 50 DM für erfolgreiche Teilnahme
jeweils 10 DM weniger für jeden nicht
erfolgreichen Akzeptanztest

Beschreibung

- Referenzimplementierung
 - Entwickelt von Prechelt und Unger
 - Stufenweise Verfeinerung mit halbformaler Verifikation
 - Niemals Fehler gefunden

Fragen?

Gültigkeit

- Innere Gültigkeit
 - Einteilung in die Gruppen nicht zufällig
 - Teilnehmer haben sich indirekt selbst eingeteilt
- Wahl der Programmiersprache
 - Weder Objektorientierung noch Speicherverwaltung waren für die Aufgaben besonders wichtig

Gültigkeit

- Äußere Gültigkeit
 - Unterschiedliche Arbeitsbedingungen
 - Zeitspanne zwischen PSP-Kurs und Experiment sehr gering
 - Zu “verschulte” Aufgabenstellungen
 - Teilnehmer mit wenig praktischer Erfahrung

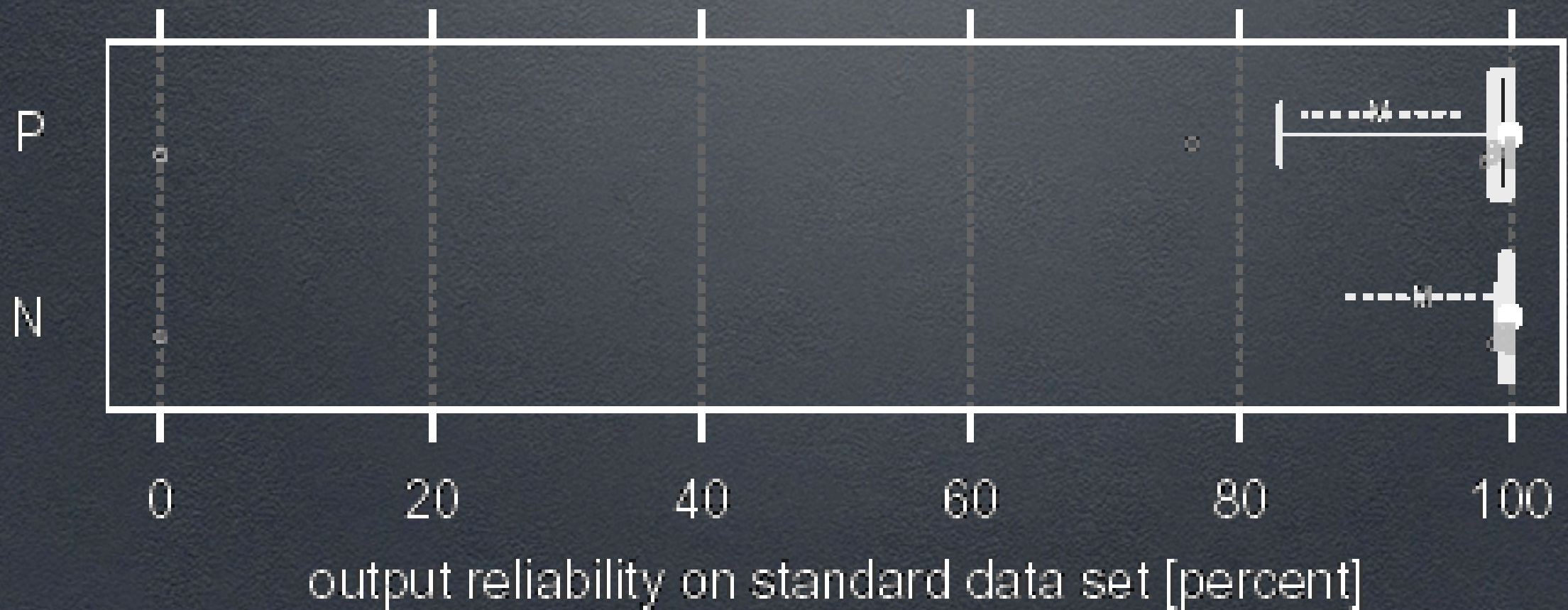
Auswertung

- Hypothese
 - Teilnehmer, die den PSP-Kurs besucht haben, ...
 1. erstellen zuverlässigere Programme.
 2. schätzen die benötigte Arbeitszeit realistischer.
 3. arbeiten schneller (Randbetrachtung).

Auswertung

- Zuverlässigkeit:
 - Messung härter als Akzeptanztest
 - Generierung von Mengen von Telefonnummern mit /, -, 0, 1, ... 9
 - Jede Menge hatte 100, 1.000, 10.000 oder 100.000 Telefonnummern
 - kritische Längen von Telefonnummern (1 und 50) gleich wahrscheinlich

Zuverlässigkeit

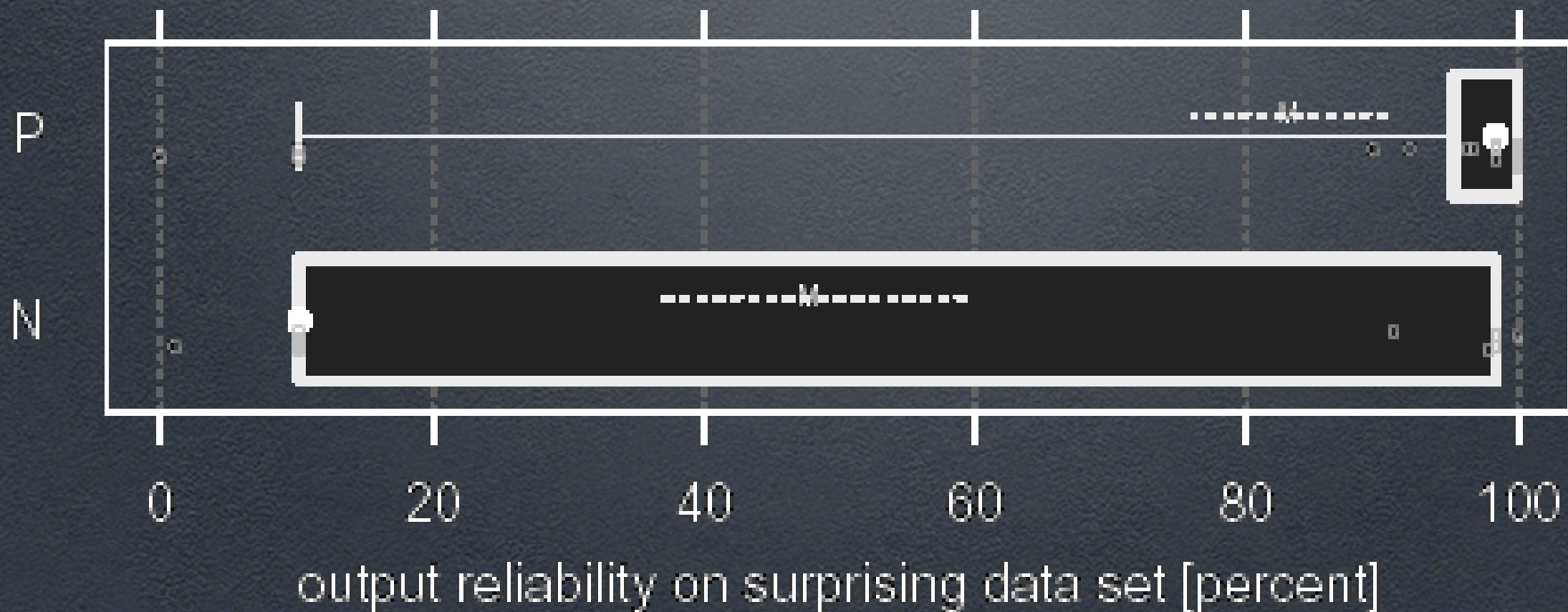


Auswertung

- Zuverlässigkeit:
 - “überraschende” Telefonnummern:
 - nur / und -, aber keine Ziffern

Zuverlässigkeit

“überraschende” Telefonnummern



Auswertung

- Zuverlässigkeit:
 - Schlussfolgerung:
 - Im Allgemeinen nicht zuverlässiger, aber bei den Spezialfällen schon.
 - Hypothese wird unterstützt, aber nicht bewiesen.

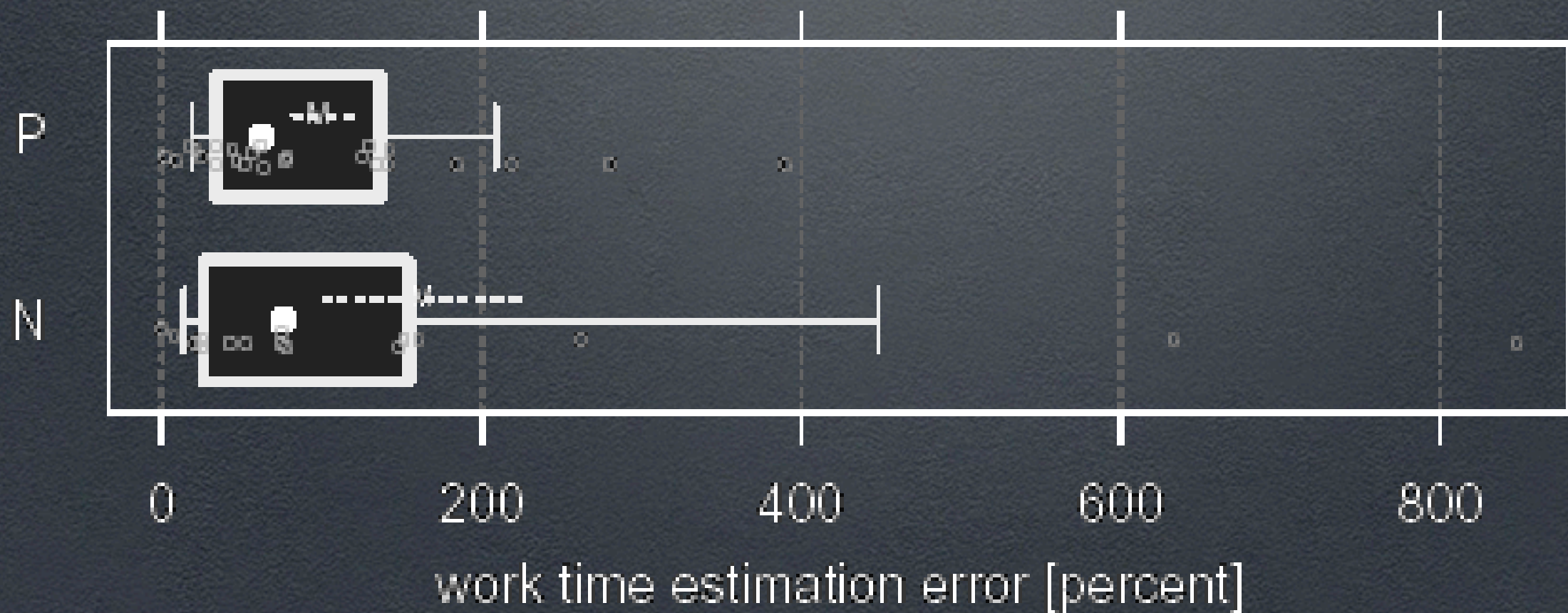
Auswertung

- Zeitschätzung:
 - Abweichung von tatsächlicher Arbeitszeit durch geschätzte Arbeitszeit
 - Median nahezu identisch
 - Teilnehmer ohne PSP haben stärkere Abweichler

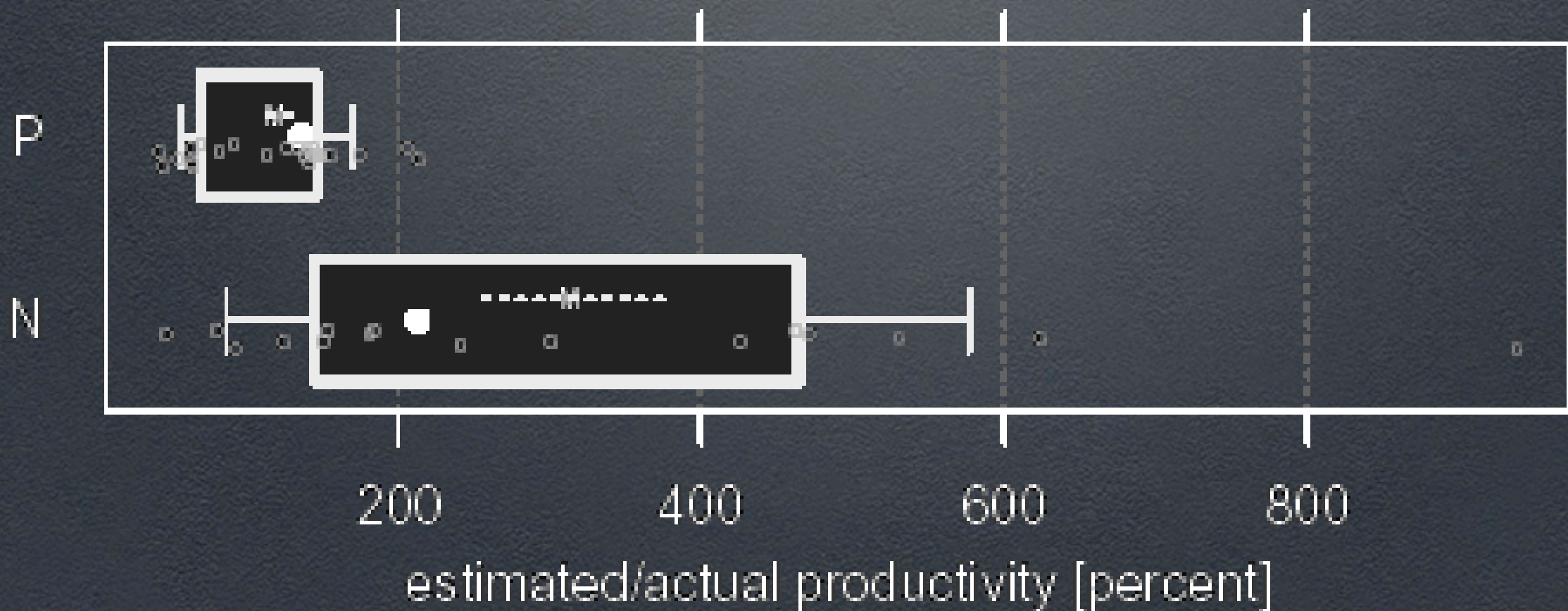
Auswertung

- Zeitschätzung:
 - PSP-Gruppe hat die *Größe* des Programms besser geschätzt, aber dadurch keine bessere Zeitschätzung erhalten

Zeitschätzung



Größenschätzung



Auswertung

- Zeitschätzung:
 - Schlussfolgerung:
 - Hypothese wird nicht unterstützt.

Auswertung

- Produktivität (Randbetrachtung):
 - Anzahl der gelösten Aufgaben pro Zeiteinheit:
 - Produktivität der PSP-Gruppe ist etwas geringer
 - PSP-Gruppe schreibt längere Programme (mehr Fehlerbehandlung)

Produktivität

Aufgaben pro Zeiteinheit

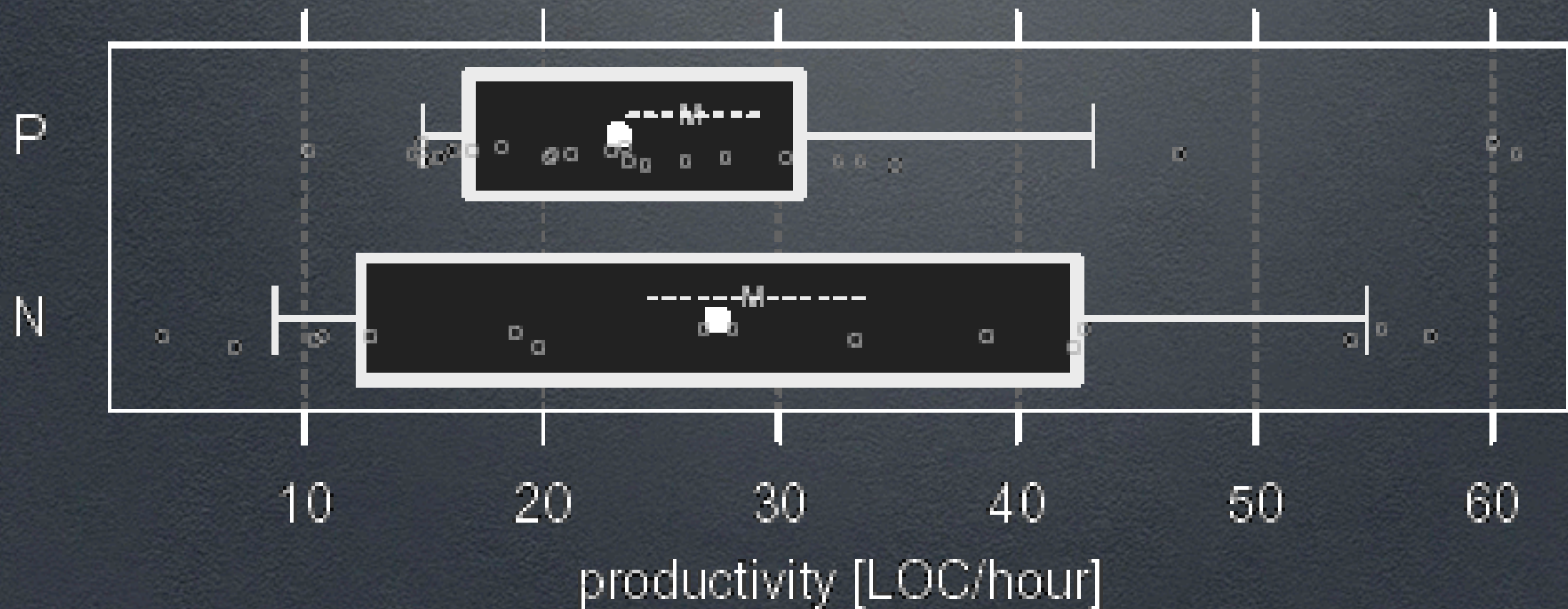


Auswertung

- Produktivität (Randbetrachtung):
 - LOC pro Zeiteinheit:
 - deutlich geringere Unterschiede

Produktivität

LOC pro Zeiteinheit



Auswertung

- Produktivität (Randbetrachtung):
 - Gesamtarbeitszeit:
 - PSP-Gruppe hat höheren Median
 - PSP-Gruppe hat deutlichen niedrigeren Durchschnitt

Produktivität

Gesamtarbeitszeit



Auswertung

- Produktivität (Randbetrachtung):
 - Schlussfolgerung:
 - Hypothese wird nicht unterstützt.

Auswertung

- Varianz:
 - Zusätzliches Ergebnis
 - Alle Auswertungen liegen bei der PSP-Gruppe näher beieinander
 - Geringere Varianz kann bei Teamwork von Vorteil sein

Auswertung

- Nutzung von PSP:
 - Nur bei 6 Teilnehmern der PSP-Gruppe konnten Beweise gefunden werden, dass sie PSP genutzt haben
 - Unterschiede: Grad von PSP-Nutzung, nicht Nützlichkeit von PSP

Auswertung

- Nutzung von PSP:
 - Quote der “Abbrecher” bei PSP-Nutzern hoch
 - Schwächere Teilnehmer benutzen wahrscheinlich PSP, weil sie der Meinung sind, dass PSP ihnen hilft.

Auswertung

- Weitere Ergebnisse:
 - PSP-Gruppe benötigte weniger Zeit um Fehler zu beseitigen.
 - PSP-Gruppe machte weniger triviale Fehler (Compilerfehler).
 - PSP-Gruppe hat mehr kommentiert.

Auswertung

- Schlussfolgerung
 - Enttäuschend
 - Gründe:
 - Verbesserung durch PSP überschätzt (Hawthorne)
 - PSP wurde nicht genutzt

Auswertung

- Schlussfolgerung
 - Gründe, wieso PSP so wenig genutzt worden ist:
 - unterschiedlicher Charakter
 - Projekt zu klein
 - keine Aufforderung PSP zu nutzen

Auswertung

- Schlussfolgerung
 - PSP ist trotzdem erstrebenswert.
 - PSP-Kurse allein reichen nicht aus.
 - Experiment unterstützt diese Hypothese.

Auswertung

- Schlussfolgerung
 - Man muss besser verstehen, wie man die Leute dazu kriegen kann, auch wirklich PSP zu nutzen.

Fragen?

Bewertung

Bewertung des Experiments

Innere Gültigkeit

- Diskussionswürdig:
 - “Einteilung” der Teilnehmer durch Kurswahl
 - unterschiedliche Programmiersprachen
 - einerseits: PSP unabhängig von Sprachen
 - andererseits: Vergleichbarkeit

Äußere Gültigkeit

- PSP-Gruppe hatte zu wenig Erfahrung mit PSP:
 - positive Ergebnisse sprechen um so mehr für PSP-Kurse
 - negative Ergebnisse lassen keine sinnvolle Interpretation zu

Äußere Gültigkeit

- Aufforderung PSP zu nutzen oder nicht?
 - einerseits: wesentlicher Bestandteil
 - andererseits: Ungleichbehandlung

Glaubwürdigkeit

- Zuverlässigkeit:
 - künstliche und sehr spezielle Messung für die Zuverlässigkeit
 - PSP-Kurs schärfte Blick für Ausnahmen
 - bessere Ergebnisse nicht aufgrund von PSP
- Kritik am PSP-Kurs lassen Autoren nicht gelten

Vorschlag

- Teilung der Hypothesen:
 - Führen PSP-Kurse dazu, dass PSP genutzt wird?
 - Entwickeln PSP-Programmierer zuverlässigere Software und schätzen die Zeit besser?

Vorschlag

- Nutzung von PSP:
 - Umfrage: “Nutzt du PSP?”
 - ca. 1 Jahr nach PSP-Kurs

Vorschlag

- Nützlichkeit von PSP:
 - Experiment
 - Problem: Versuchs- und Kontrollgruppe zu bilden

thanXX ;-)

Diskussion