

# Der Personal Software Process

... in einer didaktischen Betrachtung

Frank@Schlesinger.com

In „Ausgewählte Beiträge zum Software Engineering“

2005

---

1. PSP – ganz kurz
2. PSP Studien
  - PSP verbessert Produkte und Projekte!
3. PSP in der Praxis
  - Gründe für die Bedeutungslosigkeit
  - Unterstützung für den PSP
4. PSP – eine didaktische Analyse
  - Analyse der mit PSP zu erreichenden Lernziele
  - Einfluss von Tools und Hackystat auf die Lernziele
5. Fazit
6. Referenzen

# 1. PSP - ganz kurz

---

- 1995: PSP „Personal Software Process“ von Watts S. Humphrey, Software Engineering Institute, Carnegie Mellon University veröffentlicht in *[Humphrey 95]*
- Sammlung von Methoden, Techniken und Standards zur Verbesserung des individuellen Softwareprozesses
- Ziel: Steigerung der Produktqualität, Verbesserung der Projektplanung

---

[Humphrey 95] W. S. Humphrey. A Discipline for Software Engineering. Reading, MA. Addison-Wesley 1995

# 1. PSP - ganz kurz

---

- PSP bietet Formulare zur Datenerfassung: Arbeitszeiten, Unterbrechungen, Defekte, ...
  - Standardisierte Auswertungen dieser Daten
  - Ergebnisse fließen in den Prozess ein (Reviews mit Prüflisten u.a.)
-

# 1. PSP - ganz kurz

---

- PSP wird durch den Entwickler personalisiert.
  - Jeder bekommt im Ergebnis einen individuellen Softwareprozess
  - Vermittlung von PSP in Kursen (z.B. 15 Wochen, 1 Tag/Woche) oder durch Selbststudium
  - Lernen in mehreren Stufen möglich
-

## 2. PSP Studien

---

- In PSP Kursen wurden Studien zur Messung der „Wirksamkeit“ durchgeführt
- Studien des SEIs (*[Humphrey 96]*, *[Ferguson 97]* und *[Hayes 97]*) und externe Studien (*[Weslén 2000]*)
- Alle Studien belegen die Wirksamkeit von PSP !

---

[Humphrey 1996] W. S. Humphrey. Using a defined and measured personal software. IEEE Software May 1996. S. 77f.

[Ferguson 1997] P. Ferguson, W. S. Humphrey, S. Khajenoori, et al. Results of applying the personal software process. IEEE Computer 30(5) 1996. S 24f.

[Hayes 1997] W. Hayes, J. W. Over. The Personal Software Process (PSP): An Empirical Study of the Impact of the PSP on Individual Engineers. Technical Report CMU/SEI-97-TR-001, ESC-TR-97-001. Software Engineering Institute. December 1997.

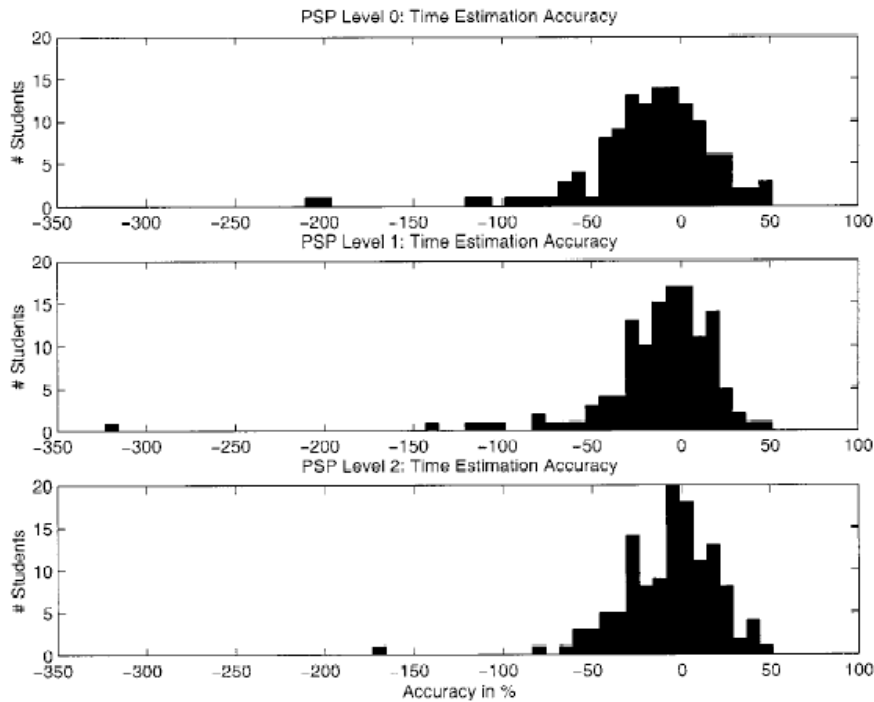
[Weslén 2000] A. Weslén. A Replicated Empirical Study of the Impact of the Methods in the PSP on Individual Engineers. Empirical Software Engineering May 2000. S. 93f.

## 2.1 PSP Studien - Ergebnisse

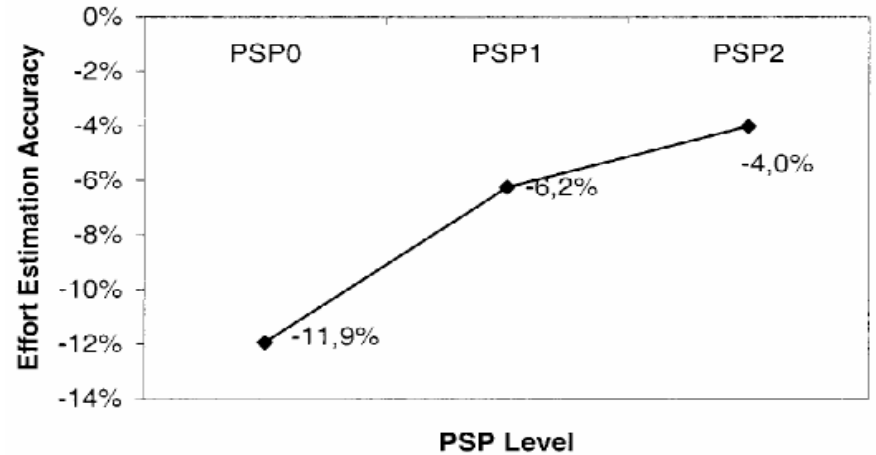
---

- Während eines PSP Kurses
  - Verbessert sich die Produktqualität (Defektdichte sinkt)
  - Verbessern sich die Abschätzungen der Teilnehmer (bzgl. Zeiten und Größen)

# 2.2 PSP Studien – Genauigkeit der Zeitabschätzungen



Entnommen: [Weslén 2000]



Entnommen: [Weslén 2000]



### 3. PSP in der Praxis

---

- *[Johnson 2002]* : Wenig Grund zu der Annahme, dass PSP eingesetzt wird
- Erfahrungsbericht *[Borstler 2002]*: 72 von 78 Kursteilnehmern wollen PSP nicht weiter benutzen. Über die restlichen 6 ist nicht bekannt, ob sie es einsetzen

---

[Johnson 2002] Johnson, Kou, Agustin et al. Beyond the PSP: Metrics collection and analysis for the differently disciplined. Proceedings of the 2003 International Conference on Software Engineering. Portland, Oregon. May 2003

[Borstler 2002] J. Borstler, D. Carrington, G. Hislop, et al. Teaching PSP: Challenges and lessons learned. IEEE Software, 19(5), September 2002.

## 3.1 PSP in der Praxis - Wo ist das Problem ?

---

- *[Prechelt 97]*: PSP erfordert viel Selbstdisziplin. Viele bringen diese nicht auf.
- *[Johnson 2002]*: Großer „overhead“ und „context-switch“
- *[Prenzel 93]*: Zur Motivation: „Maßgeblich ist die von der Person erlebte Autonomie ... “
- Management: Qualifizierung für PSP ist aufwändig und teuer

---

[Prechelt 1997] Experience Report: Teaching and Using the Personal Software Process (PSP). Submission to ESEC 1997. Online unter URL: [http://page.inf.fu-berlin.de/~prechelt/Biblio/psp\\_experience.pdf](http://page.inf.fu-berlin.de/~prechelt/Biblio/psp_experience.pdf)

[Prenzel 93] M. Prenzel. Autonomie und Motivation im Lernen Erwachsener. In: Zeitschrift für Pädagogik 1993, 39. Jg., H. 2, S. 239f.

## 3.2 PSP in der Praxis - Vorschläge

---

- *[Prechelt 97]*: Zusammenarbeit mit PSP-Kollegen und Einsatz von Tools helfen
  - *[Johnson 02]*: Tools („Leap“) steigern die Akzeptanz etwas
  - Im Stress werden sie aber auch nicht benutzt: „You can't even ask them to push a button“ ist Titel und Fazit von *[Johnson 01]*
-

## 3.3 PSP in der Praxis - Hackystat

---

- ... ist ein vollautomatisierten Datenerfassungssystem.
  - Sensoren zeichnen Daten in den Tools des Programmierers (und ohne sein Zutun) auf.
  - Daten werden zentral erfasst (Hackystat Server) und dort ausgewertet und auf Webseiten angezeigt
-

## 3.3 PSP in der Praxis - Hackystat

---

- ... misst nicht genau den PSP
  - Die Natur der Daten ist eine Andere.
  - *[Johnson 02]*: Hackystat ist „*beyond*“ the PSP.
  - Es ist „in-process“ und „non-disruptive“ und somit eine Verbesserung zum Standard-PSP
  - Johnson suggeriert Hackystat sei Weiterentwicklung des PSPs
-

## 4. PSP - eine didaktische Analyse

---

- Didaktik – die Wissenschaft vom Lehren und Lernen
- Es gibt eine Vielfalt an didaktischen Konzeptionen und Theorien
- Hier: „Berliner Didaktik“ (Lehr- und Lerntheoretische Didaktik) nach Heimann, Otto, Schulz [*Heimann 76*] und [*Meyer 94*]

---

[Heimann 76] P. Heimann. Didaktische Grundbegriffe. In: Reich/Thomas (Hrsg.) 1976. S. 103f.

[Meyer 94] W. Jank, H. Meyer. Didaktische Modelle. Berlin. Cornelsen Scriptor. 3. Aufl. 1994

## 4. PSP - eine didaktische Analyse

---

- Die „Berliner Didaktik“ ist empirisch-analytisch entstanden
  - Sie ist frei von ideologischen „Ballast“
  - Sie kann benutzt werden um jegliche Lehr- und Lernform zu analysieren
  - Sie kann benutzt werden um Unterricht zu planen
-

# 4.1 Lernziele der „Berliner Didaktik“

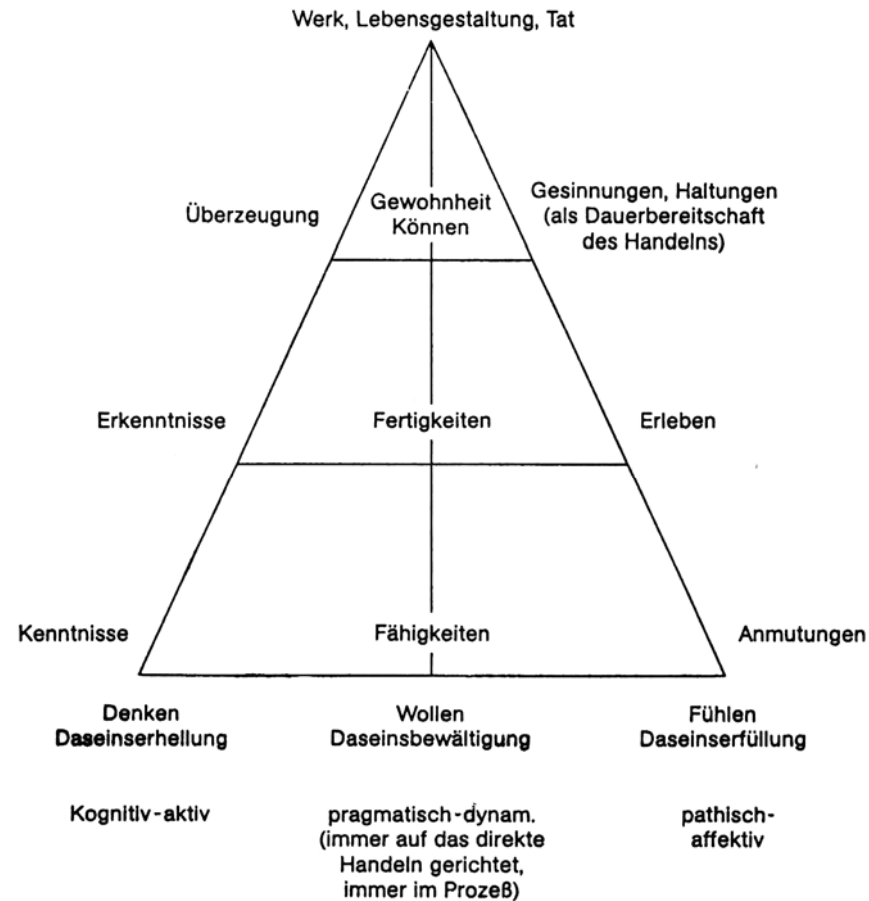


Abbildung 6.6 (aus: Heimann 1976 a, S. 125)

Entnommen: [Meyer 94]



## 4.2 Lernziele des PSPs

### • Kognitiv-aktive Dim.:

- Lernende (L) bekommen „Kenntnis“ über ihr Vorgehen (Arbeitszeiten, Unterbrechungen, Zeitaufwand für Reviews, ...)
- L bekommen „Kenntnisse“ über die Produkte (Anzahl und Art der Defekte, Zeit für deren Beseitigung, Größe, ...)



Abbildung 6.6 (aus: Heimann 1976 a, S. 125)

## 4.2 Lernziele des PSPs

### • Kognitiv-aktive Dim.:

- „Kenntnisse“ können sich in „Erkenntnissen“ niederschlagen:
  - „Ich werde zu oft unterbrochen“
  - „Ich arbeite mehr als 60 Stunden in der Woche“
  - „Ich mache zu viele Defekte (dieser Art)“
  - „Meine Software ist meist doppelt so groß wie erwartet“
  - „Ich brauche viel länger als geplant“

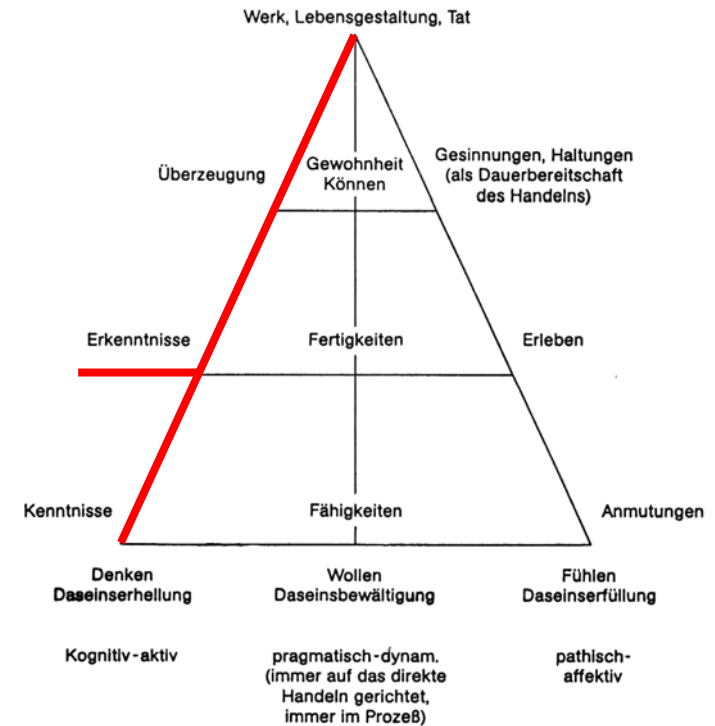


Abbildung 6.6 (aus: Heimann 1976 a, S. 125)

## 4.2 Lernziele des PSPs

- Pragmatisch-dynamische Dim.
  - PSP „befähigt“ die L konkrete Arbeitstechniken zu benutzen
    - Erfassung von Zeiten, Unterbrechungen, Defekten
    - Erstellung von Prüflisten
    - Durchführen von Reviews
    - Einhaltung von Codekonventionen
    - Zeit- und Aufwandsabschätzungen

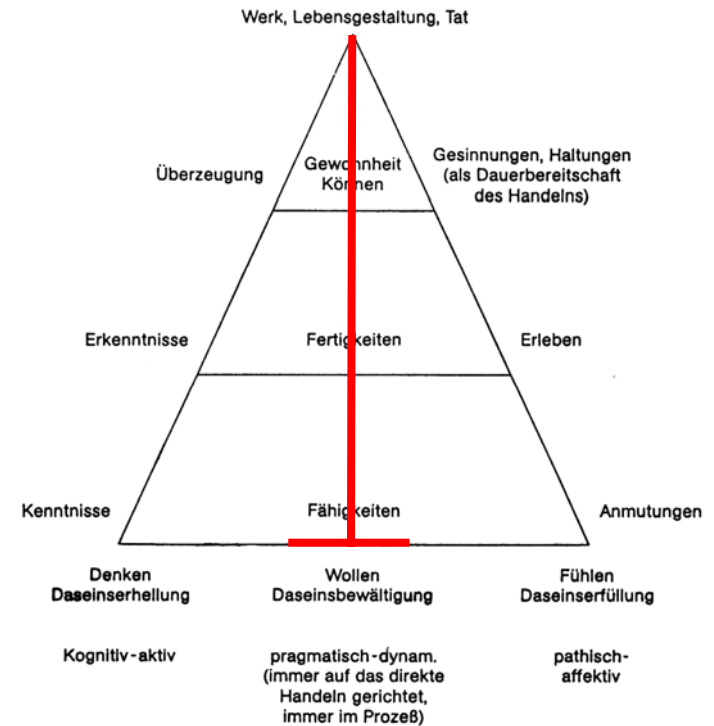


Abbildung 6.6 (aus: Heimann 1976 a, S. 125)

## 4.2 Lernziele des PSPs

- Pragmatisch-dynamische Dim.

- Durch das Einüben werden diese Techniken zu „Fertigkeiten“ ausgebildet
- Durch PSP „gewöhnt“ sich der Entwickler an:
  - Persönliches Zeitmanagement
  - Persönliches Qualitätsmanagement
  - Struktur
  - Disziplin

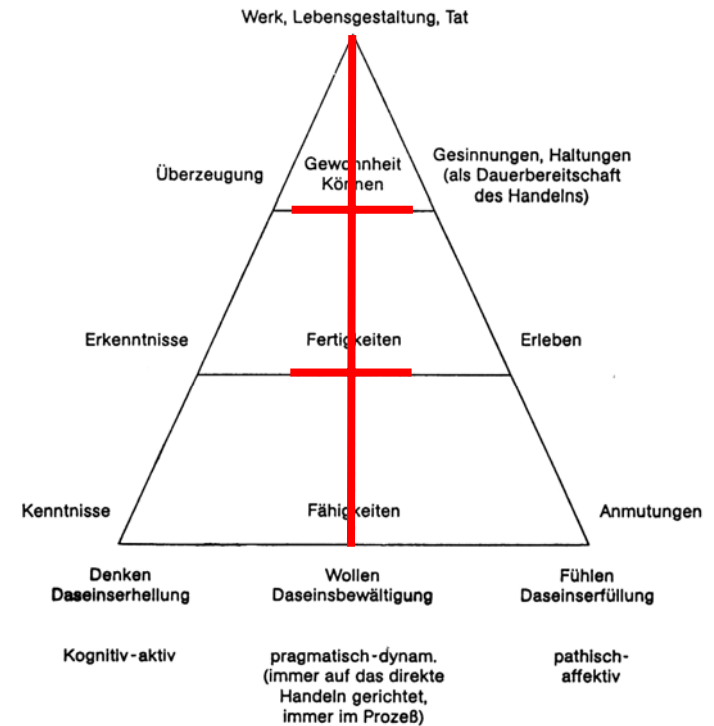


Abbildung 6.6 (aus: Heimann 1976 a, S. 125)

## 4.2 Lernziele des PSPs

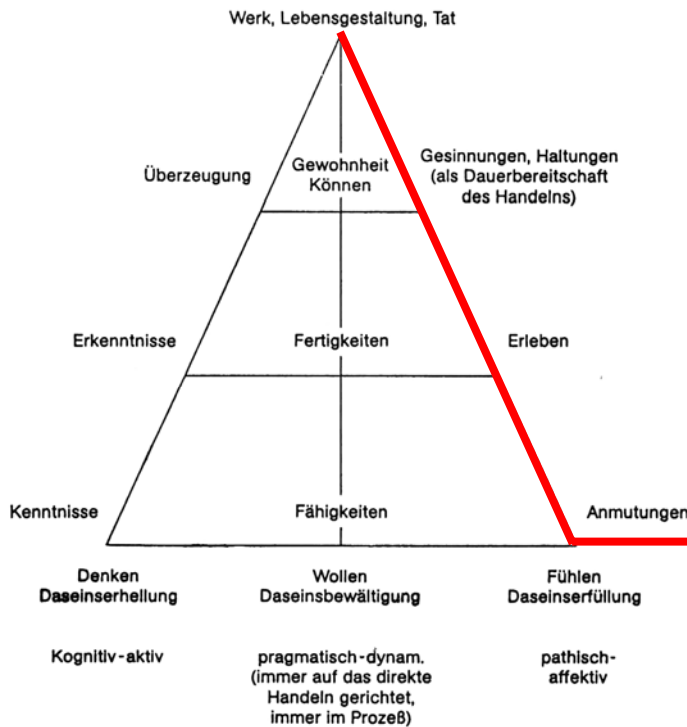


Abbildung 6.6 (aus: Heimann 1976 a, S. 125)

### • Pathisch-affektive Dim.

- Durch die manuelle Erfassung der Daten bekommen die L „Anmutungen“ bzw „Eindrücke“ von ihrer Arbeit
- Alles was sie „kennen lernen“ können sie auch „erfahren“
- Sie bekommen ein „Gefühl“ für:
  - Ihre Arbeitszeit, Unterbrechungen, ...
  - Ihre Defekte und den Beseitigungsaufwand, ...

## 4.2 Lernziele des PSPs

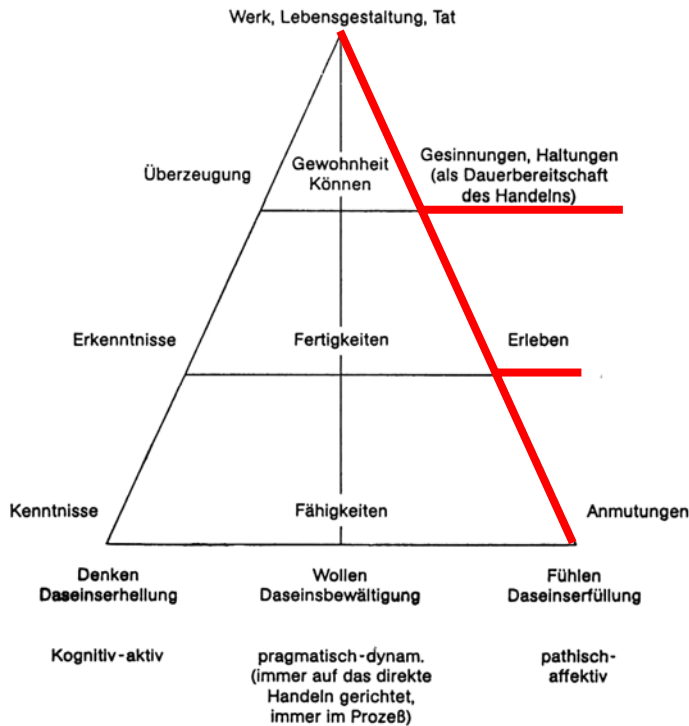


Abbildung 6.6 (aus: Heimann 1976 a, S. 125)

### • Pathisch-affektive Dim.

- Mit dem PSP machen sie „Erfahrungen“ (Erlebnis ist hier unpassend)
- Die Erfahrungen können sich in einer grundsätzlichen „Haltung“ niederschlagen
  - „Ich benutze weiter PSP“
  - „Ohne Zeitmanagement geht es nicht“
  - „Reviews helfen mir sehr“

## 4.3 Lernziele des PSPs - Zusammenfassung

---

- Lernziele der ersten Stufe gehen leicht verloren.
    - Kenntnisse werden vergessen
    - Fähigkeiten entlernt
    - Anmutungen gehen verloren
  - Gelerntes auf höheren Stufen bleibt länger erhalten, prägt sich ein
  - Man möchte als Lehrender immer Lernziele auf höheren Stufen erreichen (zumindest mittelfristig)
-

## 4.3 Lernziele des PSPs - Zusammenfassung

---

- *[Johnson 02]:* PSP lernen ist wie Latein lernen
- Die gewonnenen Erkenntnisse und Einsichten bleiben helfen weiter, selbst wenn man PSP nicht mehr benutzt



## 4.3 Lernen durch Handeln

---

- Wie erreicht man den Erkenntnisgewinn ?
    - Die drei Dimensionen der Lernziele sind eng verknüpft. Sie bedingen sich gegenseitig.
    - *[Heimann 76]:* „Danach ist uns auch der Lernende ein Handelnder! Das wäre die Grundthese.“
    - Lernen ist immer eine Form des Handelns, wer nicht handelt lernt nicht.
-

## 4.3 Lernen durch Handeln

---

- Diese Sichtweise geht zurück auf konstruktivistische Lerntheorie (Piaget, Aebli, ...)
  - Der Lernende lernt aktiv, er baut sich sein Wissen selbst auf
  - Kein anderer kann für ihn lernen! („Nürnberger Trichter“ funktioniert nicht)
  - Die ist Grundlage aller „modernen“ bzw. „reformpädagogischen“ Unterrichtsformen und Methoden (Montessori, Freinet, Offener Unterricht, Projektarbeit, Handlungsorientierter Unterricht, ...)
-

## 4.3 Lernen durch Handeln

---

- Es gilt: Je intensiver die Auseinandersetzung mit dem Stoff, desto leichter wird er gelernt
  - Intensiv heißt:
    - Unterstützung des Selbstlernens
    - Ganzheitlich lernen in allen drei Dimensionen
    - Pestalozzi: „Lernen mit Kopf, Herz und Hand“
-

## 4.4 PSP-Tools im Lernprozess

---

- Auswirkung von PSP-Tools auf die pragm.-dyn. Lernziele:
  - Die Auseinandersetzung mit den PSP-Techniken ist weniger stark
  - Arbeitstechniken, Projektplanung und Qualitätsmanagement gehen nicht „in Fleisch und Blut“ über
  - Sie werden einen z.T. abgenommen
- Auswirkung von Hackystat:
  - Sämtliche Techniken sind automatisiert und werden dem Programmierer abgenommen
  - Es findet hier kein Lernen mehr statt

## 4.4 PSP-Tools im Lernprozess

---

- Auswirkung von PSP-Tools auf die path.-affektiven Lernziele:
  - Die affektive Dimension ist stark eingeschränkt
  - Erfahrungen werden nicht gemacht
  - Eindrücke können nicht entstehen
- Auswirkung von Hackystat:
  - Eindrücke reduzieren sich auf das Betrachten der generierten Ergebnisse aus den Daten

## 4.4 PSP-Tools im Lernprozess

---

- Höherwertige kognitiv-aktive Lernziele sind abhängig von Unterstützung durch die anderen Dimensionen
  - Daher reduziert sich wahrscheinlich auch der Erkenntnisgewinn bei den halb- und voll-automatischen Verfahren
  - Er kann nur noch in der aktiven Auseinandersetzung mit den produzierten Ergebnissen stattfinden
  - Das ist schwieriger!
-

## 5. Fazit

---

- Durch PSP lernen heißt es selber tun.
- Johnsons „context-switch“ fördern den Erkenntnisgewinn
- Der „overhead“, die Auseinandersetzung und der Aufwand, fördern ihn auch.
- Die Latein-These muss erweitert werden:
  - PSP lernen ist wie Latein lernen. Es ist auch genauso anstrengend!
- Vokabeln und Grammatik „pauken“ kann einen keiner abnehmen

## 5. Fazit

---

- Möchte man also mit PSP lernen
  - Strukturiertes und diszipliniertes zu arbeiten
  - Qualitätsbewusster zu entwickeln
  - Seine Schwächen (Defekte, Abschätzungen) zu überwinden
- Dann ist es besser durch die „harte Schule“ zu gehen und PSP von Hand zu benutzen



## 5. Fazit

---

- Aber: PSP kann mehr als den Programmierer lehren!
- PSP wirkt sich positiv auf Produkte und Projekte aus
- Tools und Hackystat unterstützen den erfahrenen Programmierer bei der Benutzung von PSP (ähnlichen) Verfahren
- Wer bereits diszipliniert, qualitativ hochwertig arbeitet kann Hackystat benutzen, um sich interessant Metriken über seine Produkte und Projekte erzeugen zu lassen

## 5.1 MOTD

---

- *"Erkläre es mir, und ich werde es vergessen. Zeige es mir, und ich werde mich erinnern. Lass es mich selber tun, und ich werde es verstehen." Konfuzius*

## 6. Referenzen

---

- [Humphrey 95] W. S. Humphrey. A Discipline for Software Engineering. Reading, MA. Addison-Wesley 1995
- [Humphrey 1996] W. S. Humphrey. Using a defined and measured personal software. IEEE Software May 1996. S. 77f.
- [Ferguson 1997] P. Ferguson, W. S. Humphrey, S. Khajenoori, et al. Results of applying the personal software process. IEEE Computer 30(5) 1996. S 24f.
- [Hayes 1997] W. Hayes, J. W. Over. The Personal Software Process (PSP): An Empirical Study of the Impact of the PSP on Individual Engineers. Technical Report CMU/SEI-97-TR-001, ESC-TR-97-001. Software Engineering Institute. December 1997.
- [Weslén 2000] A. Weslén. A Replicated Empirical Study of the Impact of the Methods in the PSP on Individual Engineers. Empirical Software Engineering May 2000. S. 93f.
- [Johnson 2002] Johnson, Kou, Agustin et al. Beyond the PSP: Metrics collection and analysis for the differently disciplined. Proceedings of the 2003 International Conference on Software Engineering. Portland, Oregon. May 2003
- [Borstler 2002] J. Borstler, D. Carrington, G. Hislop, et al. Teaching PSP: Challenges and lessons learned. IEEE Software, 19(5), September 2002.
- [Prechelt 1997] Experience Report: Teaching and Using the Personal Software Process (PSP). Submission to ESEC 1997. Online unter URL: [http://page.inf.fu-berlin.de/~prechelt/Biblio/psp\\_experience.pdf](http://page.inf.fu-berlin.de/~prechelt/Biblio/psp_experience.pdf)
- [Prenzel 93] M. Prenzel. Autonomie und Motivation im Lernen Erwachsener. In: Zeitschrift für Pädagogik 1993, 39. Jg., H. 2, S. 239f.
- [Heimann 76] P. Heimann. Didaktische Grundbegriffe. In: Reich/Thomas (Hrsg.) 1976. S. 103f.
- [Meyer 94] W. Jank, H. Meyer. Didaktische Modelle. Berlin. Cornelsen Scriptor. 3. Aufl. 1994