大学におけるプロセス改善教育のあり方について
— PSP法実践の経験をもとに —

2001年6月14日 高知会館

信州大学 工学部 情報工学科

海谷 治彦

本発表の主張

- 大学ではPSPを通じて測定・開発技法の体験ができる程度でよい(というか精一杯).
- (難業である)PSPを続ける動機付けが必要.
  - 授業期間内.
  - 授業が終わってからの？
- プロセスが改善することを動機付けとして使うのは困難 – 実際，授業期間では改善しない.

発表の概要

- PSPの概要説明
- 我々の大学院でのPSP訓練結果の紹介
  - 授業の概要とアンケート結果
  - 工数，規模の見積もりについて
  - スケジュール見積もりについて
  - 欠陥除去について
- 疑問と回答案

PSPとは？

- Personal Software Process ^{SM}
- 個人用の自己改善プロセス(のトレーニングコース)
- 目的: よりよい技術者になる助けをすること (教科書†より).
- CMMの中での個人に関わる部分に注目
  - プロセス定義，記録，計画，評価(品質管理)

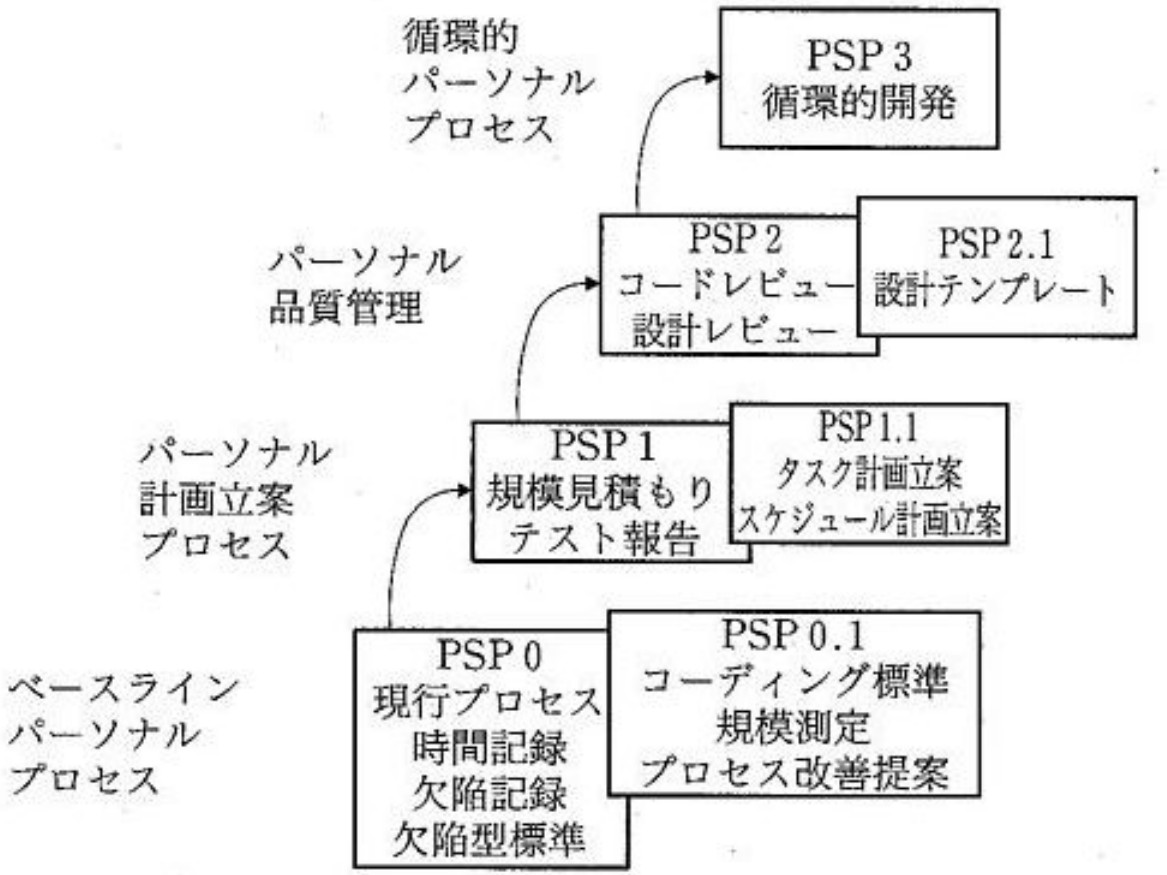
† W. S. Humphrey. A Discipline for Software Engineering(DSE). Addison-Wesley, 1995.

PSPをどう実践するか？

- 訓練として: 教科書†を使い10個程度の演習問題を遂行.
  - 15週程度のコース，10個の開発，5個の報告.
  - かなりの数の帳票を作成しなければならない.
- 実務として: 業務内で個々人が自己プロセスの定義，測定，(分析,) 改善を繰り返す.
  - CMM lev.3以上程度の組織でないと難しい？

† W. S. Humphrey. A Discipline for Software Engineering(DSE). Addison-Wesley, 1995.

PSP説明のよくある図



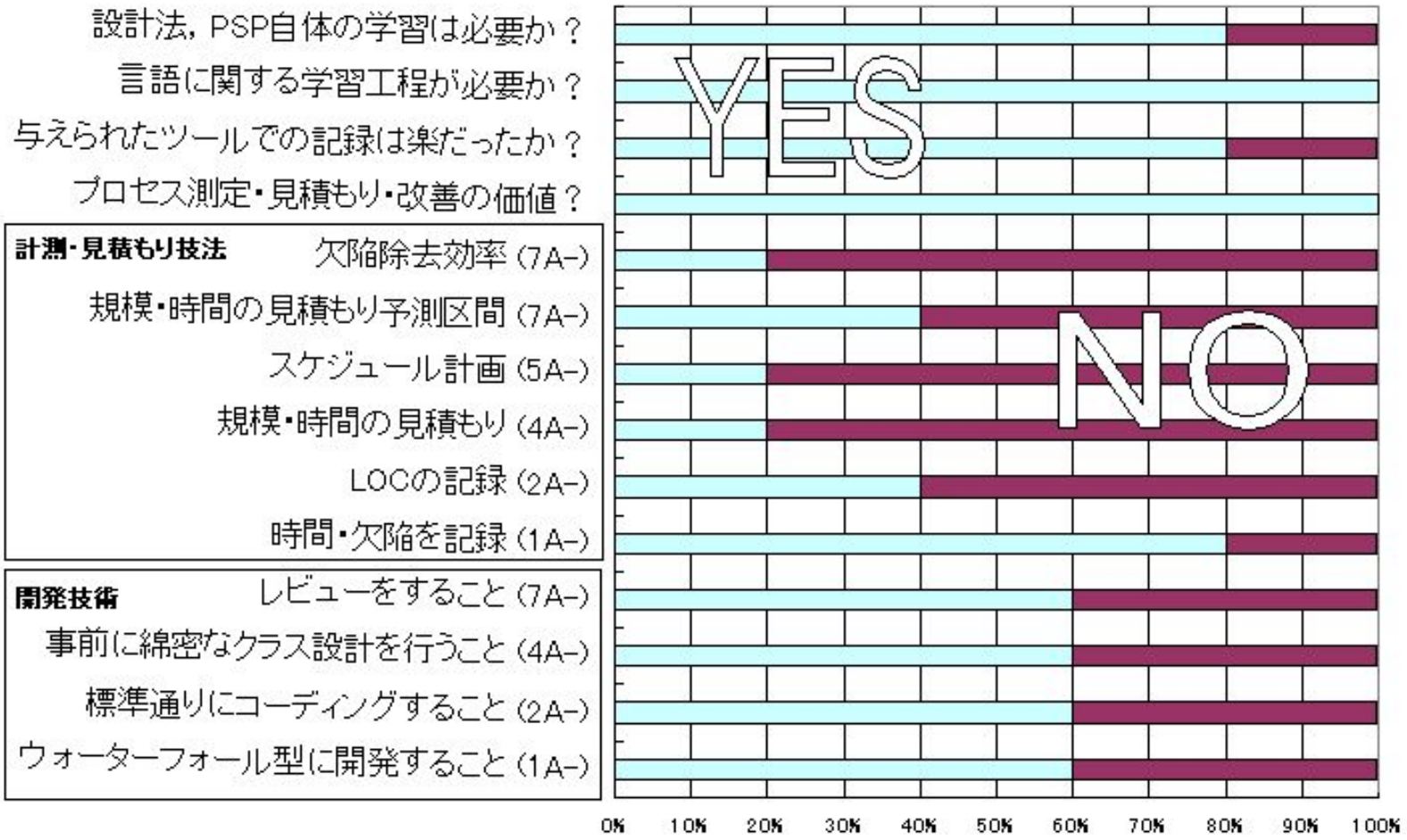
PSPの具体内容

PSP	測定技法	開発技法	問題	
0	時間・欠陥記録	ウォーターフォール開発	1A	期待値と標準偏差
0.1	LOC記録	コード標準	2A	LOCカウンタ
			3A	関数毎LOCカウンタ
1	規模・時間見積り(Probe)	**事前設計**	4A	線形回帰パラメータ
1.1	スケジュール見積もり 欠陥見積もり，見積もり予測範囲		5A	シンプソン積分・正規分布
			6A	予測区間・t分布積分
2	品質測定	**レビュー**	7A	相関とその有意性
2.1		形式手法	8A	ソート
			9A	正規分布の検定
3		スパイラルな開発	10A	重回帰分析

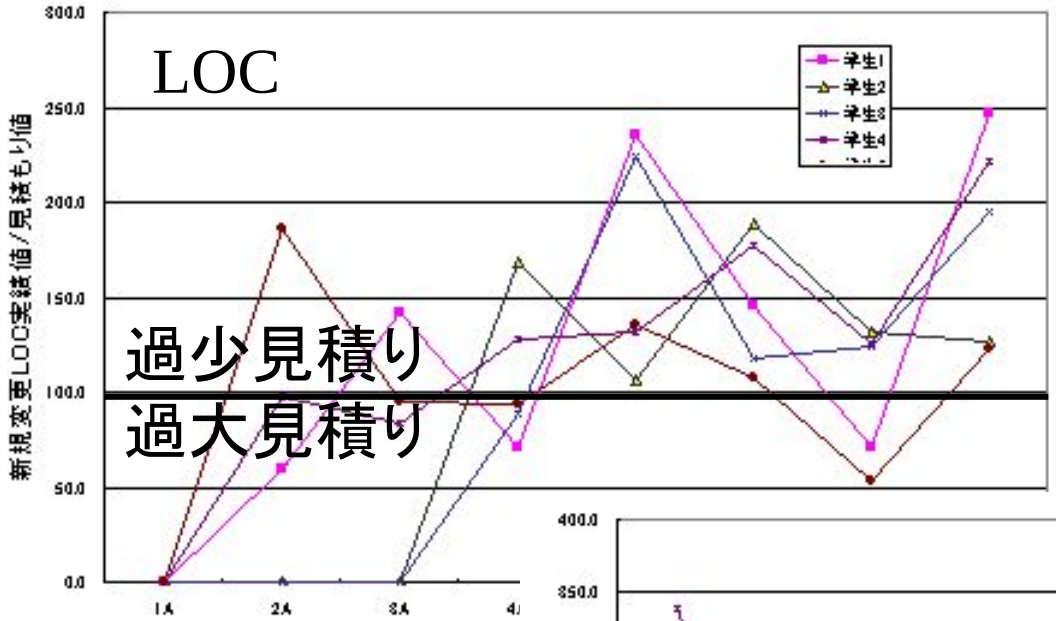
信州大 情工 修士授業の概要

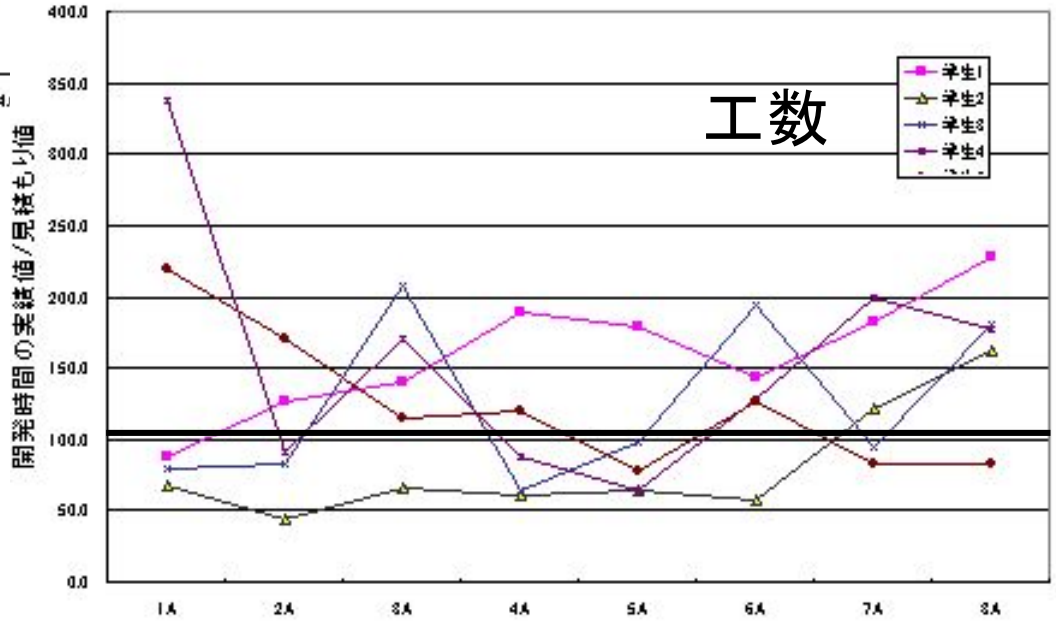
- 講師は私(海谷, PSPを知ったのは1996年)
- 2回(年)目の講義
- 2000年度受講生: M1が5名(最初は8名)
  - 本学科・修士課程は1学年30人程度
- 8A(PSP2)までしかできなかった
- 帳票用 MSエクセルのシートを提供
- 言語はCに固定，コード標準を付与
- 学習項目決定，学習工程を追加

アンケート結果


設計法，PSP自体の学習は必要か？
言語に関する学習工程が必要か？
与えられたツールでの記録は楽だったか？
プロセス測定・見積もり・改善の価値？
計測・見積もり技法
欠陥除去効率 (7A-)
規模・時間の見積もり予測区間 (7A-)
スケジュール計画 (5A-)
規模・時間の見積もり (4A-)
LOCの記録 (2A-)
時間・欠陥を記録 (1A-)
開発技術
レビューをすること (7A-)
事前に綿密なクラス設計を行うこと (4A-)
標準通りにコーディングすること (2A-)
ウォーターフォール型に開発すること (1A-)
YES
NO
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%

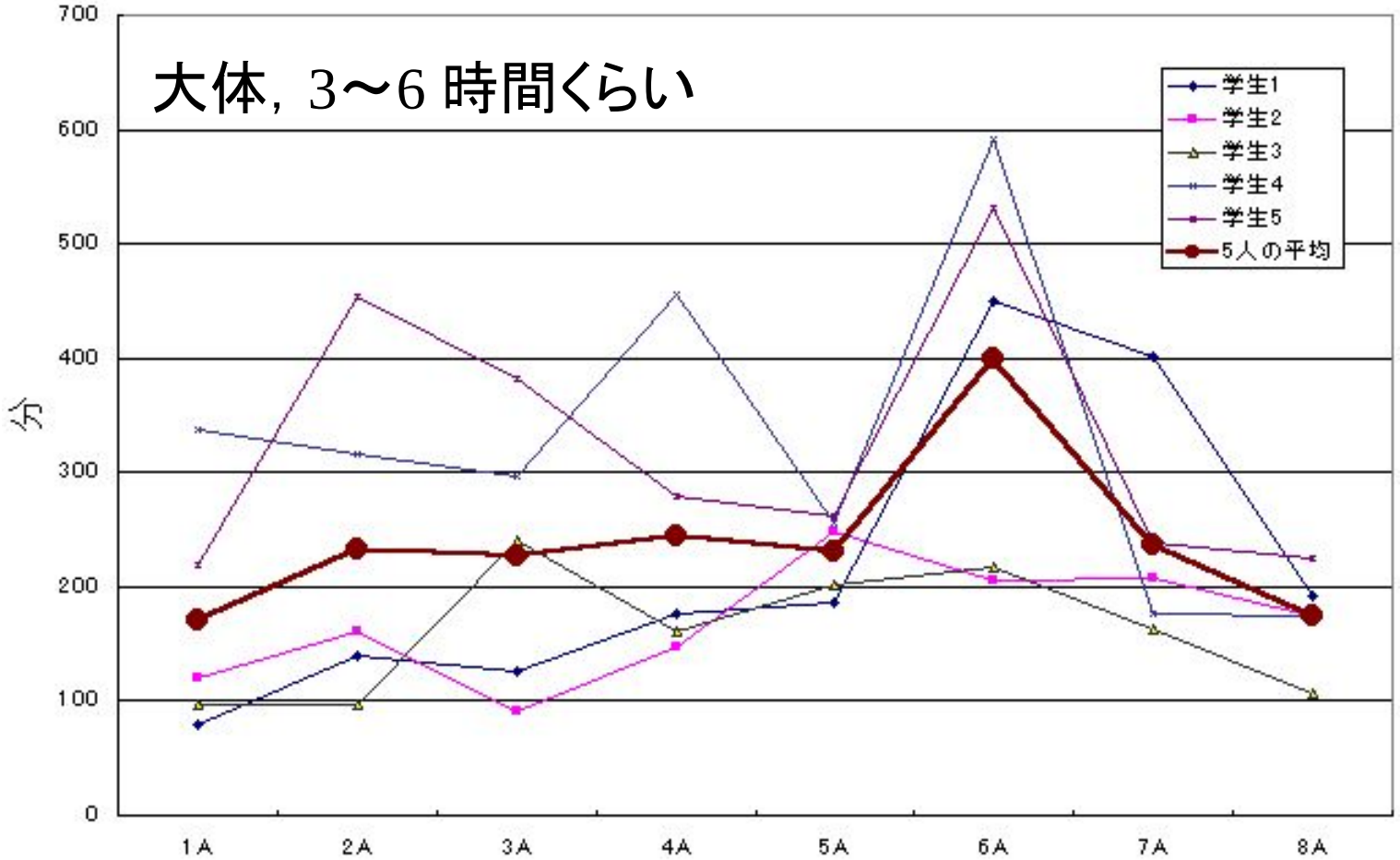
規模と工数の見積もり





- 当たらない
- 収束(改善)しない

開発時間について


大体，3〜6 時間くらい
700
600
500
400
300
200
100
0
分
1A
2A
3A
4A
5A
6A
7A
8A
学生1
学生2
学生3
学生4
学生5
5人の平均

スケジュールを計画をする

		計画			実績		
週番号	週頭日付(月	直接時間	累積時間	累積計画価	直接時間	累積時間	累積獲得価値
1	日曜日(朝)	0	0	0	0	0	0.0
2	日曜日(昼)	0	0	0	0	0	0.0
3	日曜日(夜)	221	221	100	307	307	73.0
4	月曜日(朝)	0	221	100	0	307	73.0
5	月曜日(昼)	0	221	100	94	401	100.0
6			221			401	
7			221			401	

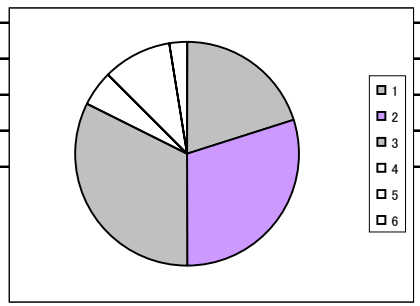
.... んですが

開発時間帯の分析

- ほとんどの開発が3ピリオドで終わっている.
- 計画する意味があるのか？

ピリオド
=1/4日
=午前 |
午後 |
夜 |
深夜

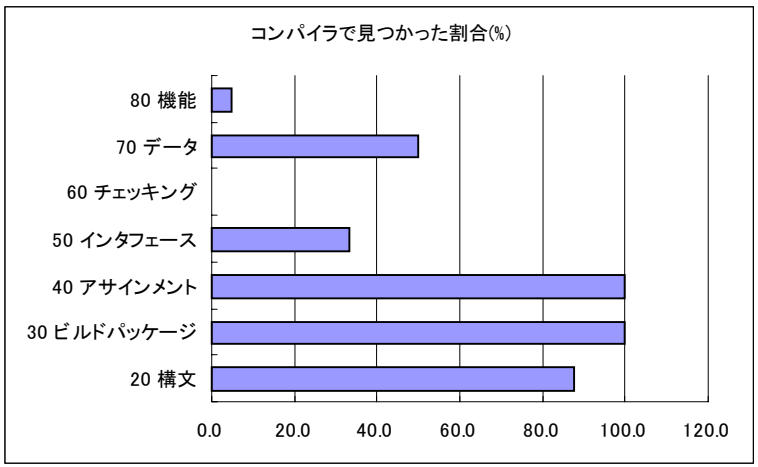
	Assigned	Deadline	# of day	# of Working Periods (= day/4) 1	2	3	4	5	6
1A	1-Oct	16-Oct	15	4	1				
2A	18-Oct	23-Oct	5	1	2	2			
3A	25-Oct	6-Nov	12	1	2			1	1
4A	8-Nov	21-Nov	13		2	1		2	
5A	22-Nov	28-Nov	6		1	3	1		
6A	29-Nov	5-Dec	6			3	1	1	
7A	13-Dec	31-Dec	18	1	2	2			
8A	10-Jan	19-Jan	9	1	2	2			
toltal (40 data)				8	12	13	2	4	1
student1				2	3	3			
student2				2	2	3	1		
student3				1	4	2	1		
student4				1	1	3		3	
student5				2	2	2		1	1



欠陥除去について

データは
7Aまでのもの

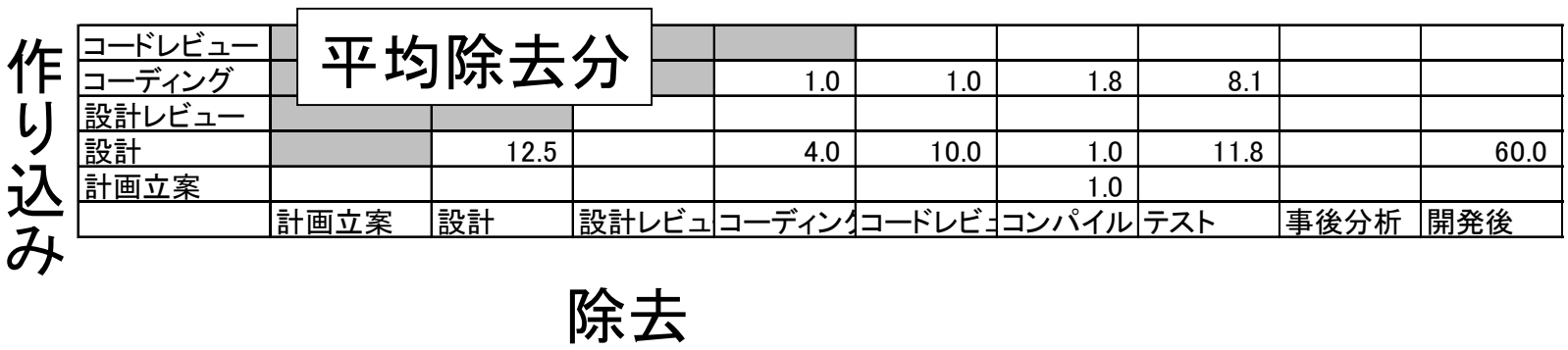
欠陥型	コンパイル開始時の欠陥数	コンパイルで発見した欠陥数	コンパイラで見つかった割合(%)
10 文書	0	0	
20 構文	24	21	87.5
30 ビルドパッケージ	5	5	100.0
40 アサインメント	3	3	100.0
50 インタフェース	9	3	33.3
60 チェッキング	1	0	0.0
70 データ	2	1	50.0
80 機能	20	1	5.0
90 システム	0	0	
100 環境	0	0	
合計	64	34	53.1



教科書通り,
困難な型はコンパイラではとれてない.

作り込み/除去の工程差

- 工程差が激しいからといって，一般に平均除去時間が長いわけではない．
- コーディングに関しては，その傾向がある．



PSPレベルアップと除去能力

PSP0

コーディング		1		129	234		
		1		64	30		
設計レビュー							
設計	20	4		2	66		60
	1	1		2	6		1
計画立案							
時間/個数	設計	コーディン	コードレ	コンパ	テスト	事後分	開発後

3.5

少しは早期発見の傾向にあるが，ほとんど誤差の範囲

PSP1

コーディング				60	150		
				38	16		
設計レビュー							
設計	5				12		
	1				2		
計画立案				1			
				1			
時間/個数	設計	コーディン	コードレ	コンパ	テスト	事後分	開発後

3.4

早期発見=
作りこみ-除去工程が近接している

PSP2

コーディング		2	3	29	54		
		2	3	16	8		
設計レビュー							
設計			10		40		
			1		2		
計画立案				1			
				1			
時間/個数	設計	コーディン	コードレ	コンパ	テスト	事後分	開発後

3.3

議論…. というか疑問

- 大学でプロセス改善教育(体験)が必要か？
  - ？？
- PSPはそれに適当か？
  - 実開発チームを作りにくい以上，悪くない解だと思う.
- 実際改善しない現状を鑑みて，どのような動機付けが必要？
  - 技法の遂行と，その効果をある程度，受講者本人が確認できる.
    - 教科書的な事実の確認が動機付けになるかも.
  - 無論単位というのも大きい，大学ですから.
- どうしたら改善活動を授業後も継続できるか？
  - ？？

PSP自体の限界

- 個人の開発履歴(経験？)に依存する.
  - 開発言語が変更になると振り出し？
  - 開発分野(ドメイン)が変更になると振り出し?
- 定義，記録，改善というプロセスを回すためのスキルは個人に残る？
- そもそも実開発に適用できるのか？
  - ?
- 投入する技法のベンチマーク法に使える？

参考場所

- SEIの本家 (TSPと)PSPのページ

  http://www.sei.cmu.edu/tsp/
- PSPネットワークのページ: PSPの情報交換をするグループです.

  http://www.azuma.mgmt.waseda.ac.jp/psp/

  海谷も参加してます.
- 私の授業のページ: 一部非公開ですが...

  http://dotcom.cs.shinshu-u.ac.jp/psp/

以上