Smalltalk Conversion Economics

Instantiations

instantiations
Build Quality Software

1

Introduction

- Purpose: To Identify Estimating Models for Smalltalk Conversions
- Topics of Discussion
  - Function Points
  - Productivity Comparison
  - Quality Comparison
  - Cost Comparison
  - Defect Injection

instantiations
Build Quality Software

2

Sizing Source Code Volumes

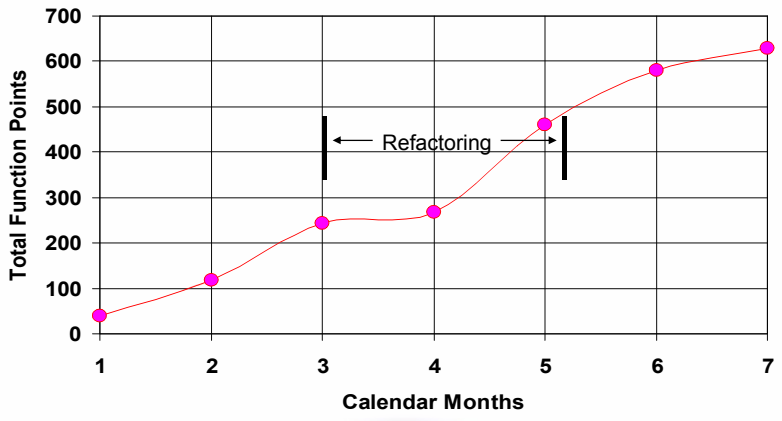
Language	Source lines of code per function point
Smalltalk	21
C++/Java	53
ADA83	71
PL/1	80
FORTRAN/COBOL	107
C	128
Macro Assembler	213
Basic Assembly Language	320

Based on SPR studies

instantiations
Build Quality Software

3

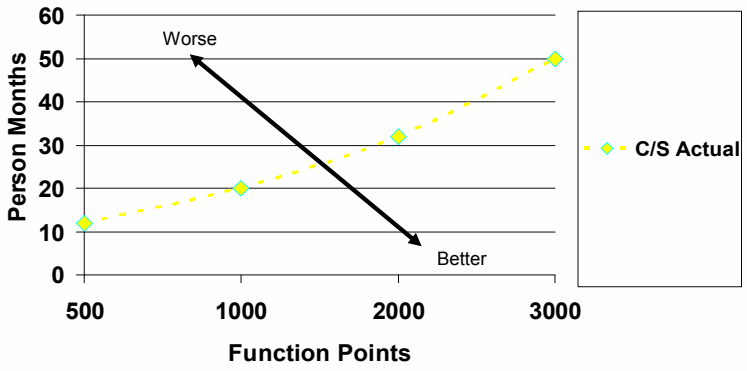
e-Web Development Ebb and Flow



instantiations
Build Quality Software

4

Actual Smalltalk Productivity Sample



Based on McConnell, "Rapid Application Development", SPR studies and actual measured results

instantiations
Build Quality Software

5

Efficient Schedules

Sample Project Effort Estimation Table

System Size (Lines of Code)	Schedule Months	Business Products Effort Man-Months
10,000	4.9	5
15,000	5.8	8
20,000	7	11
25,000	7	14
30,000	8	20
35,000	9	24
40,000	9	30
45,000	9	34
50,000	10	40
60,000	10	49
70,000	11	61
80,000	12	71
90,000	12	82
100,000	13	93

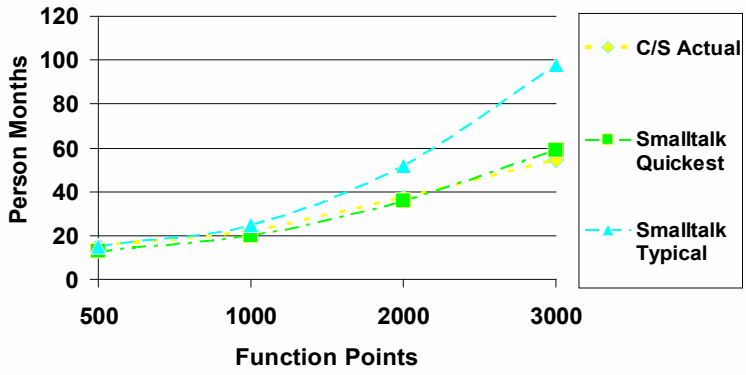
Actually 13K LOC in 7 Man-Months

Table 8-9 Efficient Schedules from S. McConnell's *Rapid Development* derived from data in *Software Engineering Economics* (Boehm 1981), "*An Empirical Validation of Software Cost Estimation Models*" (Kemerer 1987), *Applied Software Measurement* (Jones 1991), *Measures for Excellence* (Putnam & Meyers 1992), and *Assessment and Control of Software Risks* (Jones 1994)

instantiations
Build Quality Software

6

Comparison of Actual & Estimated Smalltalk Productivity

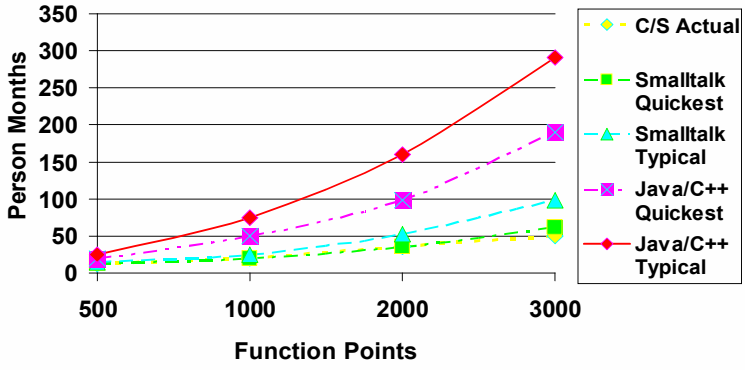


Based on McConnell, "Rapid Application Development", SPR studies and actual Linea Performance

instantiations
Build Quality Software

7

Relative Java & Smalltalk Productivity

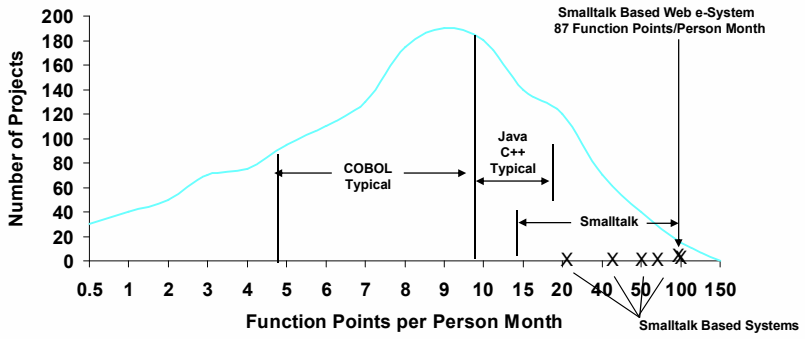


Based on McConnell, "Rapid Application Development", SPR studies and actual Linea Performance

instantiations
Build Quality Software

8

Distribution of Productivity Rates



Distribution of productivity rates of 1500 software projects Figure 16.1
from Capers Jones' "Estimating Software Costs" modified with SPR productivity data

instantiations
Build Quality Software

9

Productivity and Cost Comparisons

Smalltalk based

	e-Web	*C/S*	*Prudential*	*L*	*B*	*SPR Java C++*
Function Points/ Person Month	87	57	45	55	24	20-10
Cost/ Deployed Function Point	$185	$410	$536	?	?	$1000

instantiations
Build Quality Software

10

Cost and Schedule

	VisualAge Smalltalk		C++, Java	
Source Lines of Code Per Function Point[1]	21		53	
5000 FP Equivalent Source Lines of Code	105K		265K	
	Months	Person Months	Months	Person Months
Shortest Possible Schedule[2]	9	110	13	330
Efficient Schedule[3]	13	93	19	280
Typical Nominal Schedule[4]	15	165	22	480
Typical Number of Software Developers[5]	11		22	
Hourly Software Developer Cost	$40 to $100		$40 to $120	
Typical Project Estimated Total Cost[6]	$1M to $2.5M		$3M to $9M	

[1] Capers Jones, op. cit.,
[2] McConnell, op. cit., Business Products, Table 8-8.
[3] Ibid., Table 8-9
[4] Ibid., Table 8-10
[5] For a typical nominal schedule, Person Months/Months
[6] (Hourly Software Developer Cost) x (160 hours per person month) x (Typical Nominal Schedule Person Months)

Impact of Software Quality

- Testing now consumes half of Development Schedule
- Many organizations developing software with defect levels that give longer schedules than necessary
- Based on 4000 projects surveyed:
  - Poor quality is one of most common reasons for schedule overruns
  - Poor quality implicated in half of all canceled projects
- Up to four times the normal number of defects reported for products developed under excessive schedule pressure

Defects vs. Development Time

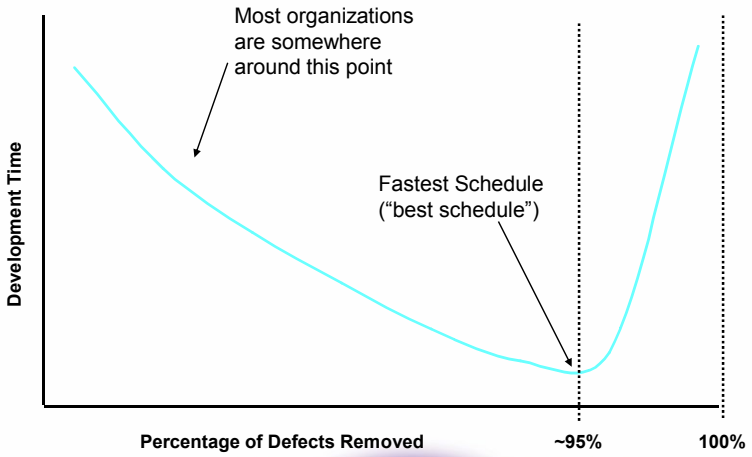


Figure 4.3 from S. McConnell's Rapid Development adapted from T. Capers Jones' "Applied Software Measurement"

instantiations
Build Quality Software

Programming Errors per FP

Smalltalk	***0.14***
SQL	0.18
Objective C	0.19
Eiffel	0.21
Visual Basic	0.21
ADA 95	0.50
Java	***0.50***
RPG	0.54
C++	0.82
Modula II	0.84
Quick Basic	1.11
Pascal	1.18
PL/1	1.30
COBOL	1.50
FORTRAN	1.68
Algol	1.71
C	2.50
Macro Assembler	5.01
Basic Assembly	9.28

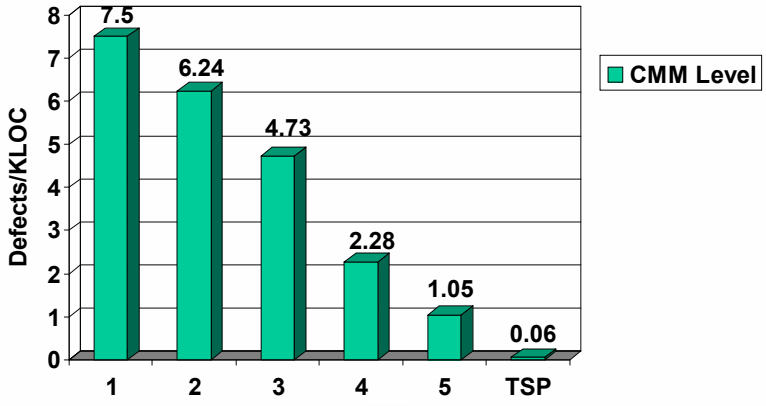
← Smalltalk can provide the basis for extremely high quality systems!

← **Industry Average 1.40**

Coding and syntax bugs only, per function point, Table 21.8 from T. Capers Jones' "Estimating Software Costs"

instantiations
Build Quality Software

Defect Density of Delivered Software



Ref SEI Technical Report 2003-014

instantiations
Build Quality Software

15

Impact of Software Quality

- Experienced developers inject 100+ defects/KLOC
- Only half of defects found by compiler
- Reminder must be detected by inspection & testing
- 50KLOC (2400FP) ST software conversion would
  - start with 5K defects
  - Enter testing with 50+ defects/KLOC (2500 defects)
  - Have half of defects found in unit test at a rate of 2-3 hour
  - Have remainder found through system testing at 10-20 hours/defect
  - Have total testing 13K-25K hours (i.e., 5 developers 12-18 calendar months, $500K - $2.5M)

Watts Humprey PSP pg 142

instantiations
Build Quality Software

16

Conclusions

When significant new functionality must be developed

- Smalltalk for 4-9x more productive than Java.
- Smalltalk software development quality is unparalleled and is 3x better than Java, on a per function point basis.
- Smalltalk to Java conversion costs can match original ST development costs
- Economics makes most ST to Java conversion VERY unrealistic

instantiations
Build Quality Software

17