Measurement for improving accuracy of estimates: the case study of a small software organisation

Sylvie Trudel
SMEF 2007 – Rome, Italy
May 9th, 2007

CRIM

Your technology accelerator

Content

CRIM

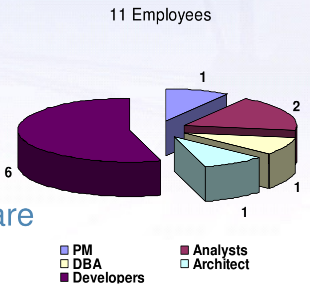
- Introduction
- Company
- Process
- Product
- Project estimation
- Improving estimation models
- Preliminary results of the “weighted size” approach
- Conclusion and future work

Introduction

- Accurate software project estimates:
  - Art?
  - Utopia?
  - No, measurement based methodologies!
- Effort and size known to be highly correlated, but…
  - These 2 measures do not guarantee estimation success
  - The team must understand other influencing factors
  - Adding factors to an estimation model may make it less accurate
- Here is the case study of a small Canadian software development company…

Company overview

- 22 years of existence
- 11 employees
  - All development team members
  - Accounting and house keeping are subcontracted
- 6 active customers
  - 1 large financial organisation ≤ 80% gross revenues
  - 10 years of development of an ERP called “SUM”
- Backlog of projects = 6 to 8 months



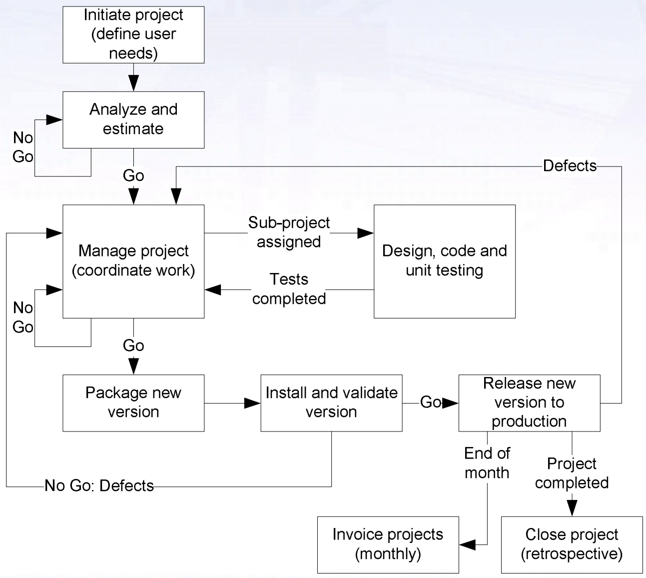
Business model

- « Not to exceed » project estimates
  - If actual cost ≥ estimate → invoice=estimate
  - If actual cost < estimate → invoice=actual cost
- When estimate considered too high by the customer → Project off-shoring to India!
  - Strong motivation for accurate estimates!
- Any defect found by the customer is to be fixed at the company's expense
  - Strong commitment to quality!
- Effort to initiate project, analyse requirement, and estimate project is billed to the customer
  - Final estimation includes 6 activities:
    - project management,
    - software development
    - testing, documentation
    - packaging
    - validation

Process improvement initiative

- Motivation
  - Missed deadlines on short bi-weekly release cycles
  - Estimates exceeded in 50% of projects
  - Loss potential projects to outsourcing organisations in 2001-2002
- Started PI in 2004, guided by the CMMI
  - Project-oriented
  - 1 project = set of related features
  - 50 hrs < project size < 1300 hrs, average=150 hrs

Process overview



Measurement program

- No measurement plan at first, but…
  - They were measuring effort and schedule
    - To invoice customer every month
- Fall 2006: start measurement plan
  - Exercise to understand information needs
    - Manager
    - Team members
- Classic “Goal-Question/Indicators-Measure” approach

Measurement plan

- Allows the manager and team members to think about their information needs and the quality of measurement
- Simply documented in Excel (only 3 worksheets)

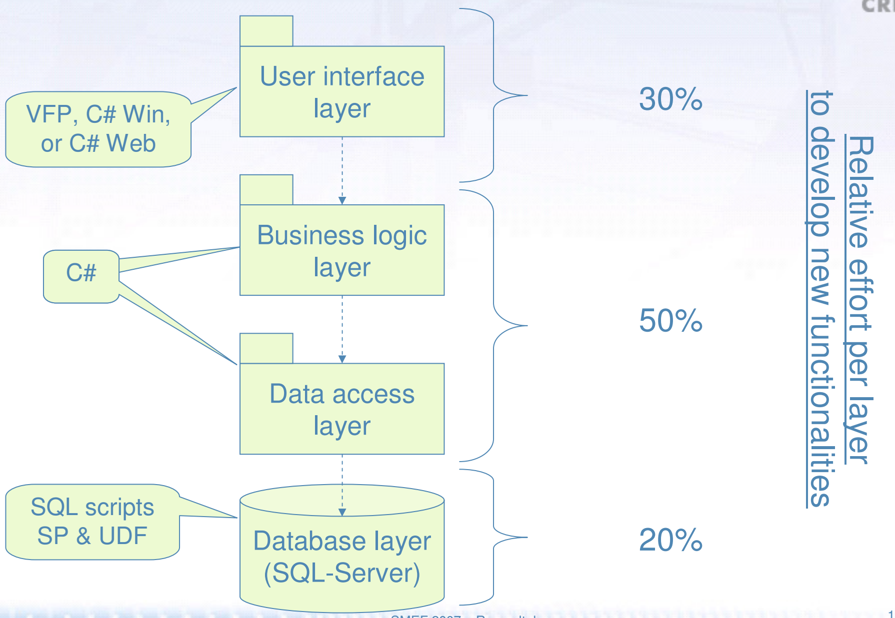
- Goals
  - Goal description
  - Reason

Example of measurement plan

- Indicators
  - Indicators (questions)
  - Formulas
  - Goal it relates to
  - Unit of measure
  - Source of data
  - Responsible
  - Where stored
  - When measured
  - Consumer
  - Analysis procedure
  - Possible actions

- Measures
  - Measures
  - Scope
  - Unit of measure
  - Precision
  - Who measure?
  - Data store
  - Data collection procedure
  - Quality assurance

Product overview



Product release cycle

- 1 product release every 2 weeks
  - 1 release = 1..N features from 1..N projects
  - 1 project = 1..N releases
  - Supplemental releases:
    - Only for urgent feature or bug fixing

	Mon	Tue-Wed	Thu	Fri	Week-end
AM	If any, fix defects 1hr Package 10m Deploy 30m		Package release 10m Feature testing 1/2d Test readiness 5m	If any, fix defects 1 hr	Deployment 30m
PM			Supervised validation testing 1..4 hrs	Re-testing	

Product quality

- In 2006, 35 product releases
  - 17 releases with ZERO defects
  - 18 releases with a total of 28 defects
    → 1.55 defect / release, all fixed within ½ day
- No bug tracking tool
  - Defects are not “managed”, they are “fixed”

Project estimation

- Before 2005:
  - Task-effort estimation only
- From 2005:
  - 2^{nd} method added based on FSM with COSMIC-FFP, and actual effort, to validate 1^{st} estimate
- Productivity ranges 1.5 to 6 hours/cfsu. Why?
  - CR not systematically measured nor estimated
  - Once performed and isolated, performance variation ranges -6% to +27%

Improving estimation models: a six-steps approach

Step 1: assess reasons for inaccuracy from product and process

- Ratios initially used to adjust estimation model
  - Add new data movement = 100% of effort
  - Delete data movement = 10% of effort
  - Modify existing data movement = 50% of effort
- Problems
  1. Seemed appropriate only if SW in a single layer
  2. With multi-layers architecture, developing new data groups requires more effort to create when developing the first functional process
  3. When modifying existing data groups and data movements, there is a significant difference of effort due to the number of attributes affected, and thus the 50% ratio for maintenance needed to be redefined
- Considering the developer's viewpoint was abandoned
  - Risk of increasing measurement effort

Step 2: evaluate impact of reuse from software architecture layers

- Developing the 2nd functional process
  - All required database components and many business logic components already exist

	Effort ratio			
Software layer	**New**	**Reuse**	**Minor change**	**Major change**
User interface	30%	15%	10%	30%
Business logic and data (C#)	50%	10%	10%	30%
Database layer (SQL)	20%	0%	10%	10%
Total:	**100%**	**25%**	**30%**	**70%**

Step 3: apply reusability factors to data movements

Module	Funct. Process	Data Group	Movement types	Reuse type	R	X	E	W	FFP total	Reuse impact	Weighted size
					13	14	13	12	52	-9,75	42,25
Create email/ fax	Display main window		Déclencheur de l'action	New	0	0	1	0	1	0	1
Create email/ fax	Display main window	DocumentHeader	Table dynamique : Read & Exit	Reuse	1	1	0	0	2	-1,5	0,5
Create email/ fax	Display main window	curDocHeader	Table dynamique : Input & Write	New	0	0	1	1	2	0	2
Create email/ fax	Display main window		Message(s) simple(s)	New	0	1	0	0	1	0	1

- It takes 1 to 2 seconds to identify movement types and reuse impact per data group per functional process
- 1.5 hour to measure an average project

Step 4: establish estimation models per technology

Technology	VFP	C# for Windows	C# Web
Estimation model (hours/WSU)	3.22	3.86	5.15

Initial estimation models based on weighted size units (WSU) per technology

Then, 3 C# projects and 2 VFP projects were measured…

Step 5: adjust effort estimation with risk factors

- 3 risk factors influencing productivity on certain projects:
  - technology: known or unknown
  - complexity: low, medium, high
  - number of other stakeholders involved: none, third party, one or many vendors
- Risk contingency = % total effort
- No risk perceived in majority of projects
  - So as in the sample of projects

Step 6: validate effort estimation with actual data

#	Techno-logy	FFP	WSU	Original est. (hours)	Actual effort (hours)	Over-run %	Hr/FFP	Hr/WSU
1	C# Win	218	159.0	598	567.4	-5%	2.6	3.6
2	C# Win	74	53.3	131	109.7	-16%	1.5	2.1
3	C# Win	124	89.5	223	236.9	6%	1.9	2.6
						Average for C# Win:	**2.0**	**2.8**
						Variance for C# Win:	**0.3**	**0.6**
4	VFP	47	42.0	102	78.7	-23%	1.7	1.9
5	VFP	66	55.5	155	138.3	-11%	2.1	2.5
						Average for VFP:	**1.9**	**2.2**
						Variance for VFP:	**0.1**	**0.2**

Preliminary results of the "weighted size" approach

CRIM

- Insufficient number of data points, but…
  - Average productivity for C# Windows projects went from 4.5 to 2.0 hrs/size unit
    - C# learning curve was not over
    - "Net negative producing programmer" dismissed
    - Software process is applied consistently
- Productivity difference of C# Win and VFP decreased significantly
  - New business opportunities?
- Perceived tendency to overestimate
  - Desired to a certain extent, due to business model

Conclusion and future work

CRIM

- Inaccurate estimates vs actual effort
  - Often results of lack of discipline to formalize CRs
- Encouraging variance < 16% on C# Win projects
- Continuously monitor actual performance data → readjust estimation models on a periodic basis, but…
  - If precision of "weighted size" < precision of COSMIC size → use COSMIC size
- Experiments on other formulas for weighted size are underway