Overcoming the Challenges of Estimating in New Development Environments

by

Douglas T. Putnam
Quantitative Software Management, Inc.
2000 Corporate Ridge, Suite 900
McLean, VA 22101
703 790-0055
Doug_Putnam@QSM.com
www.qsm.com
What Motivates Organizations to Get Serious About Software Estimation?

- Of the organizations that make a serious commitment to achieve a significant level of estimation competence
  - 60% have just had a major disaster and decided they never want to live through another
  - 30% are forced into it by their customer
  - 10% find it’s a natural step in their process improvement initiatives

Pain is the Most Significant Motivator!
Major Sources of Pain

- A failure to communicate and establish realistic expectations *(mandated schedules that are impossible)*

- A failure to renegotiate when changes take place *(your accommodating - willing to accept requirements changes)*

- Ineffective mid-course corrective action *(loading people on to accelerate the schedule - Brooks Law effect)*
What is estimation?

- It is a technical activity used to support a business objective
What Business Objectives Does Estimation Support?

- Win New Business (major contracts & multi-year outsourcing deals)
- Schedule negotiations with customer
- Functionality negotiations with customer
- Warranty negotiations with customer
- Cost negotiations with customer
- Return on investment analysis
- Risk Mitigation
Skill Sets for Success

• To be a successful software estimator
  – Good understanding of software project behavior
  – Good appreciation of company business objectives
  – Good people skills
  – Good problem solving skills
  – Good communication skills
  – Good conflict resolution & mediation skills
  – Good sales skills
  – Good data analysis skills
A Closed Loop Process is Essential

Step 1
Historical Data

Empirical Evidence & Professional Judgment

Foundation of the Commitment

Step 2
Estimation

Minimum Schedule Alternatives Baseline Plan

Step 3
Project Control

Actuals compared to plan tactical forecast to complete Manage Commitment

Viable Strategy Commitment

SEI Core Metrics

Schedule

Effort

Size

Defects
A Macro Software Project Model

- Many popular software estimation techniques model the way people solve design intensive problems
  - Software production equation
  - Resource allocation equation (time based distributions - staffing, defects, product construction)

\[
\text{Functionality} \quad = \quad \text{Effort}^x \times \text{Time}^y
\]

Efficiency
Anatomy of Single “Release”

Key Design - Construction Events

Geometry of the resource loading

Well defined
Start

Staffing Profile

Well Defined
End

Life Cycle
3 = DR
4 = FDrop
5 = FREZ
6 = Beta
7 = Del
8 = GA

© Quantitative Software Management, Inc. All rights reserved. PSM Conference 2000 #9
“Multiple Releases” the way most systems evolve today

Single Product Release Plans

Monthly Project Gantt Chart

GRC Release 4.0
GRC Release 4.2
GRC Release 5.0
GRC Release 4.3
GRC Release 7.0
Collecting Project Data

Fear of Measurement

We Don’t Have Any Data!

Facts are Friendly
Project Artifacts

- **Staffing Plans** (schedule and effort)
- **Trouble Reports** (defects discovery rates)
- **Configuration Management Reports** (size artifacts & mapping relationships)
- **Requirements Management Reports** (size artifacts)
- **Time Report Reporting Systems** (effort)
- **Schedule reports** (time line, milestones & phases)

- A subset of these are available on most projects (they aren’t 4 decimal places accurate but they are good enough to get you started now)

- **Sketch a staffing plan during a 30 minute interview with the project manager**
Sample Project Artifacts
COTS Human Resource Application

What happened here?
Making Sense out of Groups of Data
Graphing the Data - Project Positioning

I&T Duration (Months) vs # of Requirements

- **Major Releases**
  - R_1.0
  - R_2.0
  - R_2.1
  - R_2.2

- **Bug Fix - Minor Enhancements**
  - R_1.1

**Management Metric**

- All Completed Projects
- Avg. Line Style
- 1 Sigma Line Style

**Size**

- More Time
- Less Time
Positioning Estimates on History to Build a Defensible Position

The Next Major Release

This is a complex release. The customers desired schedule is a shorter time than the bug fix releases.
Using the raw data for prediction

84% of the historical occurrences have been inside the circle

3075 Test Cases ±500 with 84% confidence

500 Technical Requirements

All completed systems  Avg. Line Style  1 Sigma Line Style
Patterns in the Data
Based on Analysis of over 5,000 Projects

<table>
<thead>
<tr>
<th>Influencing Factor</th>
<th>Duration</th>
<th>Total Effort</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Size</td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
</tr>
<tr>
<td>Efficiency</td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
</tr>
<tr>
<td>Staffing/ Schedule</td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
</tr>
<tr>
<td>Accelerator</td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
<td><img src="#" alt="Graph" /></td>
</tr>
</tbody>
</table>

Problem - Schedule is what customers always want to compress & it is the least compressible variable !!!
2 Key Input Variables of the Estimate

Software Size
&
Development Team Efficiency
Estimating Software Size - A Key Variable

• Size is a major input to all the software estimation methods

• Estimating size is one of the most challenging aspects of the estimation process
  – Choice of appropriate measures (COTS, OO, Web)
  – You are guaranteed to be wrong no matter how well you do it
  – The state of the design is always at a higher level of abstraction than what you are trying to quantify
  – Requires input from engineers who are generally more detail oriented and like precision
  – It takes some thinking and little hard work
  – Usually must be accomplished in a short period of time
  – Must consider new, changed and plug and play
Size Estimation Practical Approaches

- Try to size the system using at least 2 different methods - confirms results and helps to quantify variability

- Use size metrics that are readily available or easy for the project engineers to relate to - eliminates resistance

- Look for relationships between different sizing artifacts they provide transforms into the units you need (there is usually good consistency within a product team)

- Sample from early prototype development when moving into a totally new environment
Example of a Transform Number of Classes per Function

Average function = 8 classes
Average Object = 343 C++ SLOC

Super heavy weight functions
Heavy weight functions
Middle weight functions
Light weight functions

Visual C++ SLOC (thousands)

Classes Per Function

Report Manager
Color Manager
Reference Group Manager
Resources
Graphics Engine

ALL Systems
Avg. Line Style
1 Sigma Line Style
2 Sigma Line Style

Average function = 8 classes
Average Object = 343 C++ SLOC
Example of transform in an OO Environment

Relationship of Methods to SLOC

19.4 C++ SLOC/Method
Partial data can provide insights
SmallTalk Billing System

<table>
<thead>
<tr>
<th>Sub-system</th>
<th>SLOC</th>
<th>Classes</th>
<th>Avg SLOC/Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Model</td>
<td>4312</td>
<td>74</td>
<td>58.27</td>
</tr>
<tr>
<td>User Interface</td>
<td>3200</td>
<td>17</td>
<td>188.24</td>
</tr>
<tr>
<td>Use Case Framework</td>
<td>1893</td>
<td>13</td>
<td>145.62</td>
</tr>
<tr>
<td>Use Cases</td>
<td>6585</td>
<td>54</td>
<td>121.94</td>
</tr>
<tr>
<td>Other Classes</td>
<td>1323</td>
<td>14</td>
<td>94.50</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>17313</td>
<td>172</td>
<td></td>
</tr>
<tr>
<td><strong>Average SLOC/CLASS</strong></td>
<td></td>
<td></td>
<td>100.66</td>
</tr>
</tbody>
</table>
Sizing Quarterly Maintenance Releases
Request For Service Sizing

Customer asks for capability in a formal request for service.

The developer quickly scopes the effort into time required to implement the request.

Average SLOC size statistics from past developments are used.

<table>
<thead>
<tr>
<th>RFS Number</th>
<th>SLOC/RFS</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt; 1 Week RFS #1</td>
<td>86.0</td>
</tr>
<tr>
<td>2</td>
<td>&lt; 1 Week RFS #2</td>
<td>86.0</td>
</tr>
<tr>
<td>3</td>
<td>&lt; 1 Month RFS #1</td>
<td>348.0</td>
</tr>
<tr>
<td>4</td>
<td>&lt; 1 Month RFS #2</td>
<td>348.0</td>
</tr>
<tr>
<td>5</td>
<td>&lt; 3 Month RFS #1</td>
<td>1043.0</td>
</tr>
<tr>
<td>6</td>
<td>&lt; 3 Month RFS #2</td>
<td>1043.0</td>
</tr>
<tr>
<td>7</td>
<td>&lt; 6 Month RFS #1</td>
<td>2007.0</td>
</tr>
<tr>
<td>8</td>
<td>&lt; 6 Month RFS #2</td>
<td>2007.0</td>
</tr>
</tbody>
</table>

Total RFS 8
SAP Implementation (Single Site)
Accounts Payable - General Ledger 161 SAP Transactions

Staffing Profile

SAP Transactions
Avg 29 hours/Transaction

RISK
Time
Effort
Uninf Cst
Min Pk Staff
Max Pk Staff
FOC MTTD

% 0 10 20 30 40 50 60 70 80 90 100

MB Life Cycle

Time 6.40 12.30 Months
Effort 4888 7028 PHR
Uninf Cst 405 582 $1000
Pk Staff 7.78 7.78 People
MTTD 1.78 7.69 Days
Start 10/10/1998 06/01/1998 Date

© Quantitative Software Management, Inc. All rights reserved. PSM Conference 2000 #27
Determining Development Team Efficiency

Developer Efficiency = \frac{\text{Functionality Developed}}{\text{Effort} \times \text{Time}}

A Higher value means less time and effort
Analyzing the Data

- Homogenous Data Set
- Try to get a sample of 4-6 projects if possible. However one is better than none!
- Sketch the staffing plan - often uncovers unique behavior
- Capture the contextual information - this helps you understand the project behavior
- Capture the core metric (size, schedule, effort, defects)
Using the Data to Learn - Frequency Distribution

Typical Capability 11.8

Outstanding Performance - Why?
1. Important customer really need a feature
2. Assembled top talent
3. Long hours
4. Design Process Improvements
Project Analysis
Schedule, Cost & Reliability Analysis
Project Estimation Process

- Develop major inputs (size & efficiency)
- Identify customer and project constraints (schedule, staffing, reliability, cost)
- Identify the “Impossible Zone” Minimum Build Schedule
- Develop alternative scenarios
  - How much can we build? Does it provide enough capability?
  - What if we add or reduce staff?
  - What efficiency is required to meet the schedule? Is there any evidence that can be achieved?
  - Should we adopt a multi release plan?
- Sensitivity Analysis - Impact if our major inputs assumptions are off
Project Analysis - Big Payoffs

- **Consider a group problem solving session (ala JAD)**
  - marketing, engineers, customer, business managers
  - Often uncovers issues but facilitates a consensus solution

- **High Bandwith Analysis ala Edward Tufte**
  - Processing information in parallel
  - Visualize interrelated data
  - Focus on bringing absolute attention to the data
Information Design
A Single View of the Data

Aha!
There was
Civilization on Mars!
Information Design
Multiple Views Provide a More Complete Picture

Oops
Analyzing Data in Parallel

Initial Analysis View

Monthly Avg Staff (people)

Gantt Chart By Sub-Phase

Monthly Cum Cost ($)

Monthly MTTD (Days)

Sensitivity of Life Duration to Size

MB Duration (Months) vs Effective SLOC

Sensitivity of Life Cost to Size

MB Effort (MM) vs Effective SLOC

Pi vs Effective SLOC

RISK GAUGE

Duration

MB Time <= 14 months  Effort: na

Effort

Peak Staff <= 15 ppl  Mtttd >= 21.65 Days

Quality

Joint Prob

CONTROL PANEL

MB Time <= 14 months  Effort: na

14.6

18.2

26.0

120.7

SOLUTION PANEL

Main

Time  15.13  25.21  months

Effort  310.49  435.08  MM

Cost  3.1  4.4  $ (M)

PI=14.6  MBI=2.8  Eff SLOC=120701

QSM

The Intelligence Behind
Successful Software Projects
Communicating Results
Key Points on Communicating the Results

- Make sure the audience understands the analysis method
- Briefing should be concise and stress a few main points
- Support your analysis with facts and historical data!!!
- Provide decision makers with recommended solution along with alternative solutions
- Provide execution work plans (basis project monitoring)
Project Monitoring & Tactical Plan Adjustments
Rate Charting to Visualize Progress

Product Construction in Objects vs Time

- Actual
- Interpolated
- Plan

Completion plan
End Date

Start Date

Date 12/31/93 (14.00 mos)

<table>
<thead>
<tr>
<th>Size (Obj)</th>
<th>Plan</th>
<th>Actual</th>
<th>%Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI</td>
<td>10.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MBI</td>
<td>1.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Variation from Plan

- PI: 10.8
- MBI: 1.2
- %Diff: -12.7

© Quantitative Software Management, Inc. All rights reserved. PSM Conference 2000 #40
Control Bounds & Acceptable Variation

Product Construction in Objects vs Time

Green Zone = Acceptable Variation
Yellow Zone = Unacceptable Variation

Date 12/31/93 (14.00 mos)

<table>
<thead>
<tr>
<th>Size (Objects)</th>
<th>Plan</th>
<th>Actual</th>
<th>%Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pi</td>
<td>147.82</td>
<td>129.00</td>
<td>-12.7</td>
</tr>
<tr>
<td>MBI</td>
<td>10.8</td>
<td>1.2</td>
<td></td>
</tr>
</tbody>
</table>

© Quantitative Software Management, Inc. All rights reserved. PSM Conference 2000 #41
Multiple Metrics Give a More Complete Picture

- Design Units (Cum)
- Integrated Code (Cum)
- Total Defect Rate
- Unit Coded (Cum)
- Size Est Variance (Rate)
- Aggregate Staffing Rate
- Unit Tested (Cum)
- Gantt Chart
- Total Cum Effort

Legend:
- Current Plan
- Actual
- Interpolated
- Green Control Bound
- Yellow Control Bound
- Life Cycle includes RQ_D, C&T

© Quantitative Software Management, Inc. All rights reserved. PSM Conference 2000 #42
Tactical Adjustments When Required

Design Units (Cum)

Unit Tested (Cum)

Aggregate Staffing Rate

Unit Coded (Cum)

Integrated Code (Cum)

Total Cum Effort

Gantt Chart

Forecast to Complete

Current Plan  Actual  Interpolated  Current Forecast  Life Cycle includes RQ_D, C&T

© Quantitative Software Management, Inc. All rights reserved. PSM Conference 2000 #43
Estimation is about Knowledge, Analysis, Communication & Negotiation

• It all boils down to who can make reasonable inferences, communicate, and negotiate

• The people who do it best are the ones that have some facts

• You always have a stronger negotiating position when they are based on fact rather than emotions

• Data is the key to good estimation

Facts are Friendly