On the Economic Evaluation of XP Projects

Frank Padberg & Matthias Müller
Universität Karlsruhe
Germany
Core XP Techniques

• Pair Programming
• Test-Driven Development (test-first)
• Incremental Delivery (small releases)
• Refactoring

© Dr. Frank Padberg 2003
Core XP Techniques

- Pair Programming
- Test-Driven Development (test-first)
- Incremental Delivery (small releases)
- Refactoring
Pair Programming

• all tasks performed by pairs of programmers using one display, keyboard, and mouse

• personnel cost basically is doubled

• claims higher team productivity and improved software quality as compared to conventional development
Test-Driven Development

- test cases written *ahead* of the code
- serve as a substitute for the specification
- test cases must be re-run continuously
- extra effort for writing, running, and updating the tests
- claims improved code quality as compared to conventional development
Incremental Delivery

- subdivide software into pieces
- assemble and deliver (small) releases as soon as possible
- might get early partial payment
- claims early delivery of value to customer and improved feedback to developers as compared to conventional development
Research Question

For which project settings does the extra cost of applying the XP techniques get balanced by their benefits?
Economic Modeling

- model the business value of a project
- include XP techniques, market pressure, workforce size, product size
- compute development time and cost
Study Approach

• fix some project setting
• compute development time and cost when using XP
• compute development time and cost when using conventional development
• compare value of XP project with value of conventional project
Net Present Value

• returns which are realized earlier are more valuable than returns realized later

• hence, the dollar returns of a project are discounted back at a certain rate

• use large values for the discount rate to model strong market pressure
Net Present Value Computation

NPV = \frac{AssetValue}{(1 + DiscountRate)^{DevTime}} - DevCost

- AssetValue: dollars paid upon completion
- discount back from time of project completion to time zero
- subtract development cost
Modeling Pair Programming
Pair Speed Advantage

\[
\text{PSA} = \frac{\text{time required by single programmer}}{\text{time required by programmer pair}}
\]

- average figure, for some "unit" task
- Nosek (1998) reports 1.4
- Williams e.a. (2000) report 1.8
Pair Defect Advantage

\[
PDA = 1 - \frac{\text{defect density of pair programming}}{\text{defect density of conventional development}}
\]

- average figure
- Williams e.a. (2000) report 15 percent
Development Time: Conventional Project

DevTime\(_C\) = \(\frac{1}{12} \times \frac{\text{ProductSize}}{\text{Productivity} \times \text{NumOfDevelopers}}\) + QATime

- *additional* QA needed to compensate defect advantage of Pair Programming
- QATime proportional to PairDefectAdvantage

© Dr. Frank Padberg 2003
Additional Quality Assurance

\[
\text{QATime} = \frac{1}{12} \times \frac{\text{DefectRemovalTime}}{\text{WorkTime} \times \text{NumOfDevelopers}} \times \text{ProductSize} \times \text{DefectDensity} \times \text{PairDefectAdvantage}
\]

© Dr. Frank Padberg 2003
Development Time: Pair Programming

\[
\text{DevTime}_{PP} = \frac{1}{12} \times \frac{\text{ProductSize}}{\text{Productivity} \times \text{NumOfPairs}} \times \frac{1}{\text{PairSpeedAdvantage}}
\]

- pair programming and speed advantage enter
- no additional QA
Development Cost

\[ \text{DevCost}_C \sim \text{DevTime}_C \times \text{NumOfDevelopers} \]

\[ \text{DevCost}_{PP} \sim \text{DevTime}_{PP} \times 2 \times \text{NumOfPairs} \]
Results: Pair Programming
# Sample Project: Fixed Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Productivity</td>
<td>350 LOC/month</td>
</tr>
<tr>
<td>DefectDensity</td>
<td>0.03 defects/LOC</td>
</tr>
<tr>
<td>DefectRemovalTime</td>
<td>10 hours/defect</td>
</tr>
<tr>
<td>ProductSize</td>
<td>16,800 LOC</td>
</tr>
<tr>
<td>TaskLimit</td>
<td>8</td>
</tr>
<tr>
<td>AssetValue</td>
<td>1,000,000 dollars</td>
</tr>
<tr>
<td>DeveloperSalary</td>
<td>50,000 dollars/year</td>
</tr>
<tr>
<td>LeaderSalary</td>
<td>60,000 dollars/year</td>
</tr>
<tr>
<td>WorkTime</td>
<td>135 hours/month</td>
</tr>
<tr>
<td>NumOfDevelopers</td>
<td>8</td>
</tr>
</tbody>
</table>

© Dr. Frank Padberg 2003
## Sample Project: Variable Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>PairSpeedAdvantage</td>
<td>1.4 .... 1.8</td>
</tr>
<tr>
<td>PairDefectAdvantage</td>
<td>5 % .... 25 %</td>
</tr>
<tr>
<td>DiscountRate</td>
<td>25 % .... 100 % per year</td>
</tr>
<tr>
<td>NumOfPairs</td>
<td>4 .... 8</td>
</tr>
</tbody>
</table>
Sample Project: Limited Workforce

<table>
<thead>
<tr>
<th>PSA</th>
<th>PDA</th>
<th>NPV&lt;sub&gt;C&lt;/sub&gt;</th>
<th>NPV&lt;sub&gt;PP&lt;/sub&gt;</th>
<th>rel. adv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4</td>
<td>15%</td>
<td>626,026</td>
<td>524,093</td>
<td>-16%</td>
</tr>
<tr>
<td>1.8</td>
<td>15%</td>
<td>626,026</td>
<td>627,851</td>
<td>1%</td>
</tr>
<tr>
<td>1.8</td>
<td>25%</td>
<td>600,509</td>
<td>627,851</td>
<td>5%</td>
</tr>
</tbody>
</table>

- only eight developers (8 single vs. 4 pairs)
- moderate discount rate of 25 percent
Sample Project: Limited Workforce (cont.)

<table>
<thead>
<tr>
<th>PSA</th>
<th>PDA</th>
<th>NPV&lt;sub&gt;C&lt;/sub&gt;</th>
<th>NPV&lt;sub&gt;PP&lt;/sub&gt;</th>
<th>rel. adv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4</td>
<td>15%</td>
<td>474,817</td>
<td>431,932</td>
<td>−30%</td>
</tr>
<tr>
<td>1.8</td>
<td>15%</td>
<td>474,817</td>
<td>477,233</td>
<td>1%</td>
</tr>
<tr>
<td>1.8</td>
<td>25%</td>
<td>441,177</td>
<td>477,233</td>
<td>8%</td>
</tr>
</tbody>
</table>

- only eight developers (8 single vs. 4 pairs)
- **high** discount rate of 75 percent
Sample Project: Strong Market Pressure

<table>
<thead>
<tr>
<th>PSA</th>
<th>PDA</th>
<th>$\text{NPV}_C$</th>
<th>$\text{NPV}_{PP}$</th>
<th>rel. adv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4</td>
<td>5%</td>
<td>508,803</td>
<td>511,700</td>
<td>1%</td>
</tr>
<tr>
<td>1.4</td>
<td>25%</td>
<td>441,177</td>
<td>511,700</td>
<td>16%</td>
</tr>
<tr>
<td>1.8</td>
<td>5%</td>
<td>508,803</td>
<td>617,141</td>
<td>21%</td>
</tr>
<tr>
<td>1.8</td>
<td>25%</td>
<td>441,177</td>
<td>617,141</td>
<td>40%</td>
</tr>
</tbody>
</table>

- **maximum** workforce (8 single vs. 8 pairs)
- high discount rate of 75 percent

© Dr. Frank Padberg 2003
Break-Even Discount Rate

• measures how strong the market pressure must be for Pair Programming to break even with conventional development in a given project setting

• BDR solves the equation:

\[ \text{NPV}_{PP} (\text{DiscountRate}) = \text{NPV}_{C} (\text{DiscountRate}) \]

• depends on PSA and PDA
Break-Even Discount Rate Dependent on Pair Speed and Defect Advantage
Break-Even Discount Rate Dependent on Pair Speed Advantage

looks much like an exponential curve

© Dr. Frank Padberg 2003
Log-Linear Regression

log(BreakEvenDiscountRate) depends approx. linearly on PSA
Impact of Pair Defect Advantage on Break-Even Discount Rate

6 programmer pairs; defect advantage 5, 15, 25 percent

© Dr. Frank Padberg 2003
Impact of Pair Defect Advantage (cont.)

- the larger the defect advantage, the smaller the speed advantage and discount rate required to break even (relative position of regression lines)
- impact of the speed advantage is stronger for large values of the defect advantage (slope of regression lines)
Impact of Number of Pairs on Break-Even Discount Rate

defect advantage 15 percent; 5 .... 8 pairs
Impact of Number of Pairs (cont.)

- the larger the workforce of pairs, the smaller the speed advantage and discount rate required to break even (relative position of regression lines)
- impact of the speed advantage is stronger for small number of pairs (slope of regression lines)
Observation

The stronger the market pressure, the smaller are the number of pairs, the speed advantage, and the defect advantage which are required for Pair Programming to break even.
Adding Test-Driven Development

© Dr. Frank Padberg 2003
XP Speed Factor

- almost no empirical results about speed impact of test-driven development
- first evidence suggests: test-first likely to slow development down (Müller & Hagner 2002)
- replace PairSpeedAdvantage by more general XPSpeedFactor in the model
XP Speed Factor (cont.)

\[ XPSF = \frac{\text{time required by single programmer}}{\text{time required by pair using test-first}} \]

- \( XPSpeedFactor \leq \text{PairSpeedAdvantage} \)
- \( \text{TestDrivenSpeedFactor} \leq XPSpeedFactor \)
XP Defect Factor

- no empirical results about quality impact of test-driven development
- expectation: test-first likely to improve code quality
- replace PairDefectAdvantage by more general XPDefectFactor in the model
XP Defect Factor (cont.)

\[
\text{XPDefectFactor} = 1 - \frac{\text{time required by pair using test-first}}{\text{defect density of conventional development}}
\]

- don’t really know upper bound
- PairDefectAdvantage \( \leq \) XPDefectFactor

© Dr. Frank Padberg 2003
Extended Economic Model

- replace pair speed and defect advantage by more general \texttt{XPSpeedFactor} and \texttt{XPDefectFactor} in the formulas
- sensitivity analysis remains the same
- conclusions and guidelines are very similar
Some Guidelines
Market Pressure

Consider using Pair Programming and Test-First given that the market pressure is really strong and your programmers are much more efficient when working in pairs as compared to working alone.
Size of Workforce

If the size of your workforce does not allow you to run the project with the maximum number of pairs, it might be more efficient to add single developers instead of using pairs.
Topics Not Covered

- Incremental Delivery (not shown)
- Refactoring (working on this)
- Brook’s Law (working on this)
Topics Not Covered (cont.)

- management problems for larger XP projects:
  - project control
  - controlling the requirements
  - maintaining a good design
  - staff turnover

© Dr. Frank Padberg 2003
Publications

- *Analyzing the Cost and Benefit of Pair Programming*
  (with M. Müller)

- *On the Economic Evaluation of XP Projects*
  (with M. Müller)

- *Experiment About Test-First Programming*
  (by M. Müller and O. Hagner)
Thank You!