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

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

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

- 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

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

- average figure
- Williams e.a. (2000) report 15 percent

Development Time: Conventional Project

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

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

Additional Quality Assurance

$$\begin{aligned} \text{QATime} &= \frac{1}{12} \times \frac{\text{DefectRemovalTime}}{\text{WorkTime} \times \text{NumOfDevelopers}} \\ &\times \text{ProductSize} \times \text{DefectDensity} \\ &\times \text{PairDefectAdvantage} \end{aligned}$$

Development Time: Pair Programming

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

- pair programming and speed advantage enter
- no additional QA

Development Cost

$$\mathsf{DevCost}_{\mathsf{C}} \sim \mathsf{DevTime}_{\mathsf{C}} \times \mathsf{NumOfDevelopers}$$

$$\mathsf{DevCost}_{\mathsf{PP}} \sim \mathsf{DevTime}_{\mathsf{PP}} \times 2 \times \mathsf{NumOfPairs}$$

Results: Pair Programming

Sample Project: Fixed Parameters

Productivity	350 LOC/month
DefectDensity	0.03 defects/LOC
DefectRemovalTime	10 hours/defect
ProductSize	16,800 LOC
TaskLimit	8
AssetValue	1,000,000 dollars
DeveloperSalary	50,000 dollars/year
LeaderSalary	60,000 dollars/year
WorkTime	135 hours/month
NumOfDevelopers	8

Sample Project: Variable Parameters

PairSpeedAdvantage	1.4 .... 1.8
PairDefectAdvantage	5 % .... 25 %
DiscountRate	25 % .... 100 % per year
NumOfPairs	4 8

Sample Project: Limited Workforce

PSA	PDA	NPV_C	NPV_{PP}	rel. adv.
1.4	15 %	626,026	524,093	−16 %
1.8	15 %	626,026	627,851	1 %
1.8	25 %	600,509	627,851	5 %

- only eight developers (8 single vs. 4 pairs)
- moderate discount rate of 25 percent

Sample Project: Limited Workforce (cont.)

PSA	PDA	NPV_C	NPV_{PP}	rel. adv.
1.4	15 %	474,817	431,932	−30 %
1.8	15 %	474,817	477,233	1 %
1.8	25 %	441,177	477,233	8 %

- only eight developers (8 single vs. 4 pairs)
- high discount rate of 75 percent

Sample Project: Strong Market Pressure

PSA	PDA	NPV_C	NPV_{PP}	rel. adv.
1.4	5 %	508,803	511,700	1 %
1.4	25 %	441,177	511,700	16 %
1.8	5 %	508,803	617,141	21 %
1.8	25 %	441,177	617,141	40 %

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

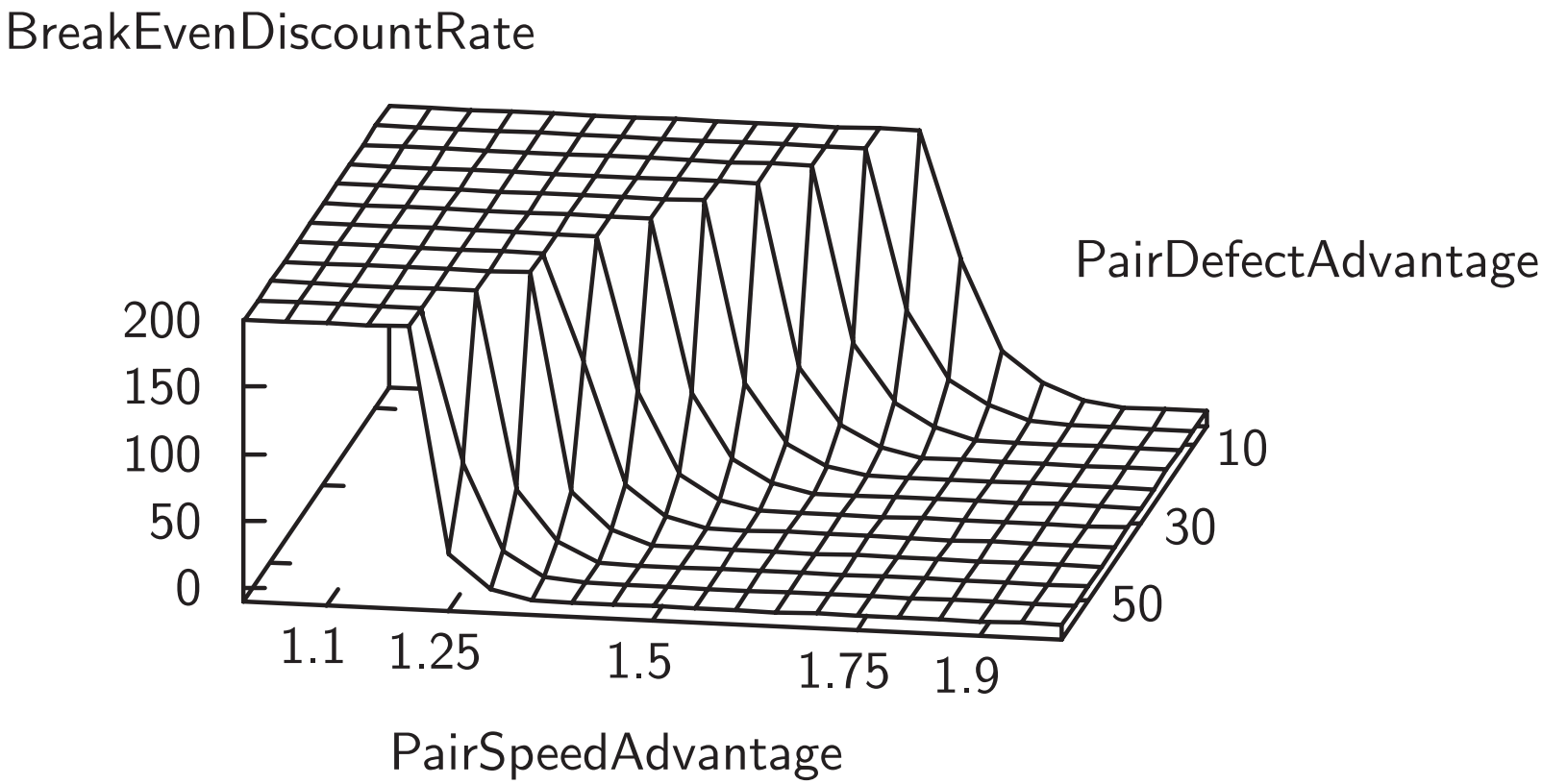
Break-Even Discount Rate

- 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:

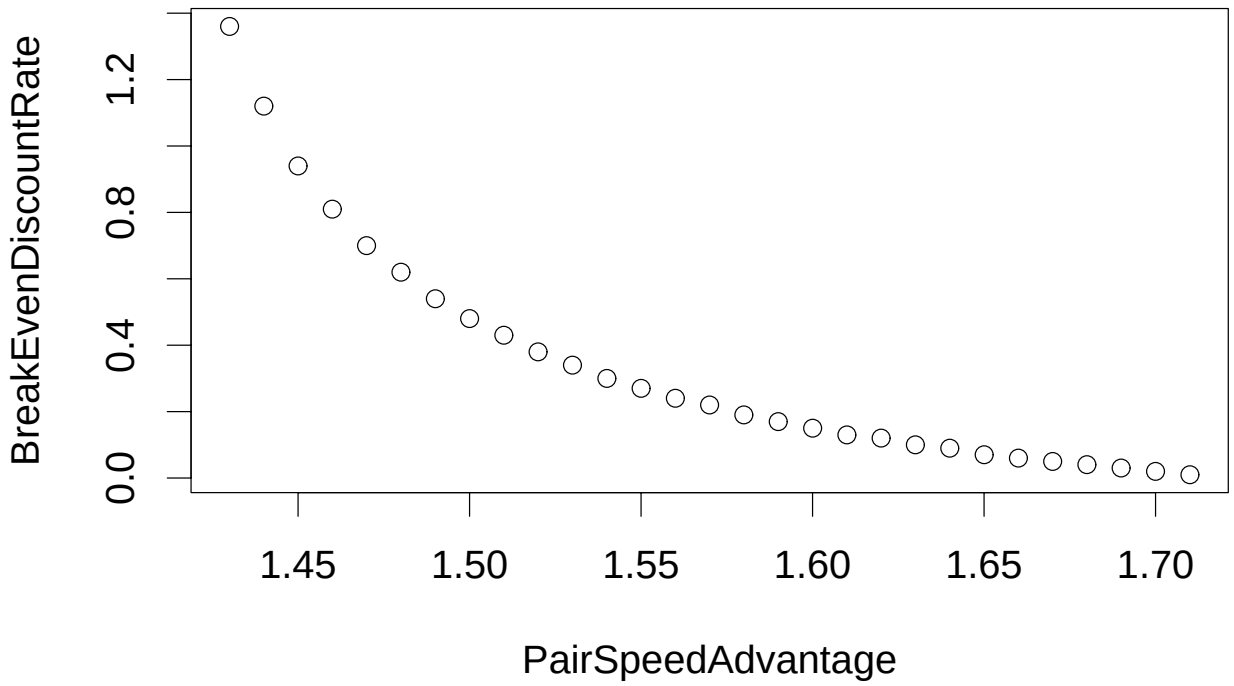
$$\mathrm{NPV}_{\mathrm{PP}}\left(\mathrm{DiscountRate}\right) \;=\; \mathrm{NPV}_{\mathrm{C}}\left(\mathrm{DiscountRate}\right)$$

- depends on PSA and PDA

Break-Even Discount Rate Dependent on Pair Speed and Defect Advantage

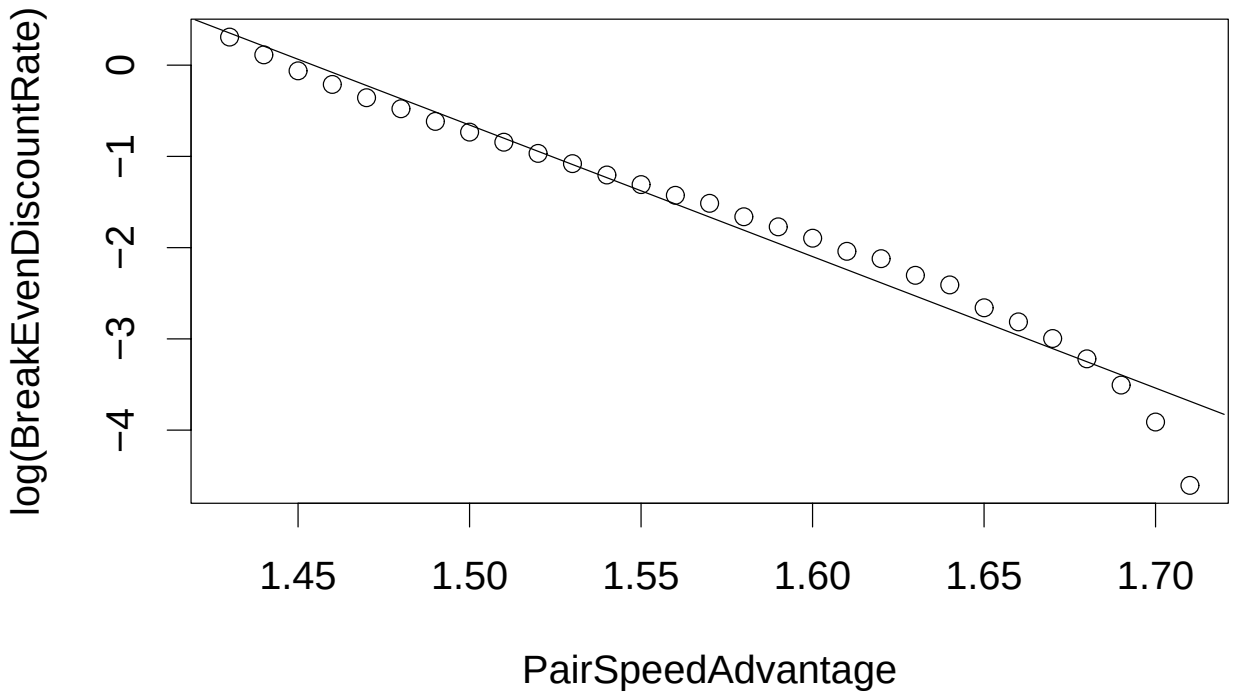


Break-Even Discount Rate Dependent on Pair Speed Advantage



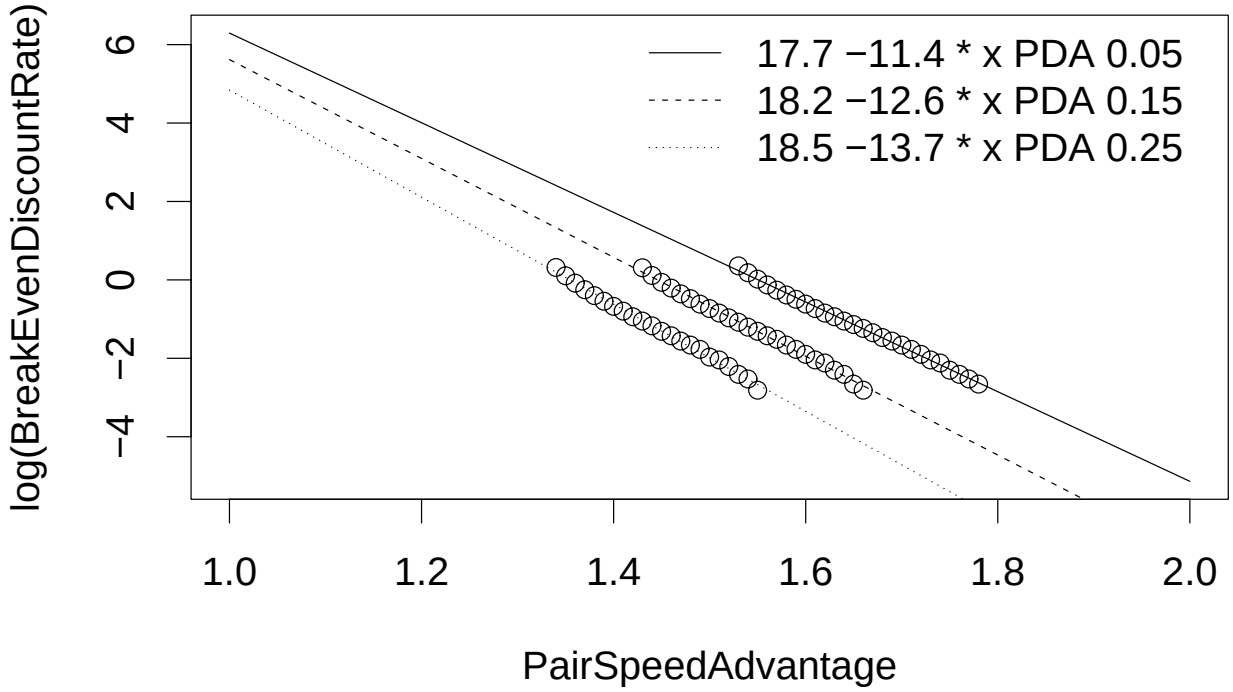
looks much like an exponential curve

Log-Linear Regression



logarithm of BDR depends approx. linearly on PSA

Impact of Pair Defect Advantage on Break-Even Discount Rate

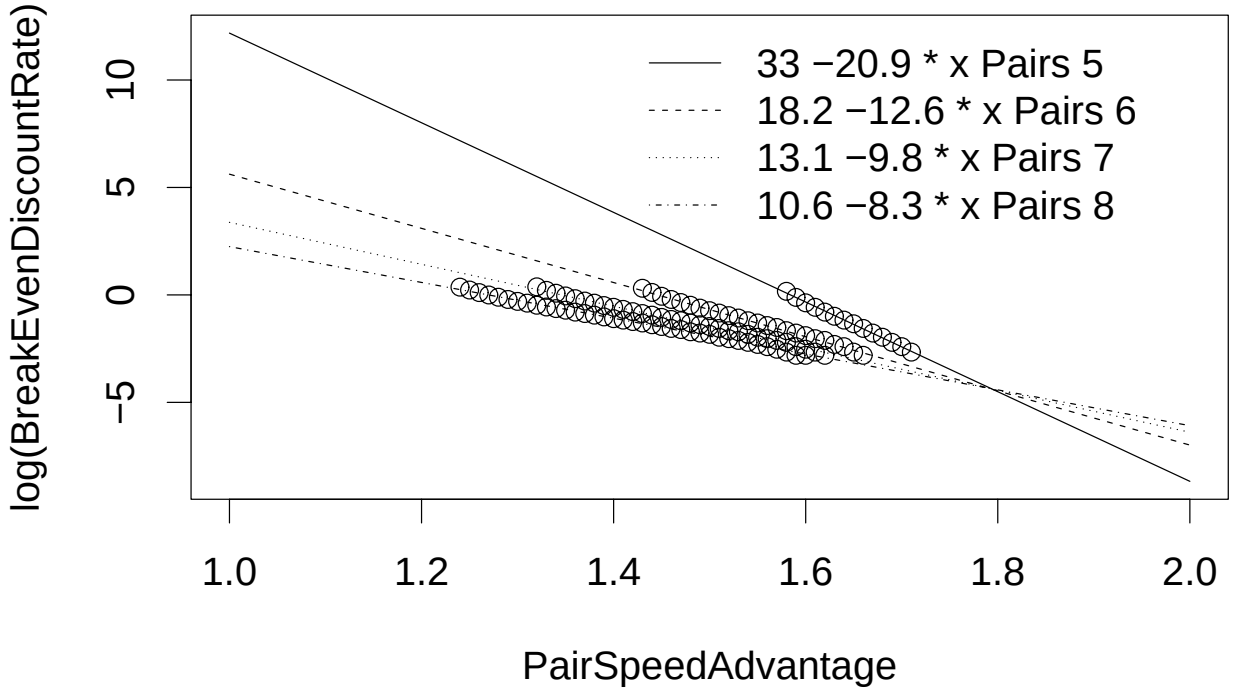


6 programmer pairs; defect advantage 5, 15, 25 percent

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

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.)

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

- XPSpeedFactor $\leq$ PairSpeedAdvantage
- 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.)

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

- don't really know upper bound
- PairDefectAdvantage $\leq$ XPDefectFactor

Extended Economic Model

- replace pair speed and defect advantage by more general XPSpeedFactor and 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

Publications

- *Analyzing the Cost and Benefit of Pair Programming*
  International Symposium on Software Metrics METRICS (2003)
  (with M. Müller)

- *On the Economic Evaluation of XP Projects*
  European Software Engineering Conference ESEC (2003)
  (with M. Müller)

- *Experiment About Test-First Programming*
  IEE Proceedings on Software 149:5 (2002) 131–136
  (by M. Müller and O. Hagner)

Thank You !