

CMRC-NRC

Institut de
technologie
de l'information

The Economics of Agile Software Development

Hakan Erdogmus

Software Engineering Group

May 26, 2005



Conseil national
de recherches Canada

National Research
Council Canada

Canada

CMRC-NRC

Institut de
technologie
de l'information

***Extreme Programming can create ten times more
value than a heavy-weight process.***

Kent Beck, XP-Universe 2001

- Are ASD methods economically viable?
- Which practices and principles add value?
- Why?

Outline

- **The Concept of Value (Terms of Reference)**
- **Tactical Value Sources in ASD: Efficiency and Time Value**
 - Incremental Delivery
 - Pair Programming
- **Strategic Value Sources in ASD: Optionality**
 - Just-in-Time Decision Making (Reactive)
 - Iterative & Incremental Development (Proactive)
- **Putting It All Together**
 - Test-Driven Development
 - A Model of Value Creation for ASD

What's Value?

- **Economic Value**
- **(Net) Value = Benefits – Costs**
- **Either decrease costs or increase benefits**
- **Not always quantifiable**

Characteristics of Value

- **Value is uncertain: because both costs and benefits are uncertain**
 - Value = Expected Value
 - A statistical artifact
- **Value is Time-Dependent**
 - Which one is worth more: \$100,000 received in 10 years or \$10,000 received today?
 - Value = Today's Value = **Present Value**

Copyright © NRC, 2005

Whose Value?

Customer's or Developer's?

Doesn't matter! Just assume developer is equitably compensated proportional to benefits realized by the customer

Developer's Profit = $y\%$ [(Customer's Benefit) – (Developer's Cost)]

Customer's Value

= $(1 - y)\%$ [(Customer's Benefit) – (Developer's Cost)]

Minimize development costs and maximize customer benefit!

Copyright © NRC, 2005

Sources of Value in ASD

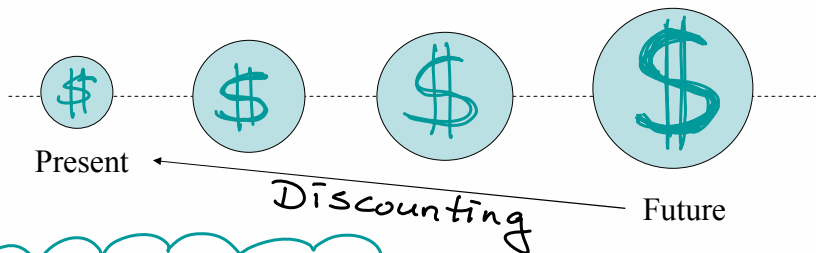
Tactical

- (Process) Efficiency
- Time Value

Strategic

- Optionality

Time Value



RECEIVE EARLY!

SPEND LATE!

Process Efficiency

- *Doing the best and most one can in the shortest time*
- Usually independent of external uncertainty
- *Mainly* driven by cost savings or cost avoidance

- **Efficiency = Productivity + Quality**

- **Productivity:** output per unit time
(something to do with speed of production)
- **Quality:** lack of defects that need repair, incur some latent cost, or prevent anticipated benefits from being realized
(something to do with avoidance of rework and a prerequisite for usability)

Copyright © NRC, 2005

Pair Programming – A Tactical Practice

Analysis based on empirical study by Laurie Williams
comparing Solo and Pair programmers

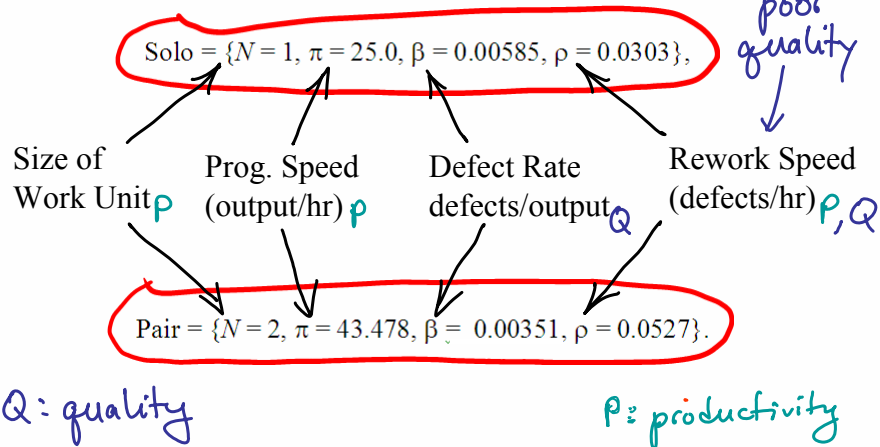
- Pairs spent on average 15% more total effort than soloists to complete the same programming task
- Code written by pairs on average passed 15% more of the specified acceptance tests (~~90%~~ vs. 75%) compared to code written by soloists **85%***
- Other studies report varying results; all claim PP is a costly proposition; none convincingly refutes the quality advantage

** Compensates for conservative error in analysis!*

Copyright © NRC, 2005

PP – Initial Model

Results converted to simple model using industry benchmarks



Copyright © NRC, 2005

PP – Economic Measure

IS PP REALLY COSTLY?

- **Benefit = (Benefit per Unit of Output) x (Output)**

Unit Value (UV)

- **Net Present Value (NPV) = PV(Benefits) – PV(Costs)**

Measure of Profitability

- **Breakeven Unit Value (BUV) = min { UV | $0 \leq NPV$ }**

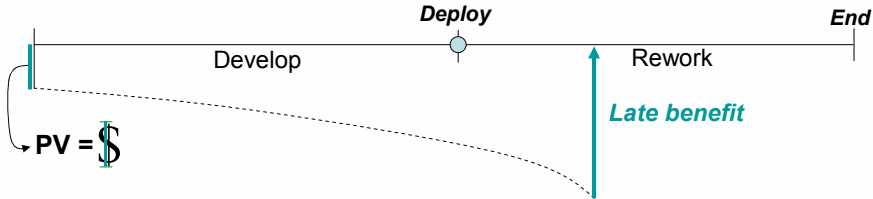
THE SMALLER, THE BETTER!

- **Compare BUVs for different benefit realization models**

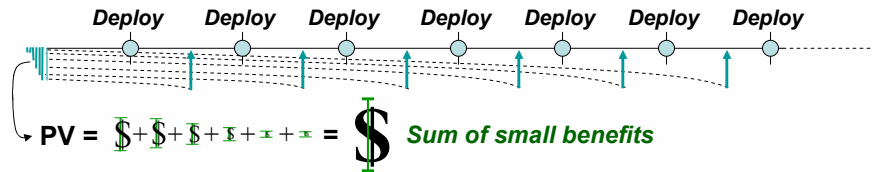
Copyright © NRC, 2005

Time Value & Benefit Realization

Infrequent -> Single-Point: Late benefits, all at once or in big chunks



Incremental -> Continuous: Early benefits, in small chunks



Copyright © NRC, 2005

PP - Results

- Overall best (lowest) BUV: Pair under *Continuous Delivery*

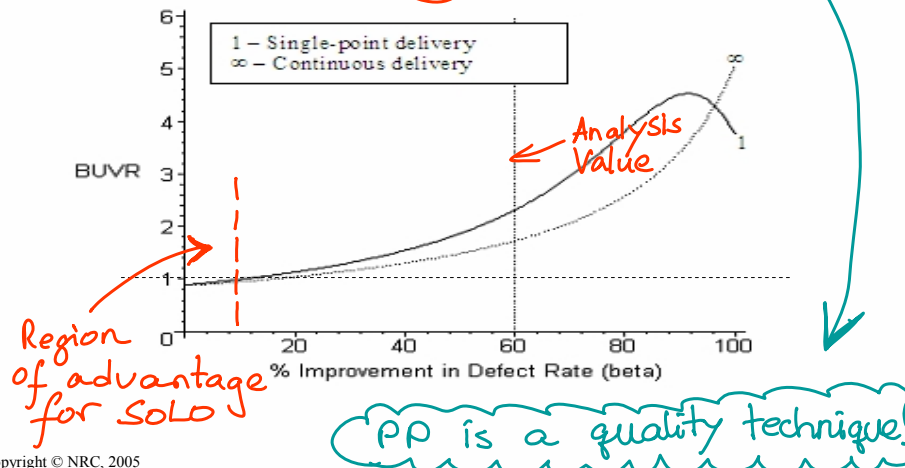
$$BUVR = \frac{BUV_{solo}}{BUV_{pair}} \quad \left\{ \begin{array}{l} > 1 \Rightarrow \text{Pair } \checkmark \\ < 1 \Rightarrow \text{Solo } \checkmark \end{array} \right.$$

Value Realization Model	BUVR Behavior	
	Single-Point Delivery	Continuous Delivery
Discount rate (r) increases $r \rightarrow 0$	BUVR increases BUVR = 2.24	BUVR = 1.73
Output (ω) increases	BUVR increases	
Overall better model	Pair \checkmark	Pair \checkmark

Copyright © NRC, 2005

PP – Sensitivity Analysis

Most sensitive to quality!



Copyright © NRC, 2005

Optionality

- **Ability to respond to change in an optimal manner**
- **Materializes only under uncertainty**
- **Mainly driven by benefits maximization**

- **Optionality = Learning + Flexibility**

- **Learning:** awareness of how uncertainty is resolved
 - Feedback ⇒ Learning
- **Flexibility:** taking advantage of learning outcomes
- **Optimal:** maximizes value given the information available at the time of the decision

Copyright © NRC, 2005

Uncertainly, Learning, Flexibility

- **No learning** \Rightarrow **Flexibility is useless**

No difference between deciding now or later, thus might as well make the decision now

- **No flexibility** \Rightarrow **Learning is useless**

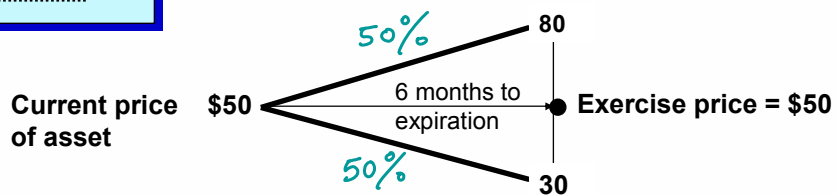
No choice any way, so why bother?

- **No uncertainty** \Rightarrow **No need for flexibility nor learning**

Copyright © NRC, 2005

Call Option

Why is optionality valuable?



Payoffs	If price rises to \$80	If price falls to \$30	$E[\text{Value}]$
without option	$\$80 - \$50 = \$30$	$\$30 - \$50 = -\$20$	5
with option	\$30.00 Option Exercised	\$0.00 Option Foregone	15

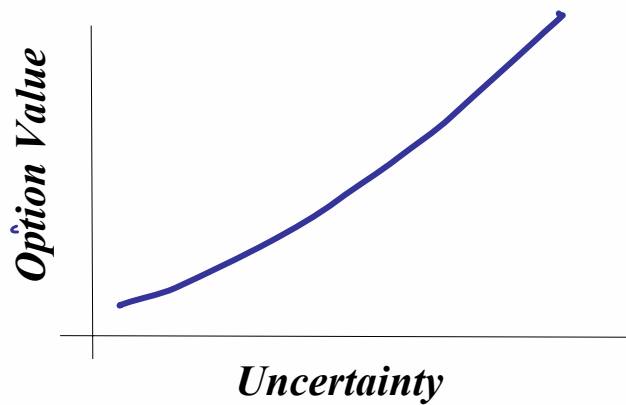
Option Pricing: How much should I pay to acquire this option now?

Copyright © NRC, 2005

Real Options Analysis

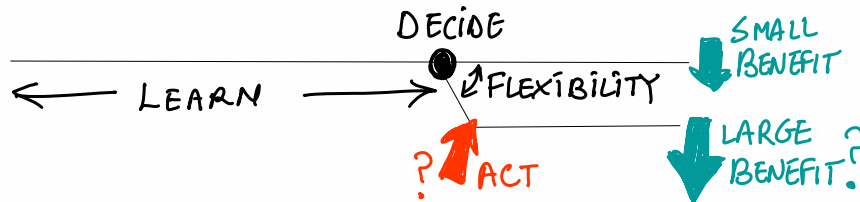
- Decision Trees (decision theory and economics)
- Option Pricing (finance)
- Combination of DT and OP

Value of options most sensitive to uncertainty



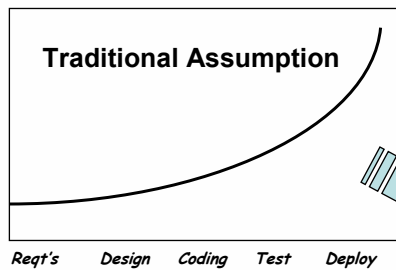
Reactive Optionality

- Wait for new information, react when appropriate



Copyright © NRC, 2005

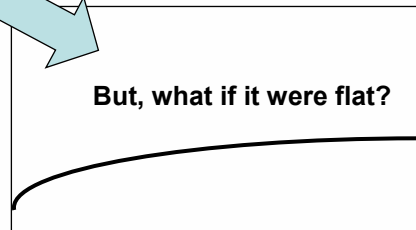
Value of Just-in-Time Decision Making



Req't's Design Coding Test Deploy

Upfront Planning

Development Decisions:
Cost of Change?



Time *JIT planning*

Copyright © NRC, 2005

YAGNI is an Example of JIT Decision Making

You are thinking of implementing a new feature*...

You could implement the
feature today, at a cost
of \$10

You think it might be
worth \$15 in benefits...
We'll know in a year!

Great uncertainty



Upside scenario:
Customer is grateful



Downside scenario:
Customer couldn't
care less—you
wasted time and
money

*Source: Beck, Extreme Programming Explained (2000)

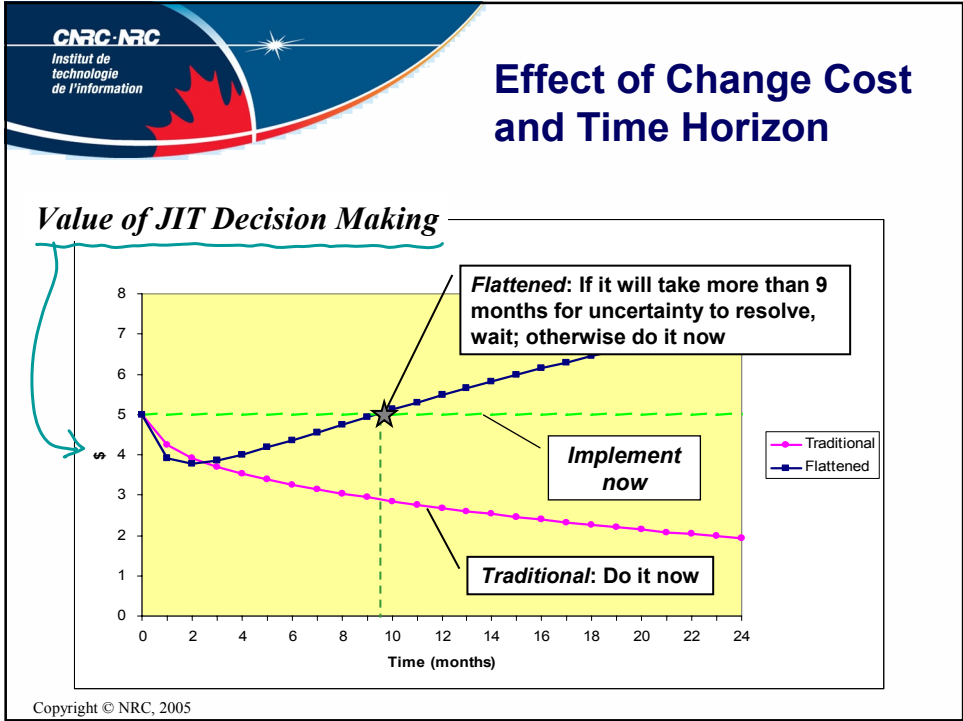
Copyright © NRC, 2005

Option Value of YAGNI (Purely Reactive Model)

Constant Cost of Change

\$15.00	PV of deferrable feature's expected benefits
\$10.00	Cost of implementation
\$5.00	Net Present Value of immediate implementation
5%	The risk-free interest rate
1.0	Years until implementation decision must be taken
100%	Volatility (uncertainty) of feature's benefit
\$7.5	Call-option value of deferred implementation

Copyright © NRC, 2005



CMRC-NRC
Institut de technologie de l'information

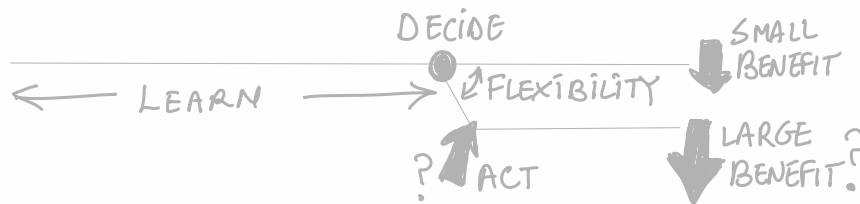
Rules of Thumb for JIT Decision Making

		<u>Cost of Change</u>		
		Traditional	Flattened	Constant
<u>Uncertainty / Time Horizon</u>	High / Long	Sooner	Later	Later
	Low / Short	Sooner	Sooner	Later

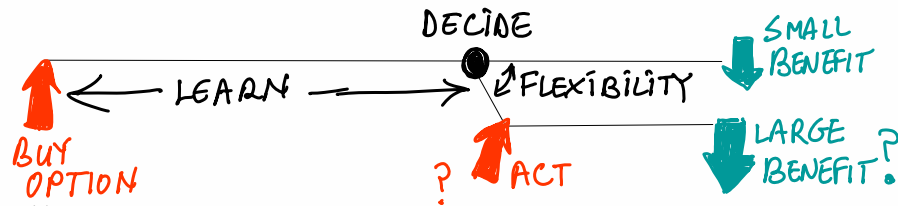
Copyright © NRC, 2005

Proactive Optionality

- Reactive



- Proactive: do something now to be able to react later

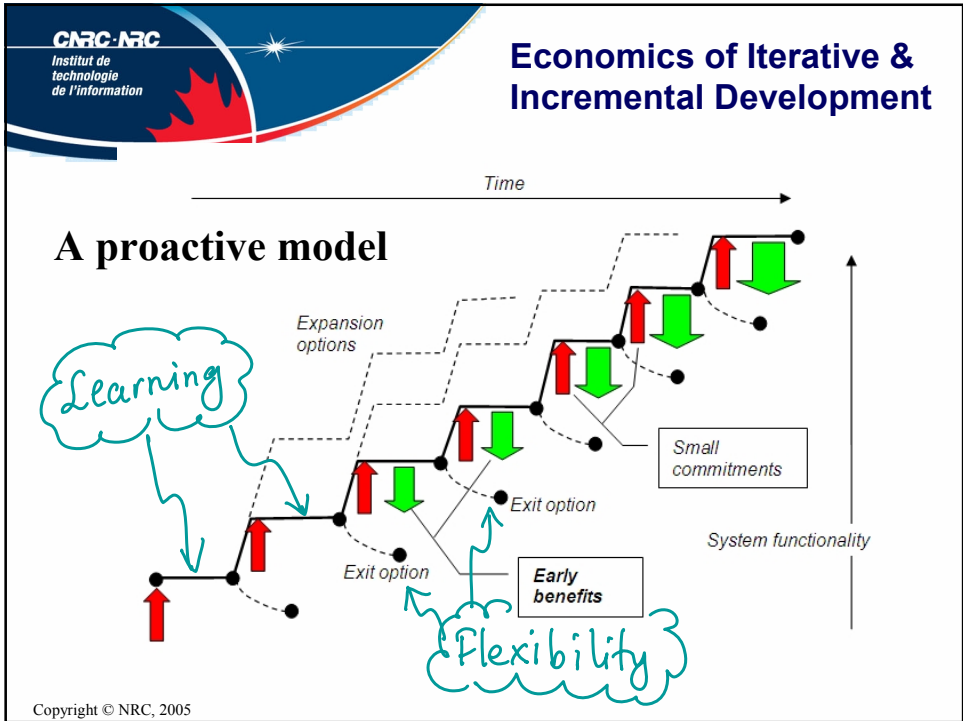
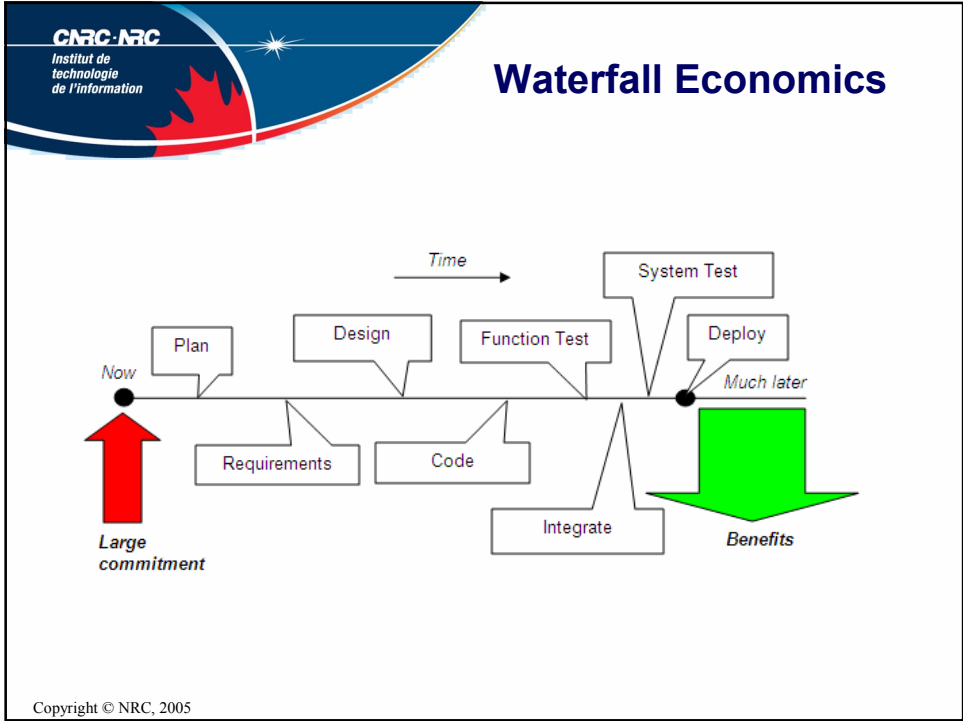


Copyright © NRC, 2005

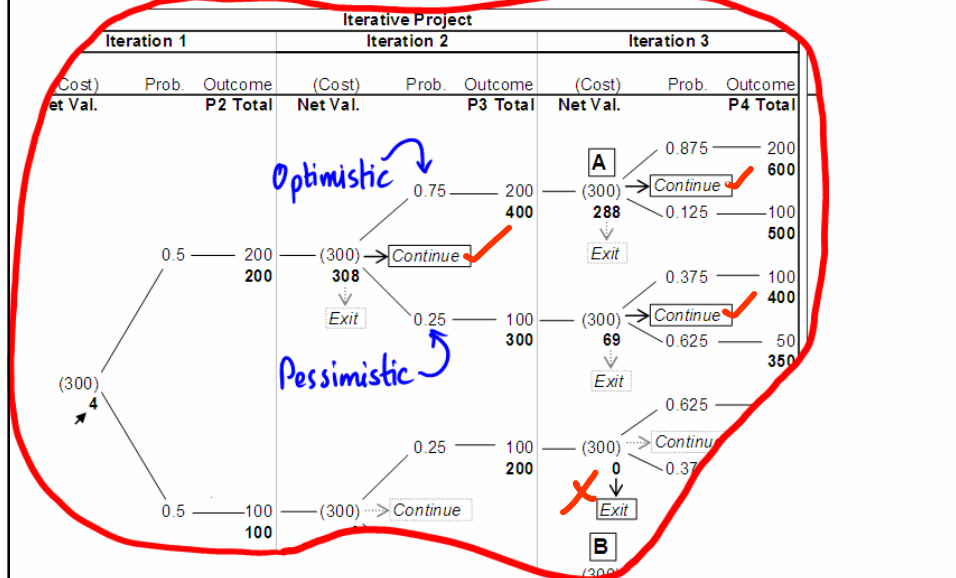
Iterative & Incremental Development

- Iterative: Repeat essentially the same process
- Incremental: Deliver usable functionality in chunks
- *Already discussed tactical value of incremental development due to time value effects*
- *Now focus on strategic value due to iterative aspect*

Copyright © NRC, 2005



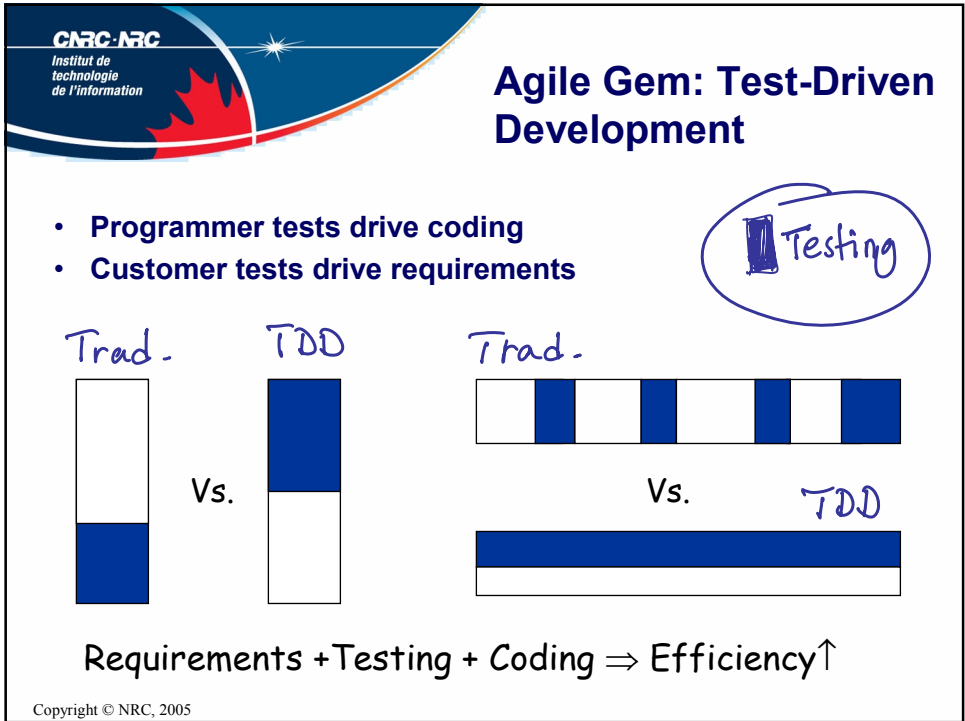
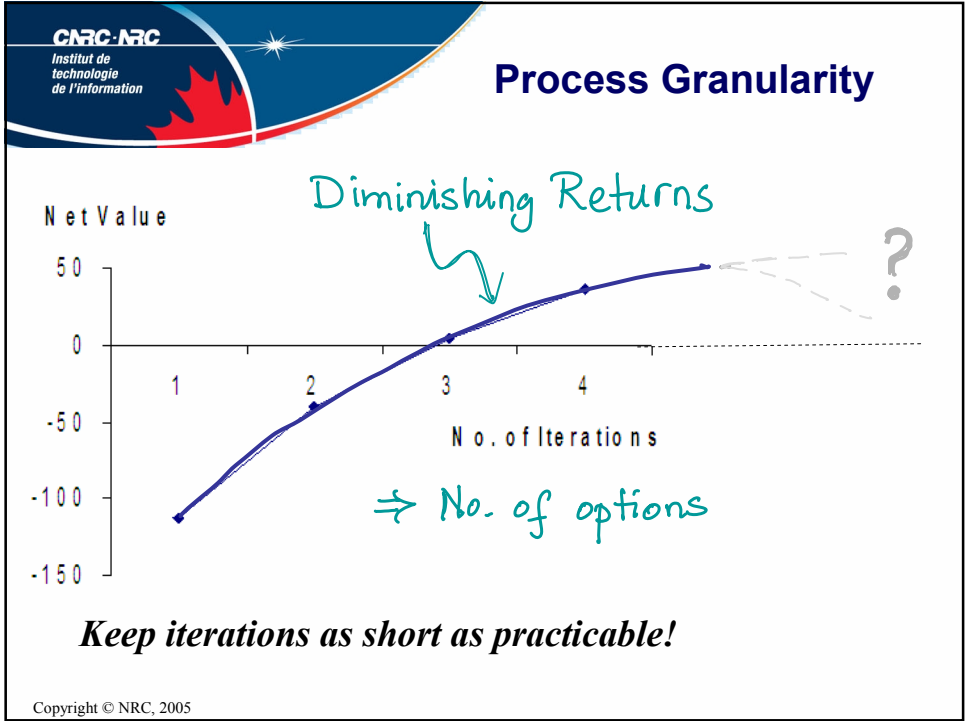
Valuation of an Iterative Project with DTs



Learning Power

- The capacity of a project stage to resolve the project's uncertainty
- Each successive iteration decreases the variance of project's benefits according to the iteration's outcome
 - Good outcome => Prob. of good outcome increases
 - Poor outcome => Prob. of good outcome decreases
 - E.g. pharma development, venture capital, nat. resources development
- Learning power => Value of iterative project

Tackle show stoppers and proof of concepts early!



TDD Empirical Results

Focus on tactical value

Authors Year	Type	Quality	Productivity	Incremental Development?	Tests required?	S/P
Müller+, 2001	CE	∅	∅	N	Y	S
Williams+, 2003	CS	+	∅	N	N	P
George+, 2003	CE	+	-	N	N	P
Edwards, 2003	CS	+	?	N	?	S
Erdogmus+, 2004	CE	∅	+	Y	Y	S

CE: Controlled Experiment
CS: Case Study
MA: Meta-Analysis

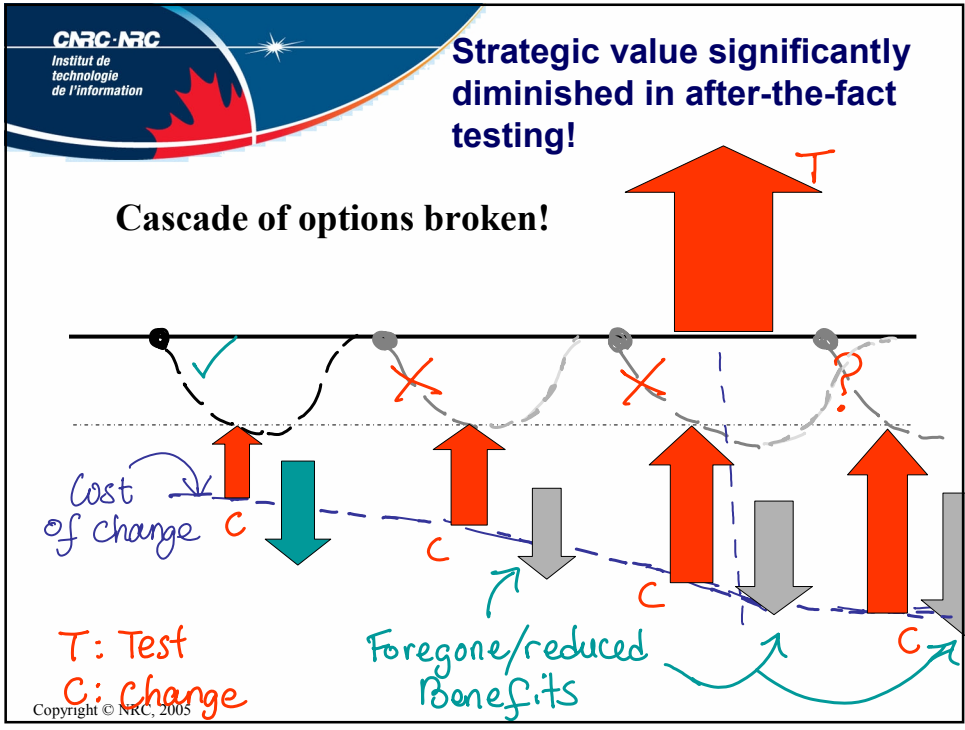
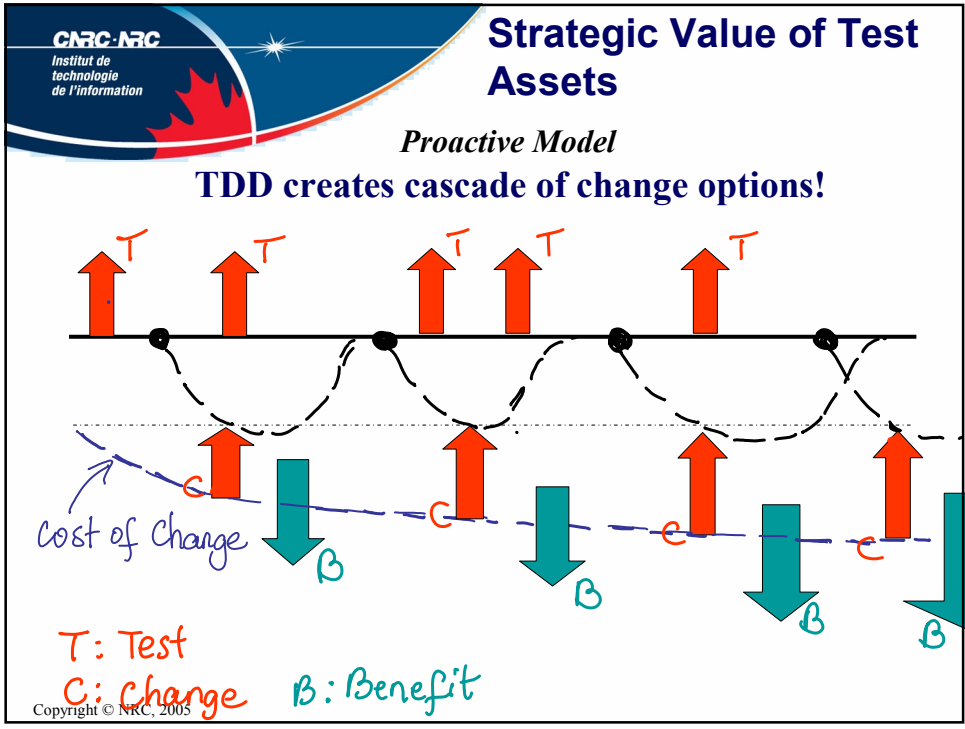
S: Students
P: Professionals

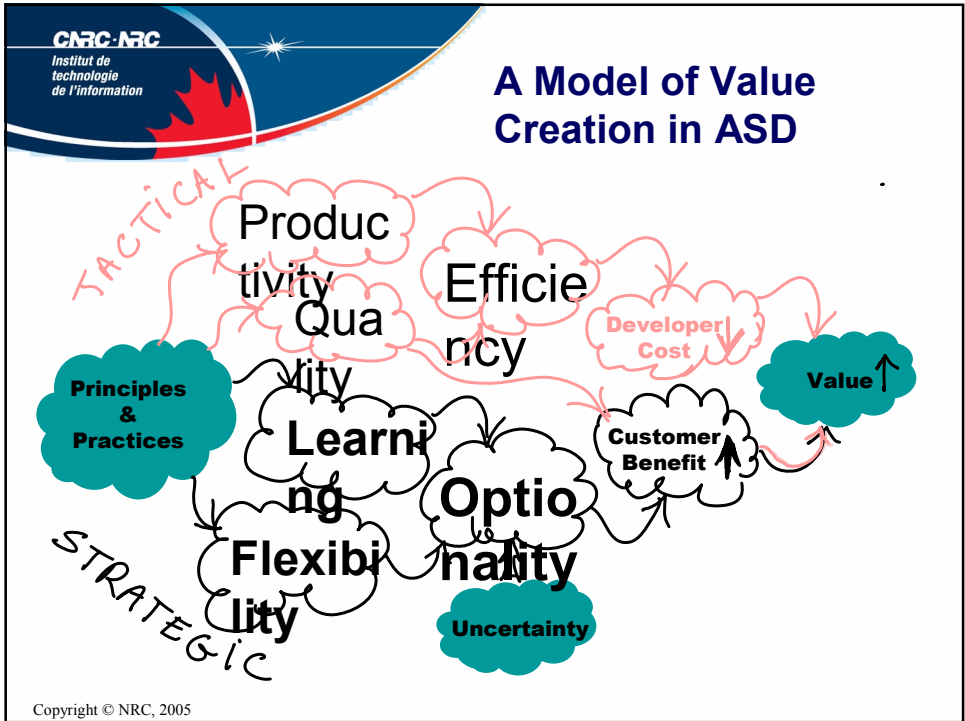
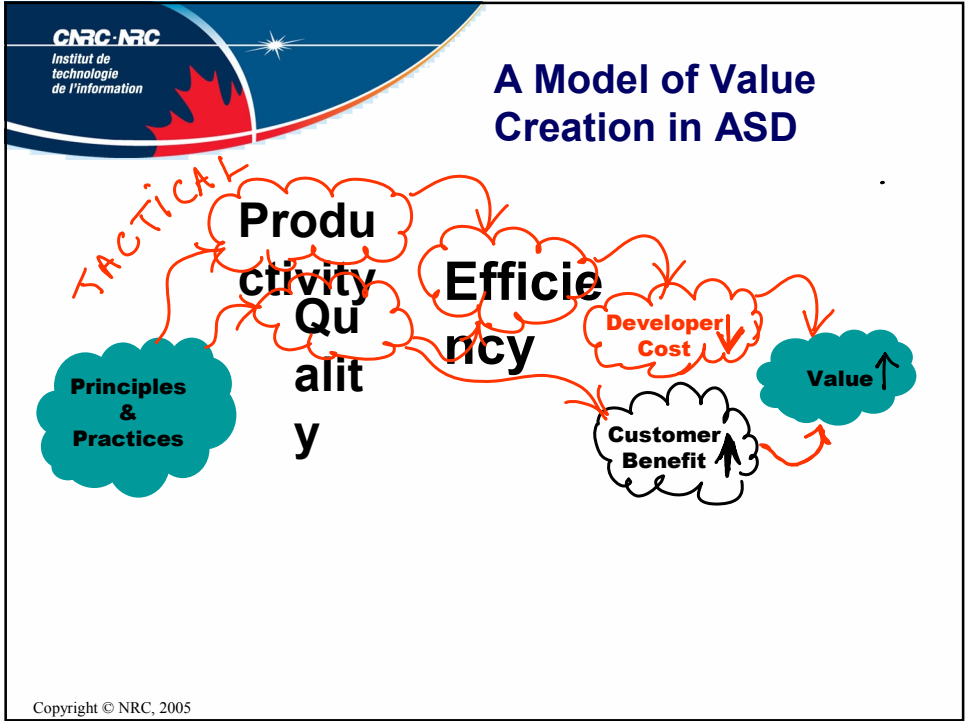
Copyright © NRC, 2005

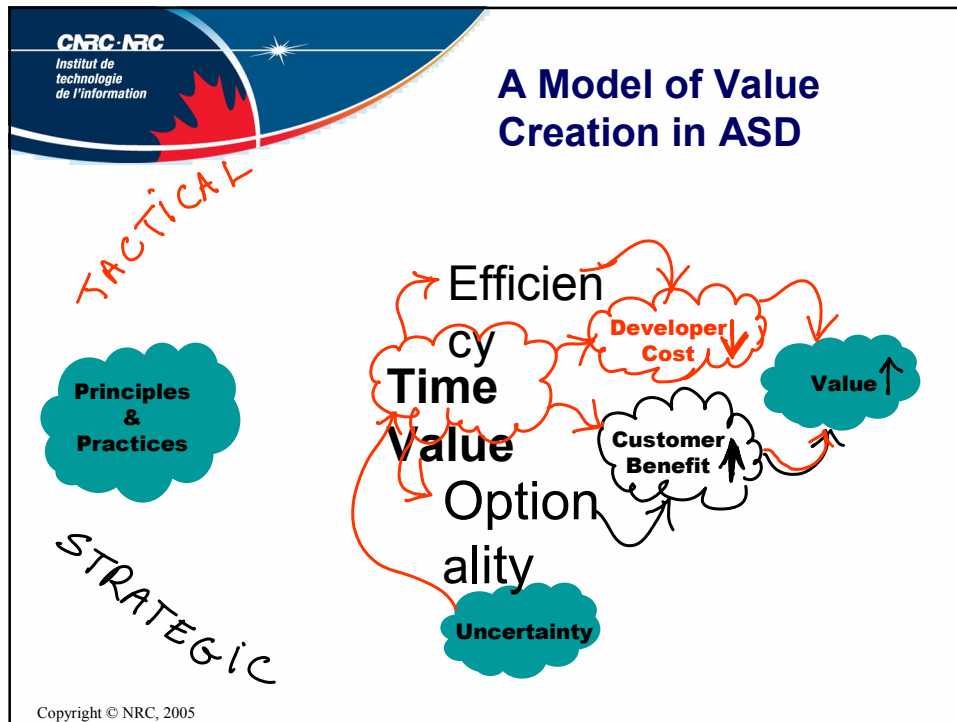
Relative Tactical Value of TDD

- **Testing is optional: TDD is mainly a *quality* technique**
 - Obvious
- **Testing mandatory: TDD is mainly a *productivity* technique**
 - Both short and long term
 - Better task focus
 - Better decomposition
 - Visible progress

Copyright © NRC, 2005







CMRC-NRC
Institut de technologie de l'information

Related Reading

- **B. Boehm & D. Turner. *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley, 2003**
- **H. Erdogmus. The Impact of Learning and Flexibility on the Economics of Iterative Development. To appear in *IEEE Software*, 2005**
- **H. Erdogmus & J. Favaro. Keep Your Options Option: Extreme Programming and Economics of Flexibility. In *Extreme Programming Perspectives*, Addison-Wesley, 2002**
- **H. Erdogmus and L. Williams. The Economics of Software Development by Pair Programmers. *The Engineering Economist*, 48(4), 2003**
- **H. Erdogmus, M. Morisio, M. Torchiano. On the Effectiveness of the Test-First Approach to Programming. *IEEE Transactions on Software Engineering*, 31(3), 2005**

Copyright © NRC, 2005