

Games Software Architects Play

Philippe Kruchten



Philippe Kruchten

Philippe Kruchten, Ph.D., P.Eng., CSDP



Professor of Software Engineering

NSERC Chair in Design Engineering



Department of Electrical and Computer Engineering

University of British Columbia

Vancouver, BC Canada

pbk@ece.ubc.ca



Founder and president

Kruchten Engineering Services Ltd

Vancouver, BC Canada

philippe@kruchten.com



Copyright © 2005-13 by Philippe Kruchten



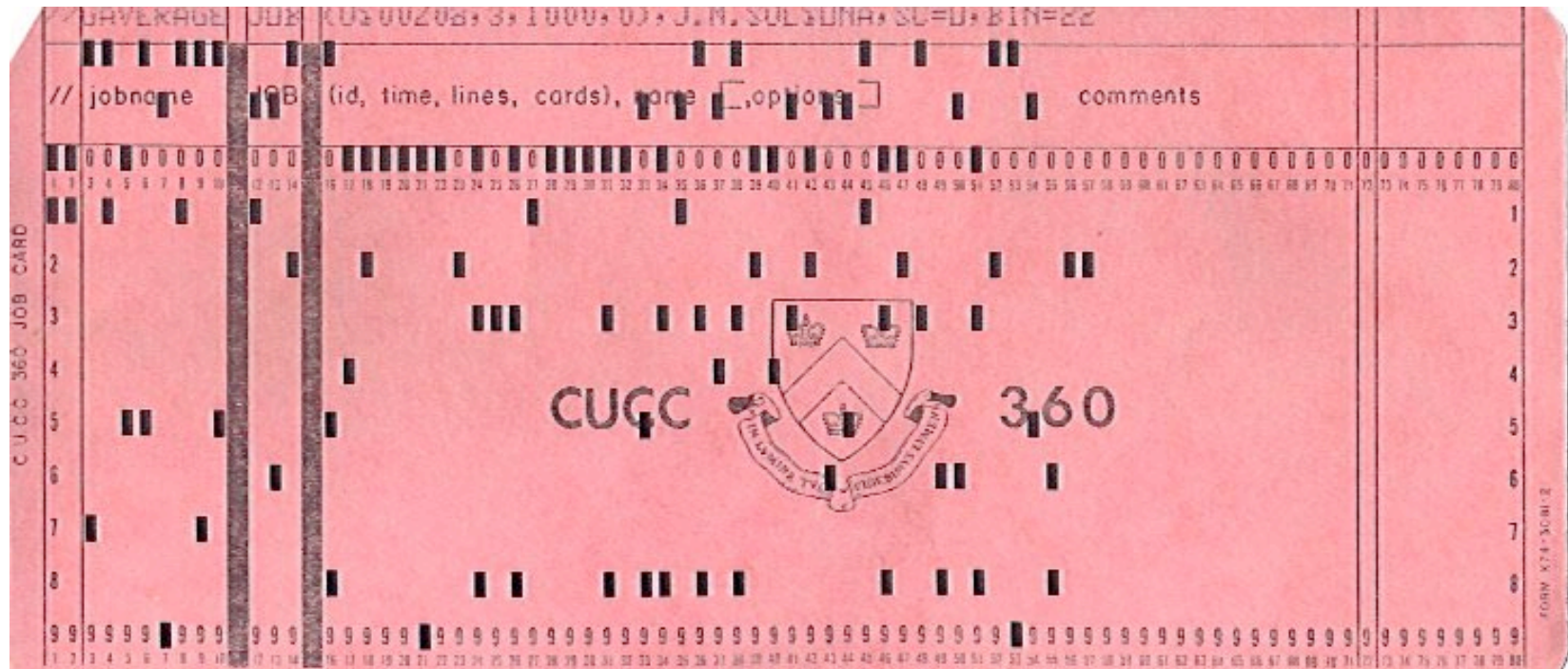
Games Software Architects Play

Philippe Kruchten

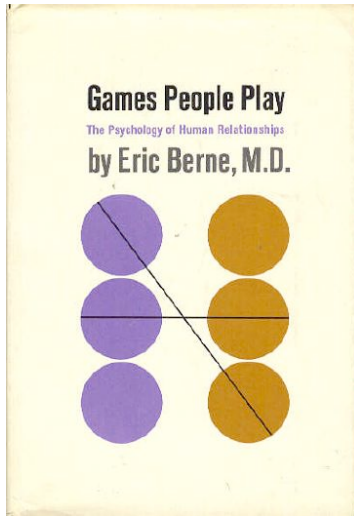
Brought to you by

**IEEE
Software**

1970

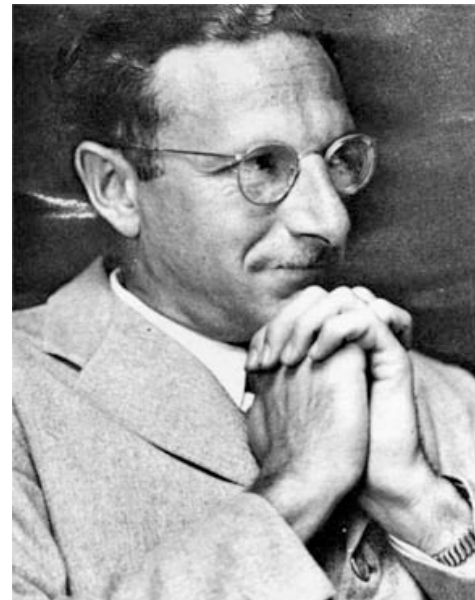


1970

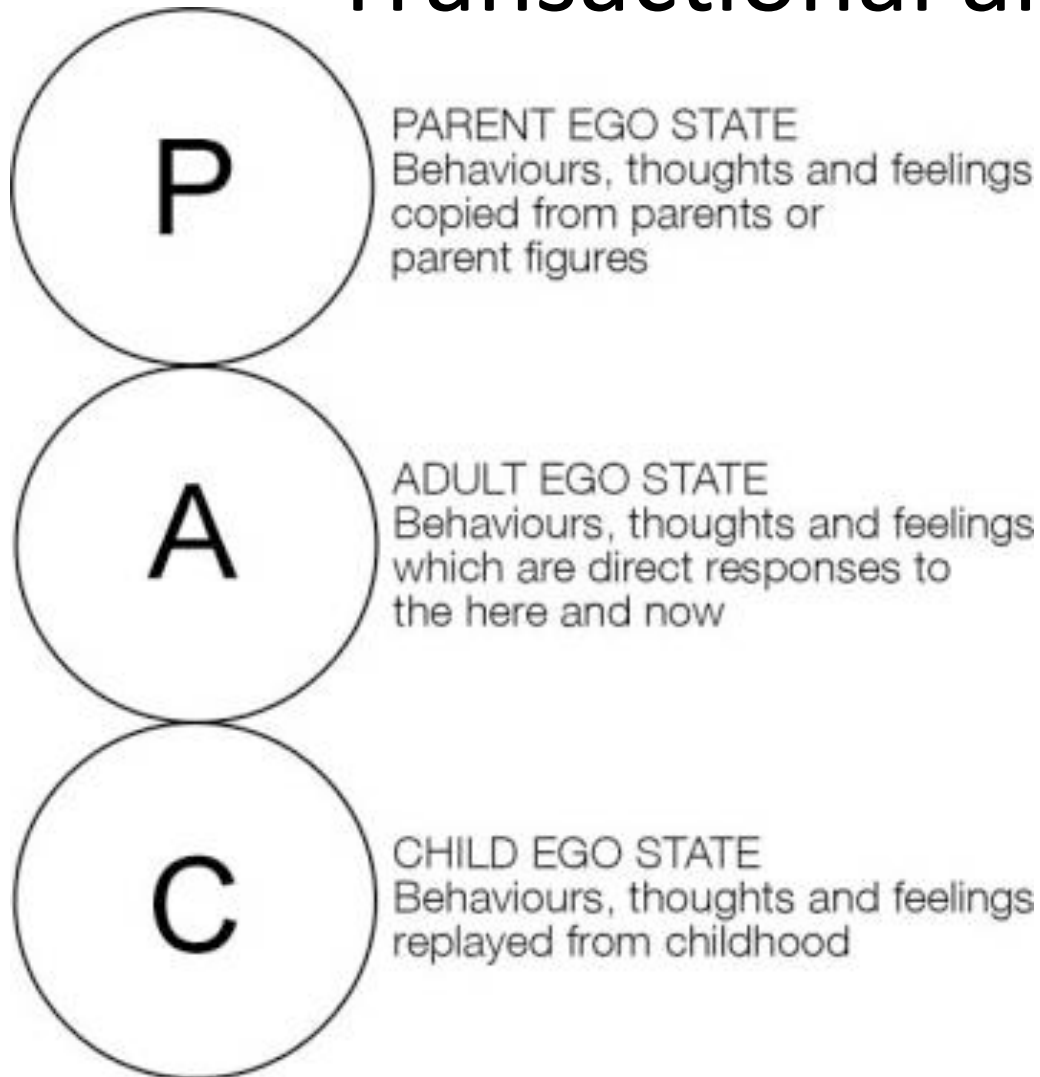


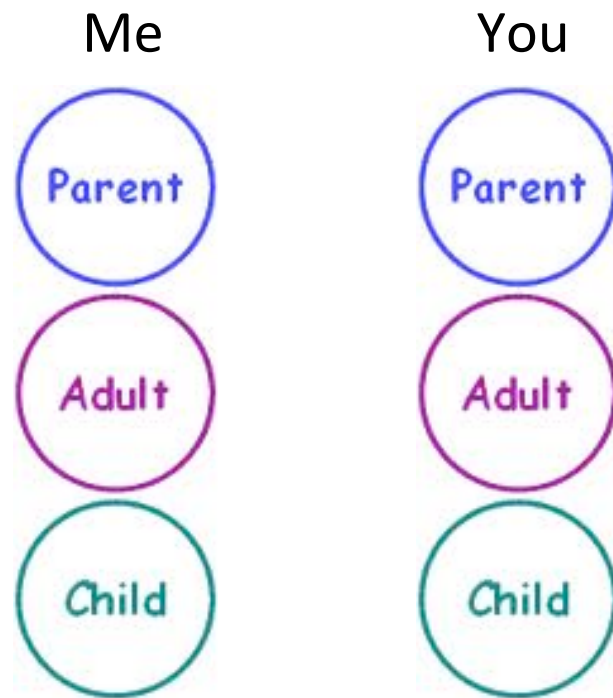
Games People Play

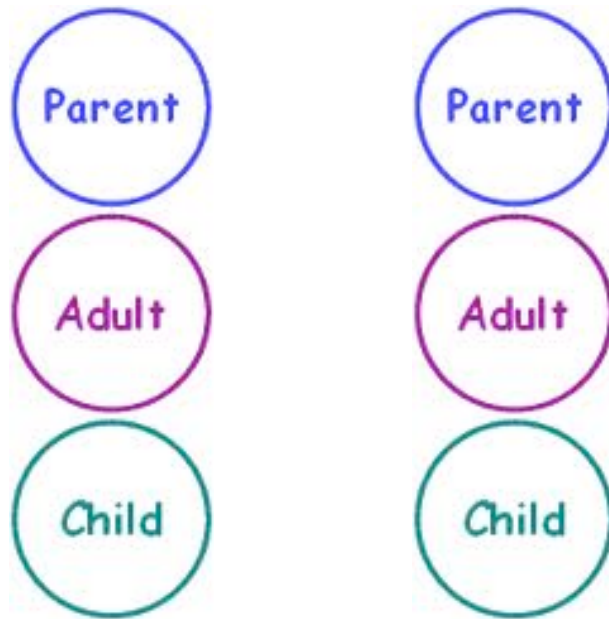
Eric Berne, 1964



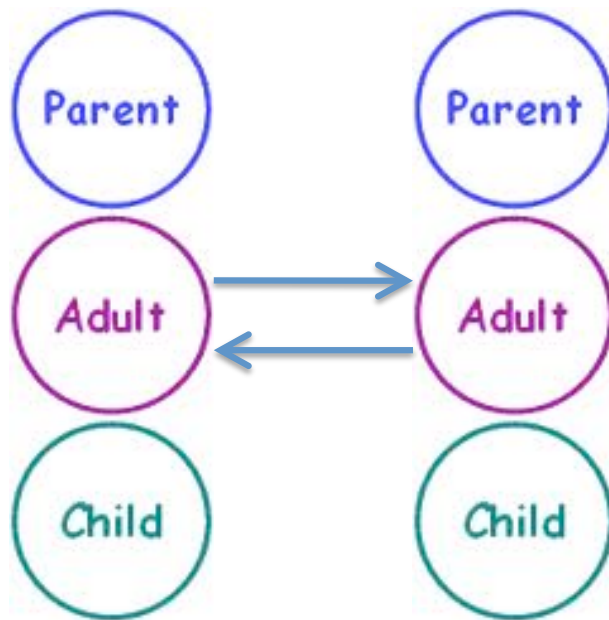
Transactional analysis





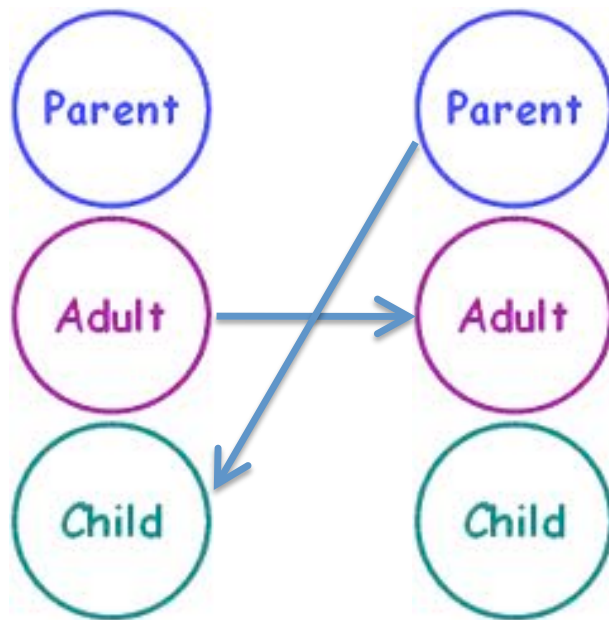


“**Games** are ritualistic transactions or behavior patterns between individuals that indicate hidden feelings or emotions...”



—What time is it?

—Three o'clock.



—Can you pass me the salt, please?

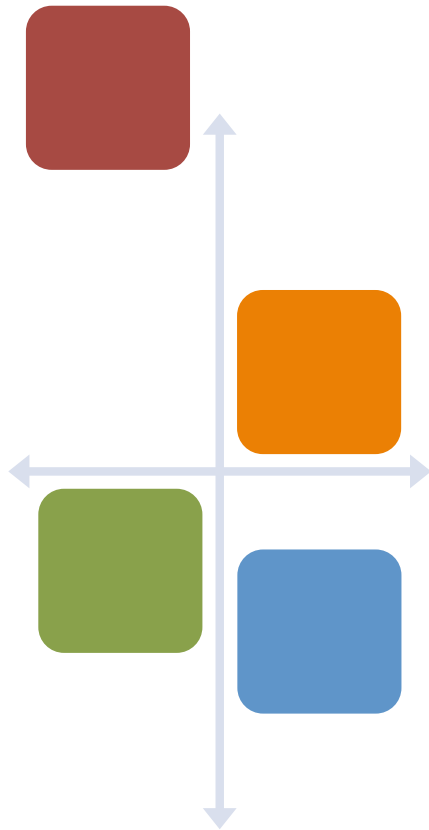
—Why did not you think about the salt before you sat down! Go and fetch it yourself now. This will teach you something about anticipating.



Career path

- 1970-1975: developed 2 large apps -> bought a car
(1975: *mechanical engineering degree*)
- 1983: very 1st Ada compiler (NYU)
- 1984: System architect at Alcatel
(1986: *PhD in information systems*)
- 1987: joined **Rational** as a consultant in SW Arch
- 1990: developed Rational architectural method
- 1992-1996: Lead SW architect for the Canadian ATC
- 1996-2003: Development of Rational Unified Process
- 2004-now: Professor at UBC

Outline



- Games people play
- Design & decision making
- Reasoning tactics for software design
- Cognitive biases, reasoning fallacies, and political games
- A gallery of simple games
- Nastier games
- Debunking and debiasing
- Design Rationale
- Critical thinking



Design = Decisions

- Designers use a **decision-making process**
- The *rationale*, when expressed, makes it partially visible
 - Arguments for...
- Tactics, heuristics

Architecting is making decisions

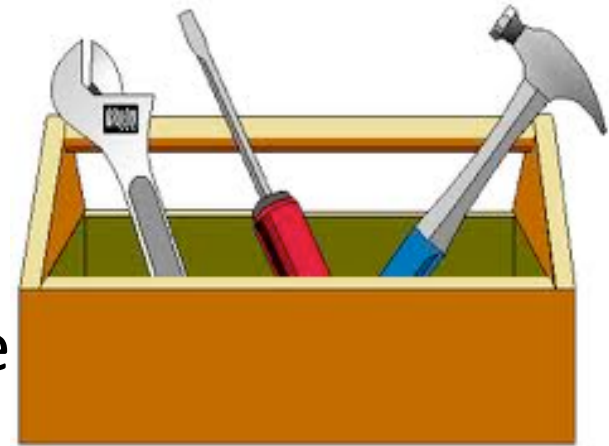
The life of a software architect is a long (and sometimes painful) succession of suboptimal decisions made partly in the dark. (me)

AK = AD + ADD

(Hans van Vliet, Patricia Lago, me)

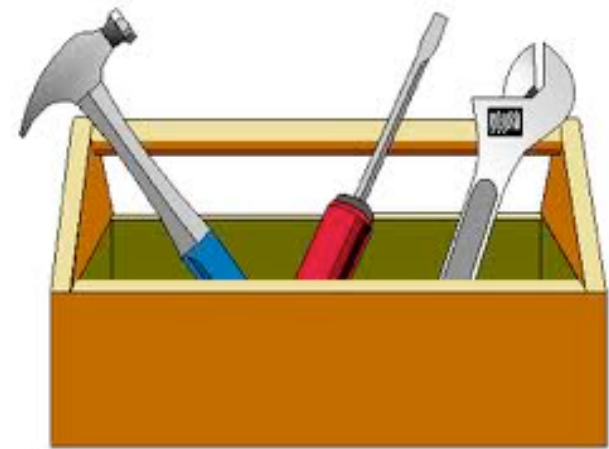
On the positive side:

- Divide and conquer
- Bring an outsider
- Reframe the problem
- Change level of abstraction
- Checklists, catalogues
- Remove constraints one by one
- Round-trip gestalt design
- Backtrack
- Spread 1/N



On the positive side (cont.)

- Time-bounded consensus seeking
- Write it down
- Formalize it (e.g., math)
- Explain it to a friend
- Sleep over it
- De Bono's 6 hats (?)
- ...



On the darker side



- Cognitive biases
- Reasoning fallacies
- Political games

On the darker side





Cognitive biases

- Software designers rely often on intuition
- But intuition may be flawed.
- Cognitive bias
 - = a pattern in deviation of judgment from accuracy or logic
 - => can lead to perceptual distortion, inaccurate judgment, or illogical interpretation

Cognitive biases



- Confirmatory bias
- Availability bias
- Representativeness bias
- ...

Cognitive biases



- Confirmatory bias
- Availability bias
- Representativeness bias
- ...
- *Anchoring bias*
- ...
- Bias bias



Reasoning Fallacies

- Flawed arguments, incorrect reasoning
=> Potentially wrong decision
- Argument /= belief or opinion
- Good argument
 - Relevant, accurate, fair premises
 - Logical
- Beliefs often presented as true facts.
- Reasoning fallacies are more likely to be accidental than deliberate, ... or not?

Political games

- A set of arguments, all superficially plausible, possibly leading to a design decision, but with a concealed ulterior motivation, maybe unrelated to the design.





A gallery of games software people play

Biases, fallacies and political games
found in real life software projects
over a period of 30 years



Golden hammer

- When you have a hammer, everything looks like a nail
- aka: we have found the “silver bullet”
- An architect has developed some deep expertise in some technique/tool/technology, and this becomes the first or sometime only possible solution to any new problem presented to her.
- See Anchoring

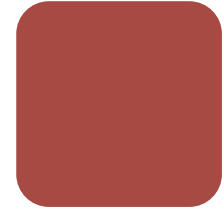
Elephant in the room



Elephant in the room

- All architects are fully aware of some major issue that really must be decided upon, but everyone keeps busy tackling small items, ignoring the big issue, pretending it does not exist, hoping maybe that it will vanish by magic or that someone else will take care of it.

Not invented here



- Avoid using or buying something because it comes from another culture or company, and redo it internally.
- Sometimes used jointly with Golden hammer, as a way to justify the hammer.
- Can be used as a political game
- aka: reinventing the wheel



Anchoring

- Relying heavily on one piece of information, to the detriment of other pieces of information, to justify some choice.
- Related to “blind spot” or “golden hammer”
- Sometimes re-inforced by confirmatory bias

Blink



- Gladwell's 2005 book: *Blink: the power of thinking without thinking*
- Aka. Fast and frugal (Gigerenzer)
- Extreme form of anchoring?

“Obviously...”

- When there is no relationship between the premises and the conclusion (or decision), and there is nothing obvious to any other stakeholder.
- Reasoning fallacy
- aka: Non sequitur
- See also:
 - post hoc ergo propter hoc: temporal succession implies a causal relation.
 - um hoc ergo propter hoc: correlation implies a causal relation

“Yes, but...”

- A delaying tactic which pushes back onto the requirements side to know more, get more details, sometimes on minute details or secondary use cases...
- Aka. Analysis paralysis (?)



Perfection or bust

- We need an optimal solution (the fastest, cheapest, nicest, etc. way to...
- Aka: searching for the silver bullet
- Decision avoidance strategy or delaying technique. Though one 'satisficing' solution maybe known, rather than provisionally using it and moving on, architects continue the search for something better, or optimal.
- Unboundedly rational decision maker

Cargo cult

- A group of people who imitate the superficial exterior of a process or system without having any understanding of the underlying substance.
- a flawed model of causation, when necessary condition are confused for sufficient conditions
- Straddling between cognitive bias and reasoning fallacy



It has worked before

- The conditions when it has worked before were significantly different, though.
- This is often following a “blink”, as a first line of defense.
- Related to “Hasty generalization”
- Representativeness bias with a sample of 1 ?

Not ripe and just good for scoundrels



- *French: “Ils sont trop verts et bon pour des goujats” from a fable by La Fontaine (1668), where the fox, unable to catch some grapes, decides that they are probably “not ripe and just good for scoundrels”*

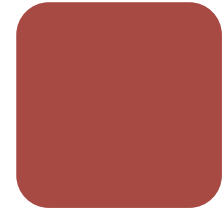
Not ripe and just good for scoundrels

- Architects try to use a certain solution/tool/technology/method, but by lack of time, resources, understanding, training or other, they fail to make it work, and then decide that it is intrinsically a bad solution/tool/method. (Ironically, 3 months later, they are leap-frogged by their competition, who have successfully used the said technology.)
- See also Pilot study

Swamped by evidence

- Repeating something in public often enough that in the end, it becomes familiar and will look more likely to be true (seen as true) in a subsequent argument.
- Political game element, with intent to induce a reasoning fallacy
- Aka. Argumentum verbosium (?)

“It’s a secret”



- Impose a solution withholding any evidence, claiming that there are some business reasons to do it that cannot be disclosed at this state.
- Political game
- Milder form: “it is too hard to explain now...” and “Trust me on this...”
- Contemptuous form: “You would not be able to understand”



Teacher's pet



- Aka: The boss will like it
- Though there is no technical evidence of it, a solution is adopted just because one stakeholder in position of power need to be appeased, pleased, and this can be used later to trade something else (maybe not at all related to this project)
- Political game
- May be played as an alternate to “it’s a secret”.

Groupthink

- Within a deeply cohesive in-group whose members try to minimize conflict and reach consensus without critically testing, analyzing and evaluating ideas.
- aka: bandwagon effect, herd behaviour, lemming behaviour
- Cognitive bias



Let us have a vote

- Sometimes thought of a technique to resolve a deadlock, it is often a political strategy by the responsible person to avoid taking a personal risk.
- Related to: “they made me do it...”
- Political game



Conscious versus Unconscious



Accidental versus Deliberate



Naïve versus Malevolent

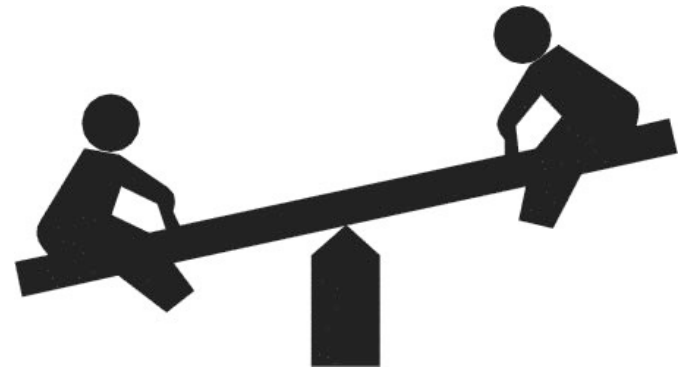


Assemble your own game

- Deliberate, maybe malevolent
 - Exploit one or more bias to your advantage
 - Active destruction of trust
-
- Example:
the Pilot Project



Note: Straddlers



- Anchoring-and-adjustment
 - Possible reasoning tactic
 - Bad only when proper adjustments do not occur
- Blink
 - Practiced by inexperienced people
 - May leave out many great possibilities
- Divide-and-Conquer
 - When no integration occurs, or late
- ...

Debiasing, debunking



- Increased awareness could help
- Contrarian in the team (the debunker)
- Reframing problems so that more information is visible
- Re-structuring arguments
 - Premise 1, premise 2, ... premise n, => Conclusion
- Challenge the premises
- See “positive tactics” (earlier)
- Back to the importance of **design rationale**



Critical thinking

Richard Paul & Linda Elder

www.criticalthinking.org

We think, we reason...



- for a *purpose*,
- within a *point of view*,
- based on some *assumptions*,
- leading to *implications* and consequences.
- We use *data*, facts, and experiences
- to make *inferences* and judgments
- based on *concepts* and theories
- to answer a *question* or solve a problem.

Intellectual standards



- Clarity
- Accuracy
- Precision
- Depth
- Breadth
- Relevance
- Logic-ness
- Significance
- Completeness
- Fairness



Research

- Evidence of cognitive biases in software engineering
- Experimentation on some specific biases
 - Lab, then practitioners
- Hypotheses on mitigation strategies
- Experimentations on mitigation strategies
- Interdisciplinary

THINKING, FAST AND SLOW



DANIEL
KAHNEMAN

WINNER OF THE NOBEL PRIZE IN ECONOMICS



May 2013

Copyright © 2005-13 by Philippe Kruchten

68

Dual process: System one, system two

- S1
 - Unconscious, implicit, experiential
 - fast
- S2
 - Conscious, explicit, rational, analytical
 - Slow
 - result of evolution of Homo Sapiens, 50,000y



WIKIPEDIA
The Free Encyclopedia

[Main page](#)

[Contents](#)

[Featured content](#)

[Current events](#)

[Random article](#)

[Donate to Wikipedia](#)

▼ [Interaction](#)

[Help](#)

[About Wikipedia](#)

[Community portal](#)

[Recent changes](#)

[Contact Wikipedia](#)

[Article](#)

[Talk](#)

Cognitive bias mitigation

From Wikipedia, the free encyclopedia

Cognitive bias mitigation refers to the prevention and reduction of the n influences on human decision making that reliably produce 'reasoning err

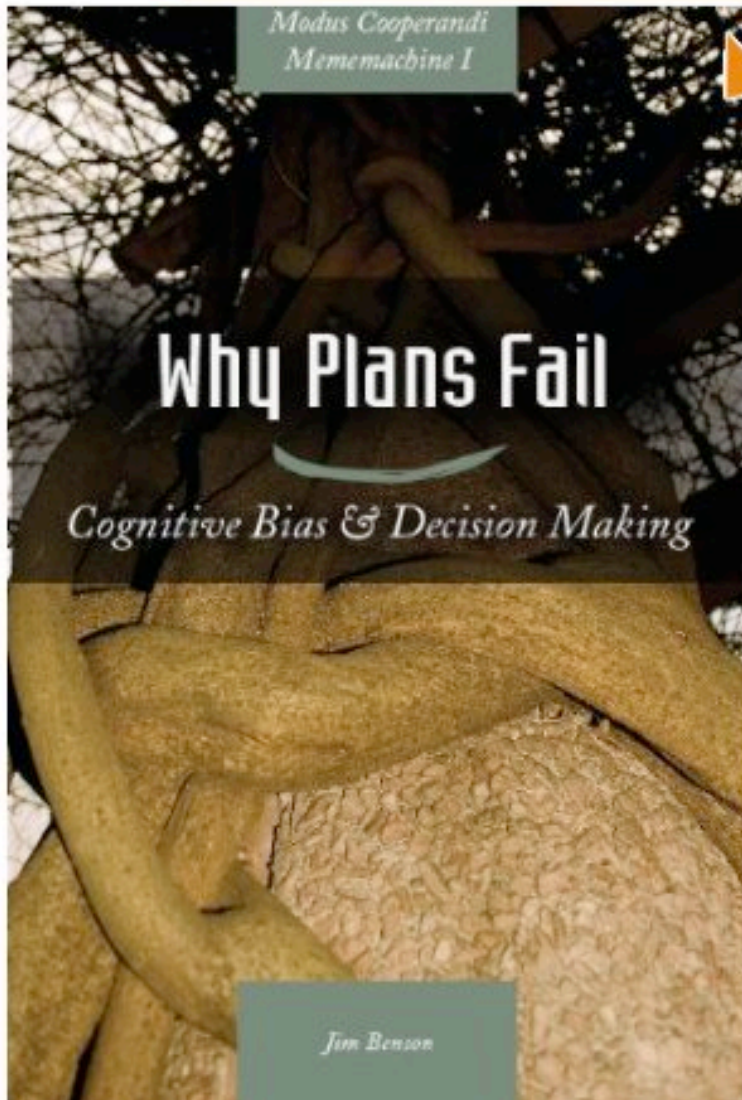
There is no coherent, comprehensive theory or practice of Cognitive Bias initiatives, in academic and professional disciplines concerned with the ef Bias Mitigation; most address Cognitive Bias Mitigation tacitly rather than

A long-standing debate regarding human decision making bears on the d debate contrasts the [rational economic agent](#) standard for decision makin debate also contrasts the methods used to analyze and predict human de versus heuristics emphasizing emotional states. This article identifies, wh

Contents [\[hide\]](#)

[1 Context](#)

LOOK INSIDE!



2011
Jim Benson
Modus Cooperandi



May 2013

Copyright © 2005-13 by Philippe Kruchten

71

- Developing a large, software-intensive system is made of thousands of **decisions**, from tiny ones to large, wide-ranging ones.
- Contrary to what we believe, we (humans beings) are not fully rational agents. Our decision-making process is marred by **cognitive biases** and **reasoning fallacies**. Plus some of us play nasty **political games**, exploiting these biases.
- Our first line of defense is awareness.



References (1)

- Berne, E. (1964). *Games People Play, the Psychology of Human Behavior*. New York: Grove Press.
- Benson, J. (2011), *Why Plans Fail: Cognitive Bias, Decision Making, and Your Business*, Seattle: Modus Cooperandi Press
- Calikli, G., Bener, A., & Arslan, B. (2010). *An analysis of the effects of company culture, education and experience on confirmation bias levels of software developers and testers*. Paper presented at the Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2.
- Damer, T. E. (2009). *Attacking Faulty Reasoning—A practical guide to fallacy-free arguments* (6th ed.). Belmont, CA: Wadsworth Cengage.
- Dörner, D. (1996). *The logic of failure*. Cambridge, MA: Perseus Pub.
- Epley, N., & Gilovich, T. (2006). The anchoring-and-adjustment heuristic--why the adjustments are insufficient. *Psychological Science*, 17(4), 311-.
- Gigerenzer, G., & Goldstein, D. G. (1996). Reasoning the Fast and Frugal Way: Models of Bounded Rationality. *Psychological Review*, 103(4), 650-669.
- Gladwell, M. (2005). *Blink: The Power of Thinking Without Thinking*. New York: Little, Brown and Company.

References (2)

- Hammond, J. S., Keeney, R. L., & Raiffa, H. (2006). The hidden traps in decision making. *Harvard Business Review*, 84(1), 118-126.
- Janis, I. (1983). *Groupthink: psychological studies of policy decisions and fiascos* (2 ed.): Houghton Mifflin.
- Kahneman, D. (2011). *Thinking, Fast and Slow*. New York: Farrar, Straus and Giroux.
- Levy, M., & Salvadori, M. G. (1992). *Why buildings fall down: how structures fail*. New York: W. W. Norton.
- Nutt, P. C. (2002). *Why decisions fail: Avoiding the blunders and traps that lead to debacles*. San Francisco: Berrett-Koehler Pub.
- Parsons, J., & Saunders, C. (2004). Cognitive Heuristics in Software Engineering: Applying and Extending Anchoring and Adjustment to Artifact Reuse. *IEEE Trans. on Software Engineering*, 30, 873-888.
- Paul, R. W., & Elder, L. (1999). *The Miniature Guide to Critical Thinking : Concepts and Tools*. USA: The Foundation for Critical Thinking.
- Robbins, J. E., Hilbert, D. M., & Redmiles, D. F. (1998). *Software architecture critics in Argo*. Paper presented at the Proceedings of the 3rd international conference on Intelligent user interfaces.
- Schön, D. A. (1983). *The reflective practitioner: How professionals think in action*. New York: Basic Books.





References (3)

- Siau, K., Wand, Y., & Benbasat, I. (1996). When parents need not have children—Cognitive biases in information modeling. In P. Constantopoulos, J. Mylopoulos & Y. Vassiliou (Eds.), *Advanced Information Systems Engineering* (Vol. 1080, pp. 402-420) Berlin: Springer.
- Simon, H. A. (1991). Bounded rationality and organizational learning. *Organization Science*, 2(1), 125-134.
- Stacy, W., & MacMillan, J. (1995). Cognitive bias in software engineering. *Communications of the ACM*, 38(6), 57-63.
- Tang, A. (2011). *Software Designers, Are You Biased?* Paper to be presented at the *SHARK workshop* at ICSE 2011.
- Wendorff, P., & Apšvalka, D. (2005, April 14-15). *Human Knowledge Management and Decision Making in Software Development Method Selection*. Paper presented at the 12th Annual Workshop of the German Informatics Society Special Interest Group WI-VM, Berlin.
- Williams, T., Samset, K., & Sunnevåg, K. (2009). *Making essential choices with scant information: front-end decision-making in major projects*. Basingstoke, UK: Palgrave Macmillan.

Slides at philippe.kruchten.com/Talks



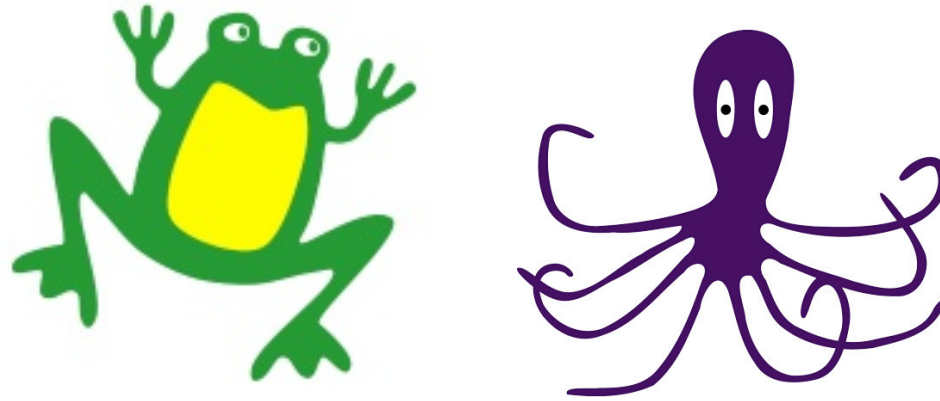


May 2013



Copyright © 2005-13 by Philippe Kruchten

78



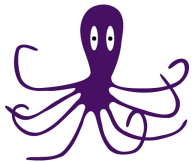
The Frog and the Octopus

A Conceptual Model of
Software Development

As of August 2012



*Once upon a time, a frog and an octopus,
Met on a software project, that was deep in the bush.
The frog said, “you know, all these projects are the same;
Over time we fill with our work the gap that we find
Between the burgeoning product, and our dreamed intent.”*



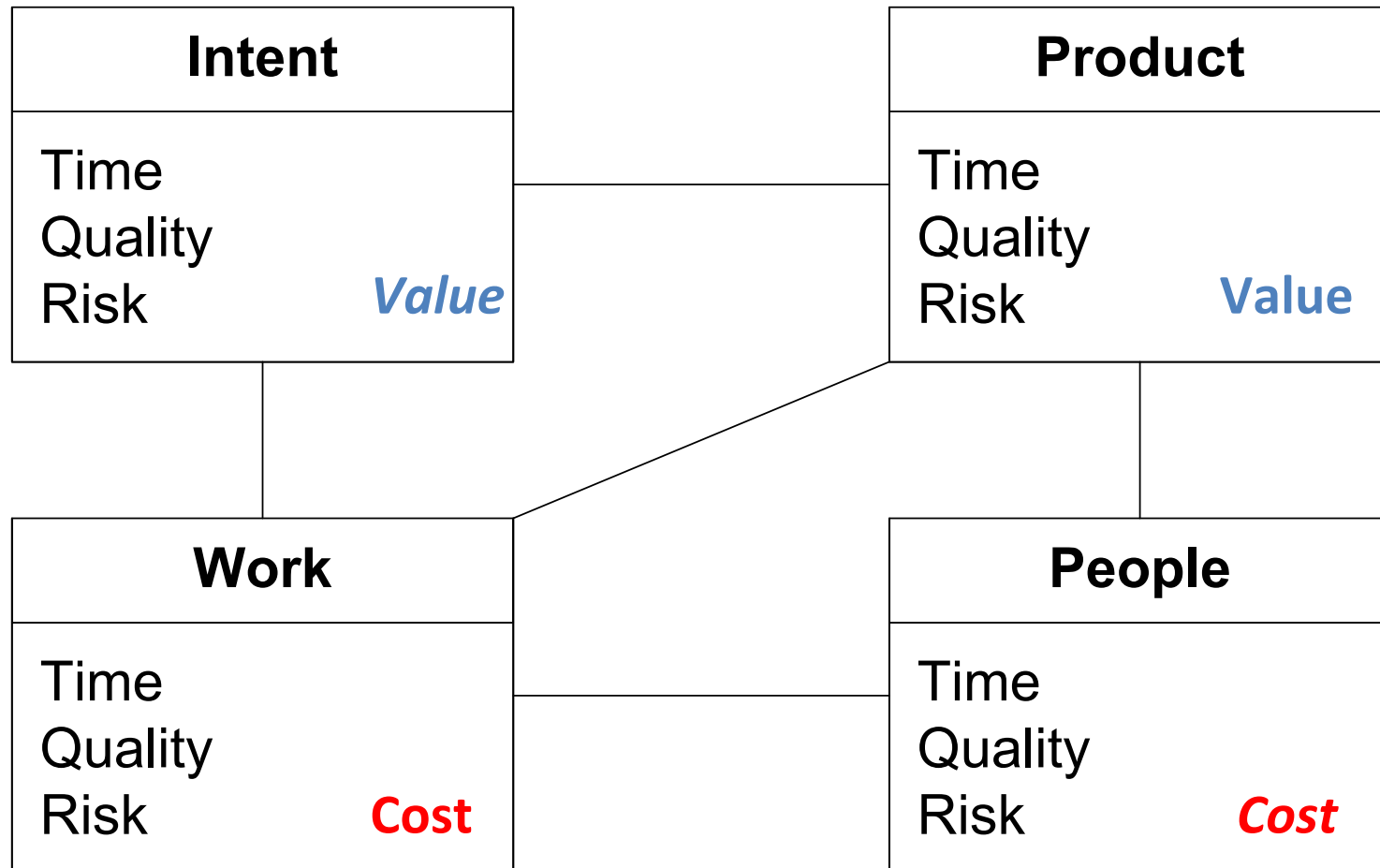
*“Oh, no” objected the octopus, “they cannot be the same;
They come in all forms or shapes and sizes and colours,
And we cannot use the same tools and techniques
Like in the cobbler shop, one size does not fit all.”*

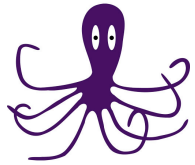
“The purpose of science is not to analyze or describe but to make useful models of the world. A model is useful if it allows us to get use out of it.”

Edward de Bono

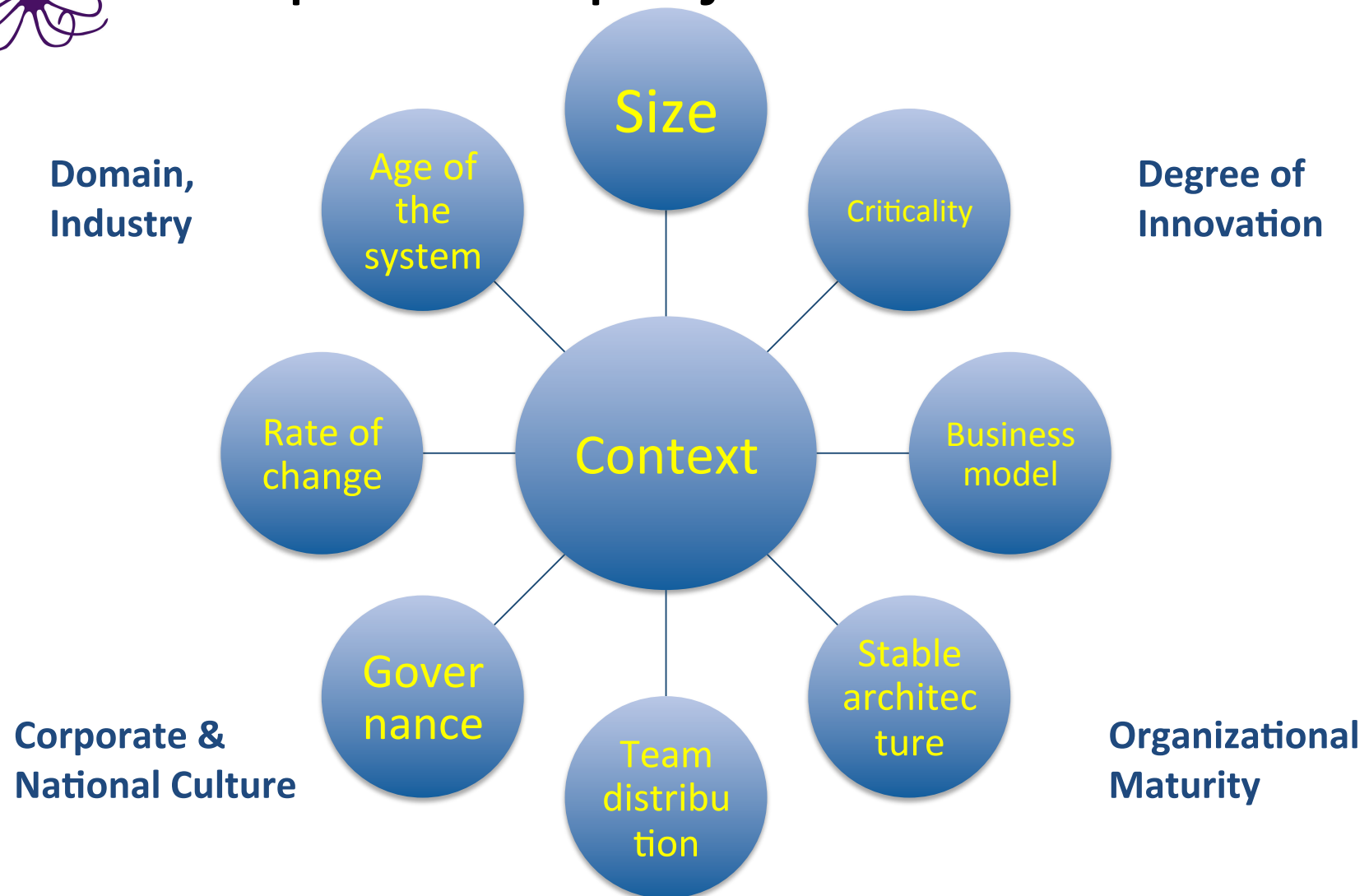


Frog: “All projects are the same”



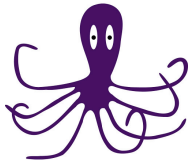


Octopus: “All projects are different!”

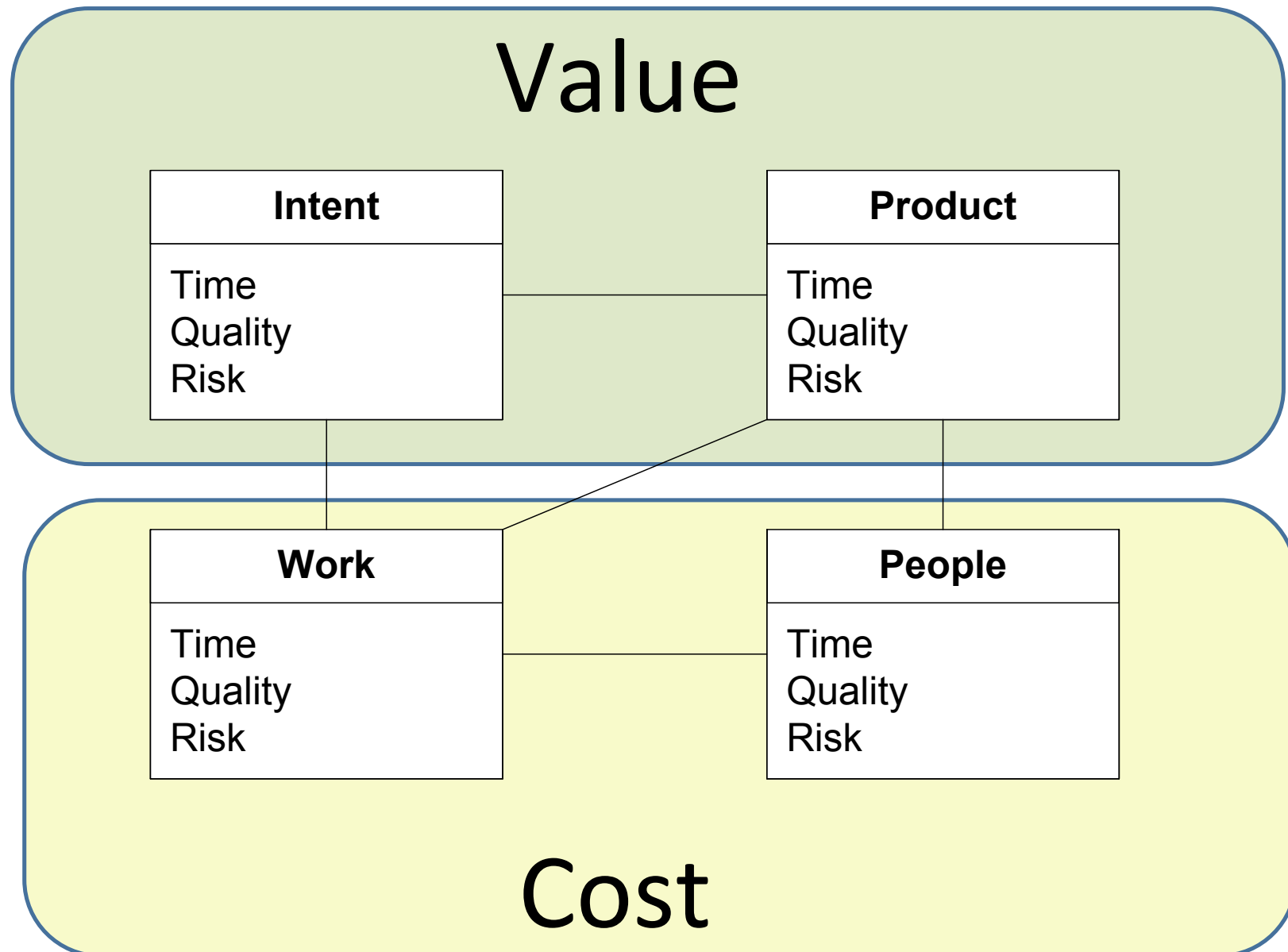




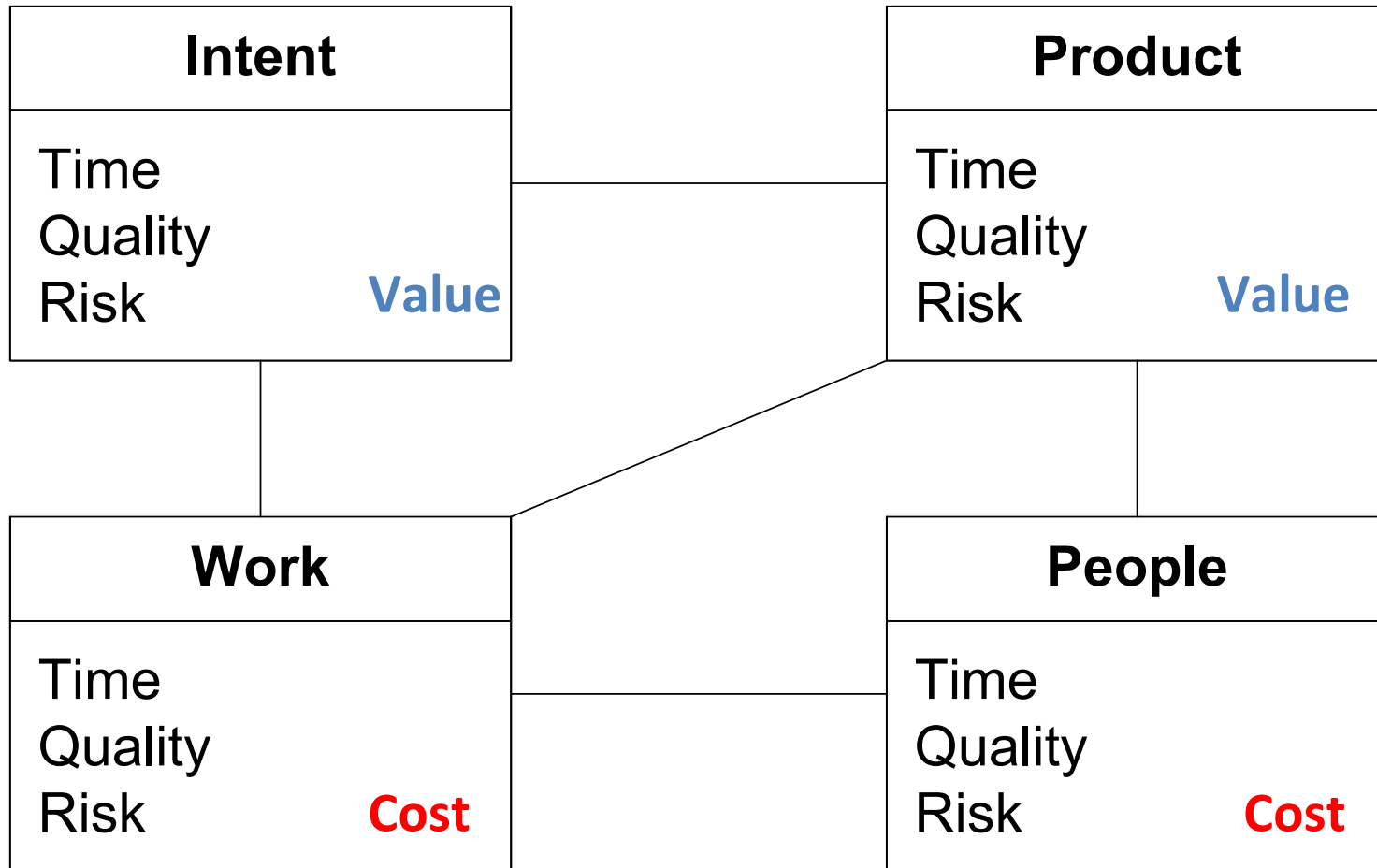
A **project** is all the **work** that **people** have to accomplish over **time** to realize in a **product** some specific **intent**, at some level of **quality**, delivering **value** to the business at a given **cost**, while resolving many **uncertainties and risk**.



All aspects of software projects are affected by **context**: size, criticality, team distribution, pre-existence of an architecture, governance, business model, which will guide with practices will actually perform best, within a certain domain and culture.



Adding Value and Cost



“... a model is useful if it allows us to get use out of it.”

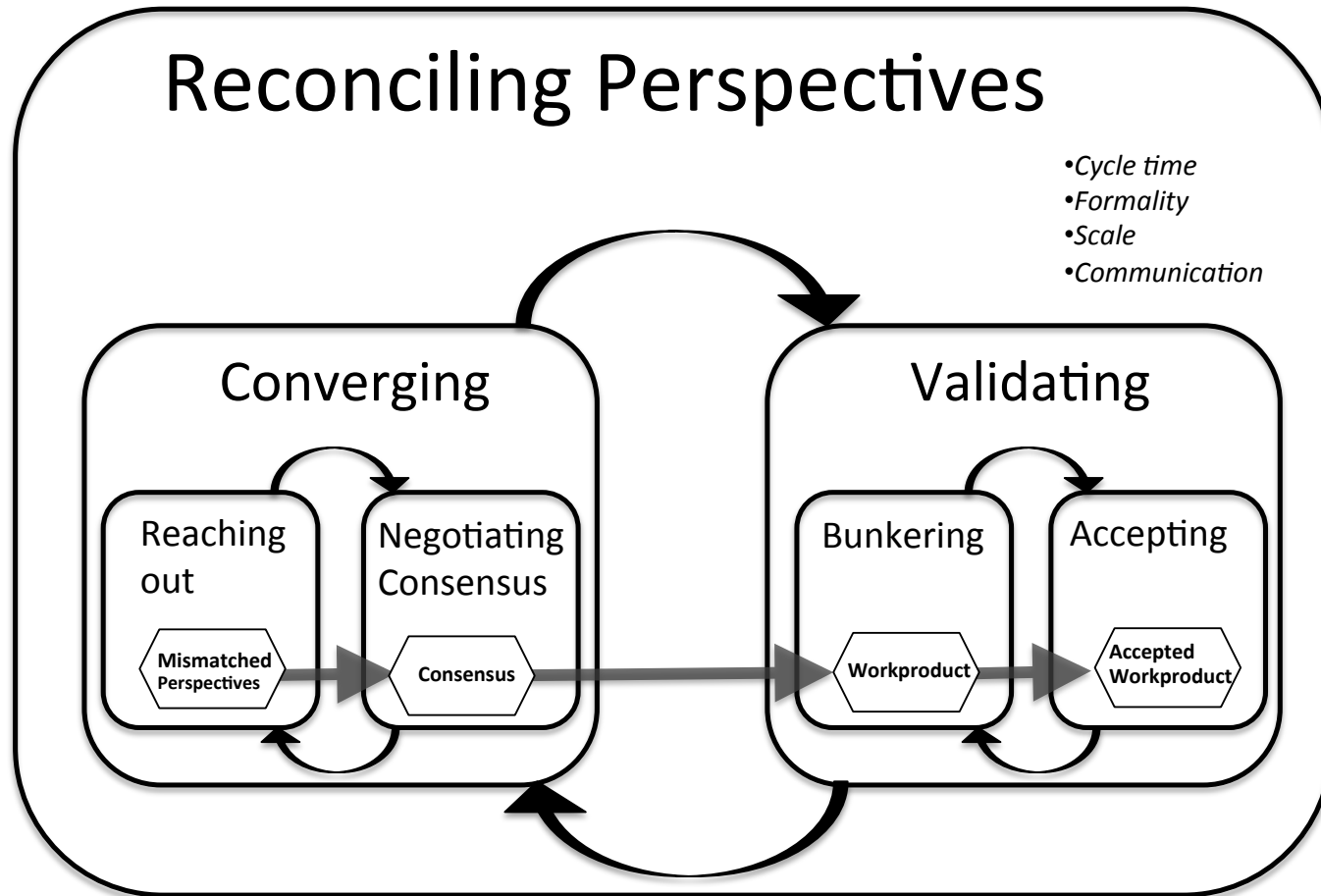
Edward de Bono



From Steve Adolph

Reconciling Perspectives

- Cycle time
- Formality
- Scale
- Communication



Legend:

Process

Artifact

Precedes

Transfer

• Property