Business Value of Options in SW Products, Projects & Processes

Kevin Sullivan

University of Virginia

Department of Computer Science

A vision: to connect technical decisions to economics in a way appropriate to the high levels of uncertainty & dynamism in our field, to improve life for developers, managers, and society more broadly

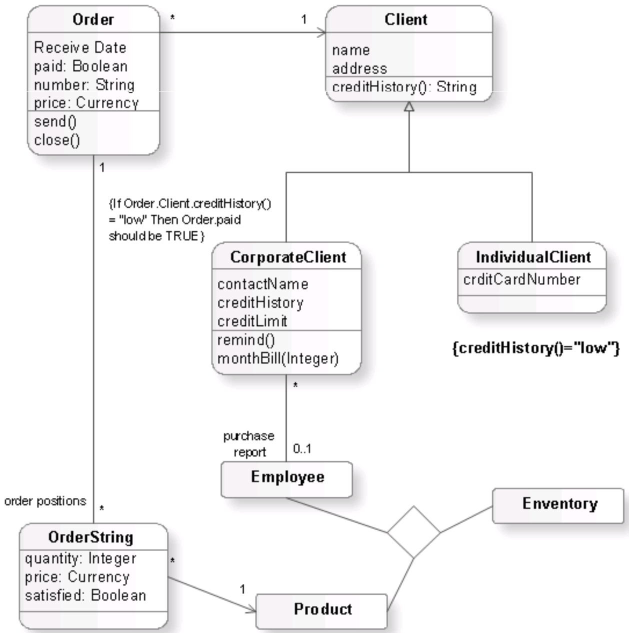
Dealing with uncertainty demands flexibility and dynamic management; these issues can be understood in terms of *options and optimal exercise strategy;* if we look around we find these issues cropping up everywhere; and we might benefit by treating these issues explicitly

Technical decisions in software development often have significant *economic* implications

Technical decision making is too disconnected from economic modeling and analysis

Architecture ⟷ Value of Firm

Languages and conceptual frameworks of developers and managers not well connected



Net Present Value (NPV)

$$NPV = \sum_{t=1}^{T} \frac{\text{Cash Flow}_t}{(1+i)^t} - \text{Initial Cash Investment}$$

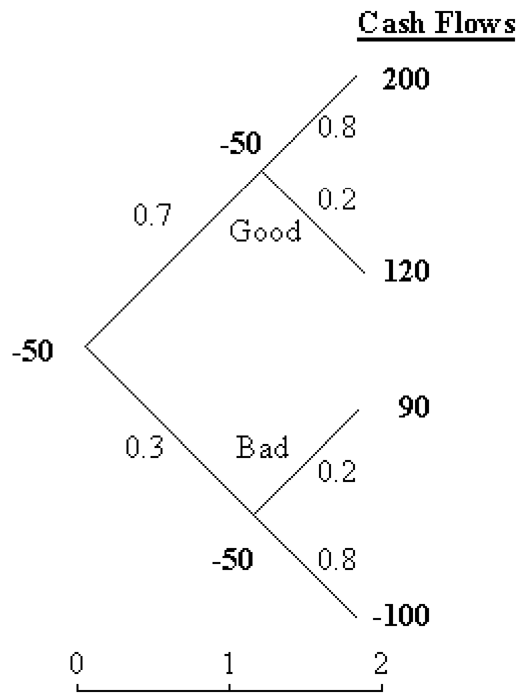
t = Cash Flow Period
i = Interest Rate Assumption

Appropriate financial theory is a bit complex

Net Present Value (NPV)

$$NPV = \sum_{t=1}^{T} \frac{\text{Cash Flow}_t}{(1+i)^t} - \text{Initial Cash Investment}$$

t = Cash Flow Period
i = Interest Rate Assumption



Lack modeling techniques and tools that make economic modeling/analysis useful in practice

Developers lack tools to *explain* how key technical decisions affect wealth of the firm

Management often lacks *understanding* of how software development can create value

Need to better align software development decisions with economic objectives of firm

This is an important problem

- for management:
  - improve returns on investments in SW/IT
  - exploit strategic value of SW/IT
- for development
  - improve basis for technical decision making
  - better defend technical decisions to management
  - strengthen business case for investments in SW/IT
  - Contribute more effectively to health of firm

This is a hard problem

– management often lacks technical knowledge and developers often lack training in finance

– required financial reasoning is somewhat exotic

– lack methods and tools to make ideas accessible

Progress is possible but it requires

- research to develop & validate new theories connecting technical realm to financial realm, tailored to unique nature of software
- Modeling and analysis approaches and tools to deliver theory in a practical and useful form

Today's talk

- explain why nature of software development demands use of advanced ideas from finance: *options value* and *dynamic investment management*
- value and exploitation of flexibility under uncertainty
- link options thinking to key issues in SW development
- leave you believing these ideas are worth developing

Example

- Should we invest to restructure a system?
  - Reduce costs of meeting future demands
  - But future demands are often uncertain
  - Payoff is therefore often uncertain
- Technical decision has financial consequences

Example

- Suppose restructuring costs $1,600 (K)

- How do we decide?

- MBA Finance 101 is not good enough

Answer Given by Finance 101: Net Present Value Rule

- Invest if the net present value (NPV) of the investment is positive, otherwise decline
- Compare discounted future cash flows to up-front investment and invest if there’s a surplus
- Simple if future is certain; little more complex if future cash flows is a random variable

Net Present Value (NPV)

$$NPV = \sum_{t=1}^{T} \frac{\text{Cash Flow}_t}{(1+i)^t} - \text{Initial Cash Investment}$$

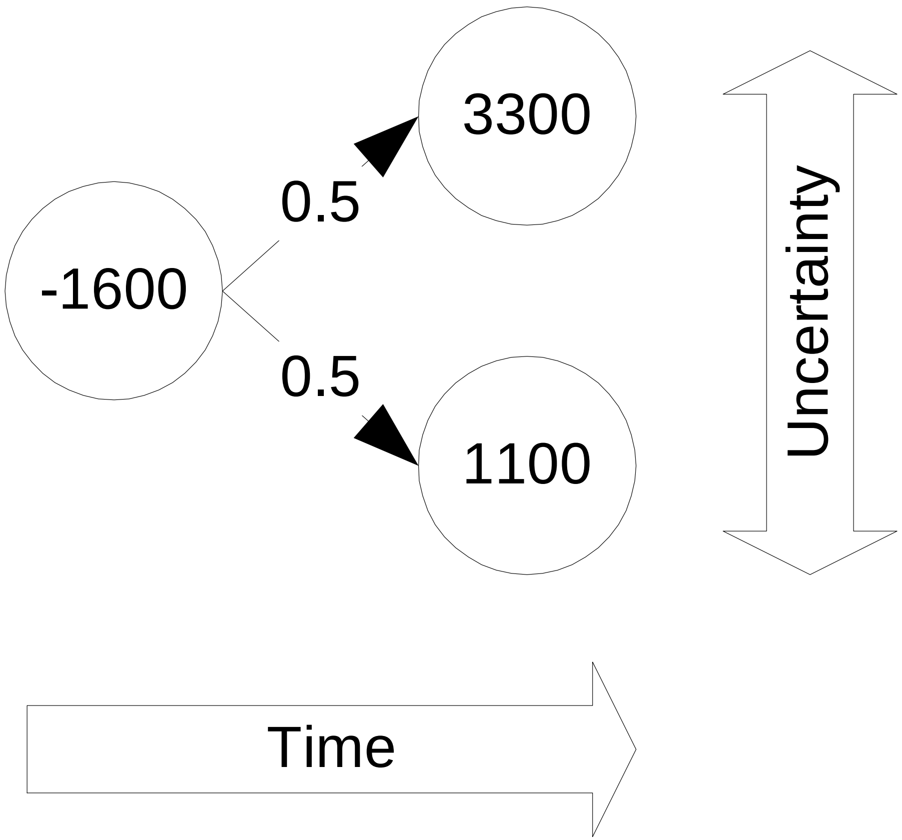
t = Cash Flow Period
i = Interest Rate Assumption

NPV: Discounting for Risk

- Benefits are often uncertain
  - E.g., if customer wins contract, demand high
  - high future demand favorable ($3,300)
  - low future demand unfavorable ($1,100)
  - let’s assume a 50/50 chance of either outcome

- Use event trees to model uncertainty; and use a probability weighted version of NPV rule

Static NPV Under Uncertainty

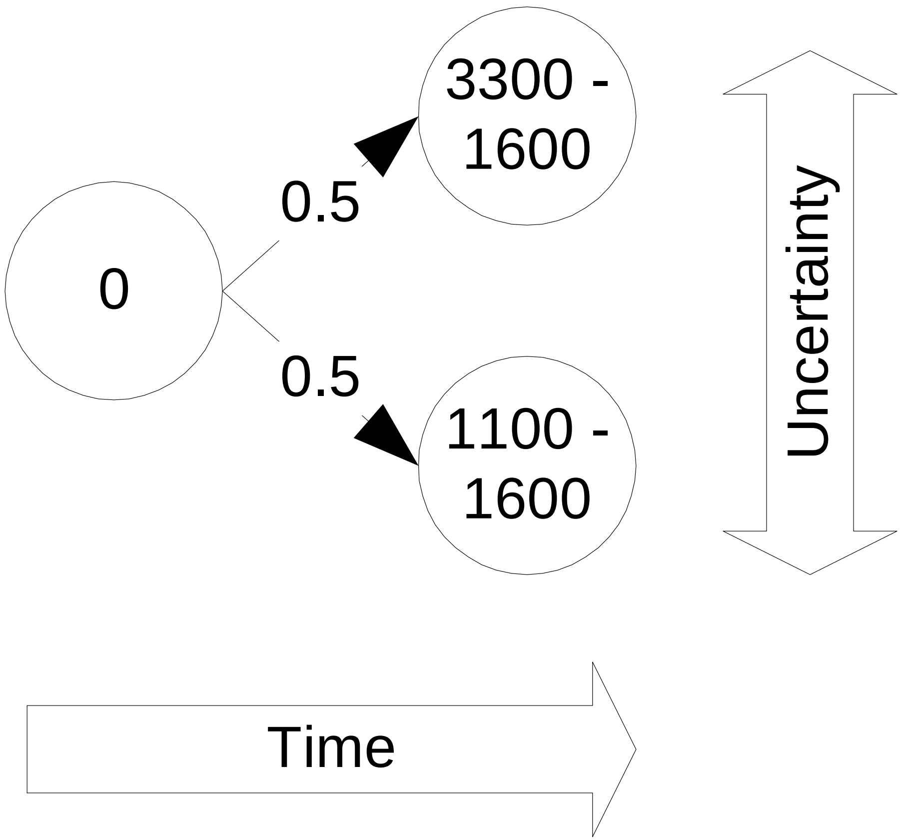


- Discount rate = 10%
- PV(3300) = 3000
- PV(1100) = 1000
- PV = 0.5*3000 + 0.5*1000
- NPV = 2000 - 1600 = 400
- Decide to invest?

Is NPV Rule the Best Strategy?

- What if you have flexibility to wait to see how uncertainty is resolved?
- Delaying preserves *valuable option* to invest until after you know how the future turns out!
- Leads to dynamic investment decision strategy

Dynamic NPV



- PV (1600) (t1) = 1454
- Favorable payoff
  max{0, 3000-1454} = 1546
- Unfavorable payoff
  max{0, 1000-1454} = 0
- DNPV = 0.5*(1546+ 0) = 773
- Static NPV model is wrong!

Options Value

- What is the value of the *option* to invest in restructuring?

- Option value is payoff on optimal exercise strategy
  - Exercising at t=0 is not optimal; but deciding at t=1 is
  - There are no other strategies in this case, therefore …
  - At t=0, the value of the *<u>option to restructure</u>* is $773

- No sense in killing a $773 option at t=0 for an expected payoff of only $400 (the static NPV)

Options are real assets in software development

- decision rights without corresponding obligations
- often implicit in product and project structures
- provide flexibility to adapt as uncertainties resolve
- today we don't model and analyze them explicitly
- not clear we're creating & exploiting them effectively

Basic Options Concepts

- Underlying random process (S_k)
- Non-linear payoff: max(0, S_k-L)
- Optimal strategy: exercise only when payoff compensates for both L and value of option
- Options have value that
  - often exceeds immediate payoff
  - depends on S_k, L, *variance (uncertainty)*, time, …

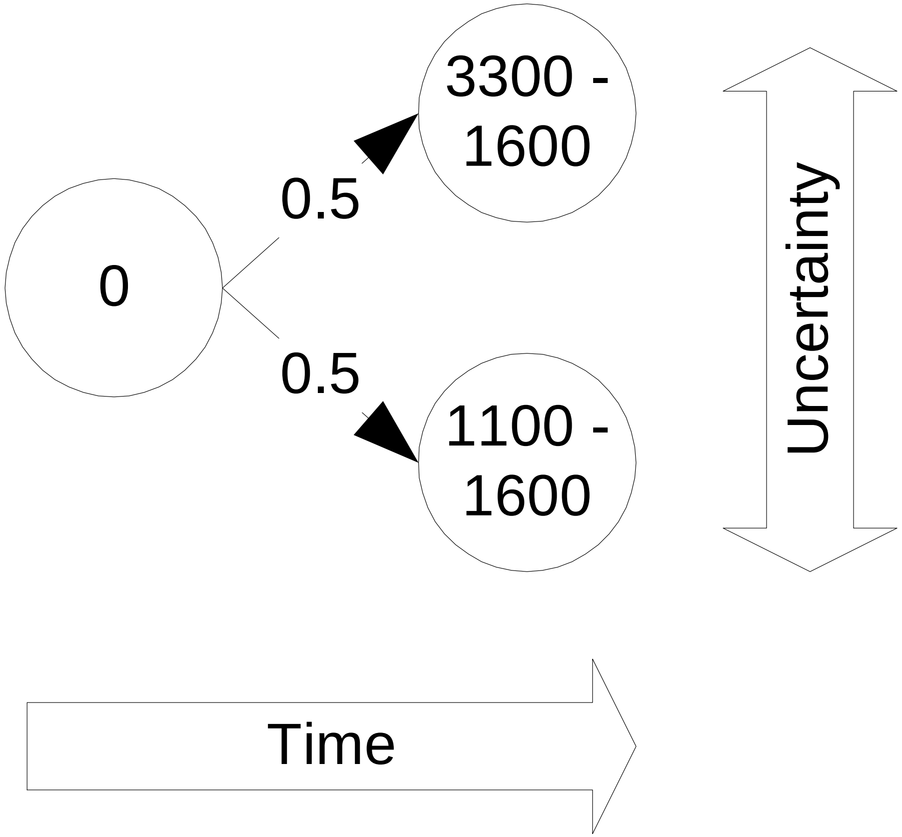
Options values increase with risk!

– uncertainty is endemic to our field

– options can thus have great value

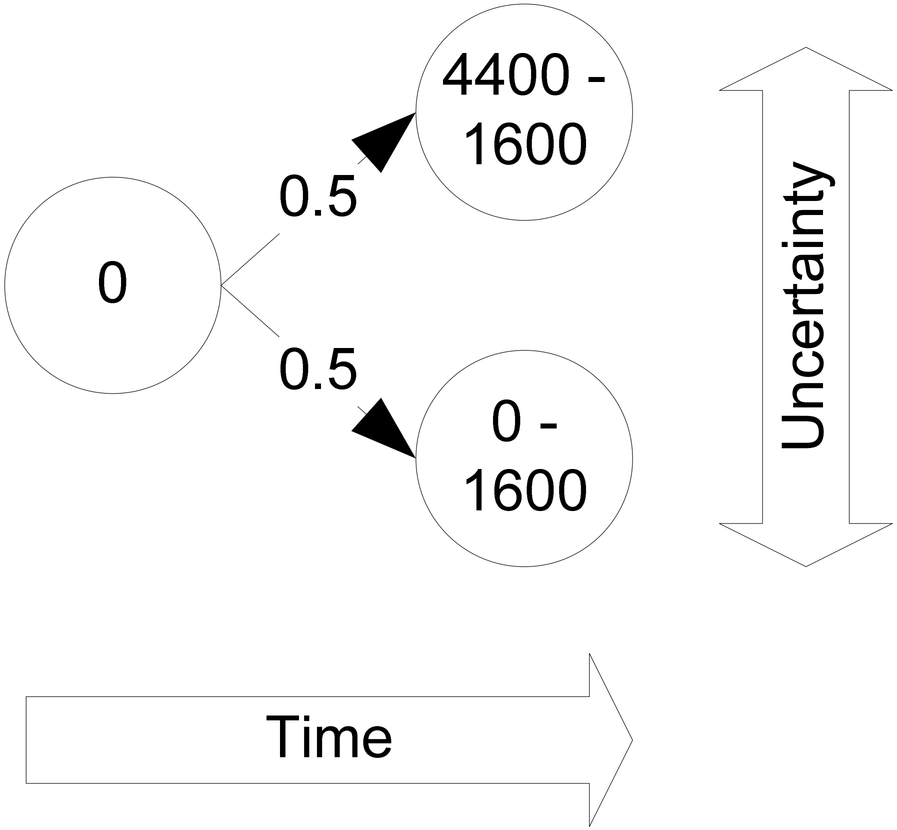
– need to understand options to manage uncertainty

Dynamic NPV



- PV (1600) (t1) = 1454
- Favorable payoff
  max{0, 3000-1454} = 1546
- Unfavorable payoff
  max{0, 1000-1454} = 0
- DNPV = 0.5*(1546+ 0) = 773
- Static NPV model is wrong!

Options Values Increase w/ Risk



- PV (1600) (t1) = 1454
- Favorable payoff
  max{0, 4000-1454} = 2546
- Unfavorable payoff
  max{0, 0-1454} = 0
- DNPV = 0.5*(2546+ 0) = **1273**

Real options

- Options to defer investing
- Option to default early on a project
- Option to expand or contract production
- Option to substitute one material for another
- Options on options – e.g., phased projects
- Option to select from set of risky assets

Common practices implicitly options-oriented

- stopping problems: e.g., when to ship, or commit to build (e.g., in agile, do not invest until requirements are certain)

- phased project structures, e.g., as in spiral model
  - explicitly (albeit informally) models risk
  - exploration to create options to select
  - prototyping, incremental development to resolve risks
  - gating of project phases to create options e.g., to abandon

- modular architectures create substitution options

Optimal Stopping

- Is *delay for as long as possible* always best policy?
- Theorem: In the absence of dividends, waiting until expiration is optimal for an American call
- However, dividends create early exercise incentive
- Need to understand whether waiting has a cost

Phased Projects (e.g., Spiral Model)

- *From risk-minimization to value maximization*
- Must we decide based on whole project NPV?
- Enhancement in two phases, first costs 1000
- Equal odds second costs 0 or 3000 (PV=$1500)
- Profit: 200 per period, 10% disc. rate: 2200
- NPV is -300 – is right decision not to invest?

- Phase one value is 0.5(2200) - 1000 = 100

Modularity

- Starting whole system over
  - Throws away good with bad
  - Gets bad with good in new system

- Modularity create options to search for and use better *parts*

- Option on portfolio vs. portfolio of options

- Uncertainty in results of R&D investments in search

Baldwin & Clark Model

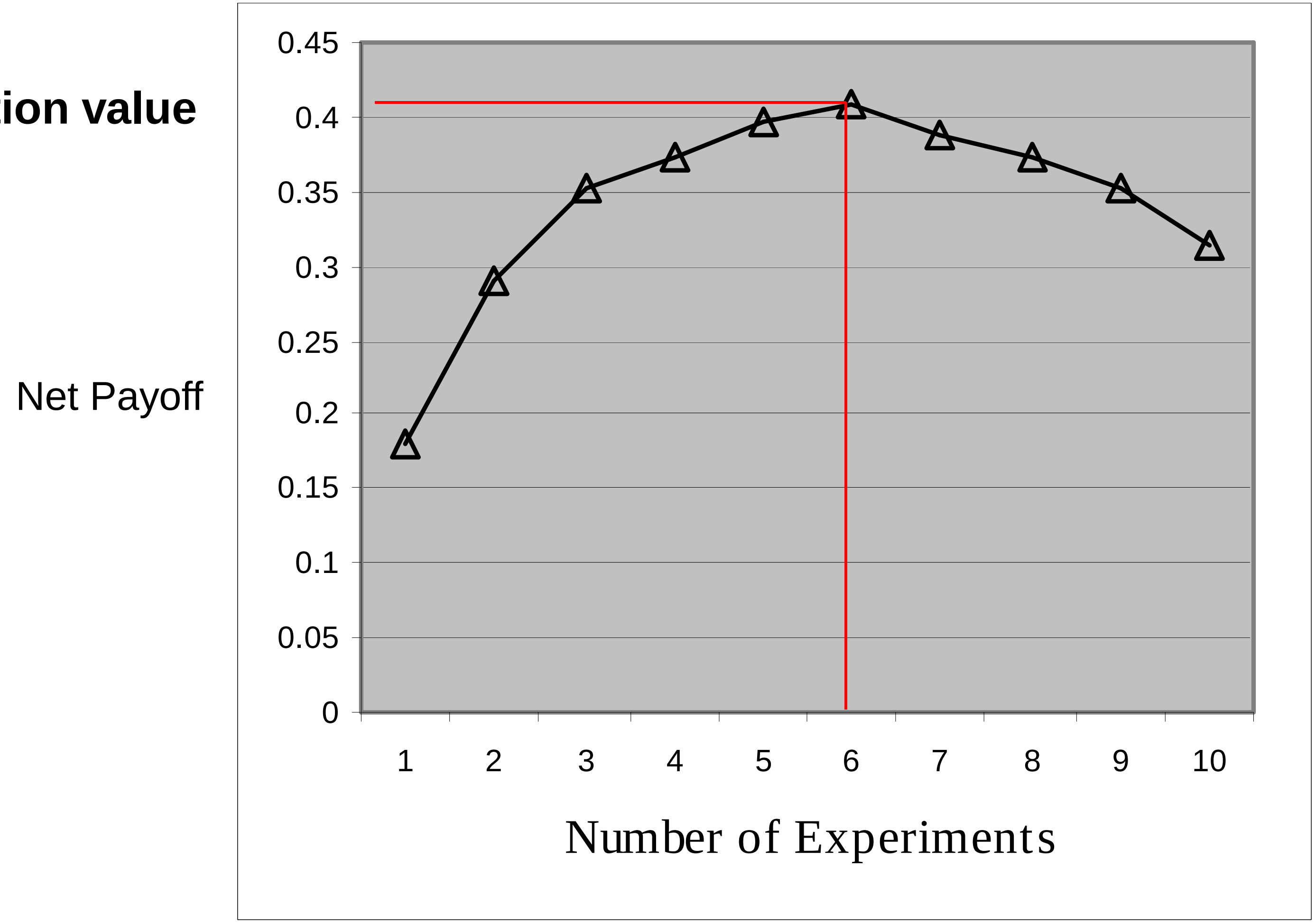
- System value = base value + value of options created by modules
- Option value of module is payoff on optimal R&D investment
- R&D means funding *k* projects in search of replacement
- Values of results assumed to be normally distributed
- Option value is payoff on best choice of *k:*
  expected value of best of *k* R&D products –
  cost of creating *k* them –
  cost of ripple effects when substituting in a replacement

Baldwin & Clark Model

- R&D costs increase linearly with k
- Likelihood of finding high valued result depends on variance
- Diminishing returns on search (process of sampling distribution)
- Leads to characteristic payoff curve
- Option value is at peak

R&D costs increase linearly with k vs. diminishing returns
Variance is critical parameter determining benefits of R&D

option value



Optimal number of experiments = 6

Baldwin/Clark Design Options Valuation Model

Technical Potential

Square Root Complexity

Best of k_i positive draws from s.n.d.

$$NOV_i = max_{ki} \{ \sigma_i n_i^{1/2} Q(k_i) - C_i(n_i)k_i - Z_i \}$$

Benefit of k_i experiments for module complexity n_i and technical potential σ_i

Cost of k_i experiments on module *i*

Cost to substitute (visibility)

So What

- Vision of future in which software developers **decide** on basis of economics, in general, and value of flexibility in particular
- Development **environments** that display not only code and technical models, but all major valuable assets, including **decision rights**, and key parameters, especially **uncertainty, evolving** over time
- Dynamic **management** of decision rights (**options**) in face of ongoing resolution of risk

Difficulties

- Some work on “real options” problematical, e.g., applying Black & Scholes to risks not implicitly priced by market; Baldwin and Clark model not strongly validated yet
- Expert subjective risk judgments necessary; no silver bullet
- Unreasonable to ask developers to understand the math
- Modeling and analysis can be intractable at scale
- Need to start to prototype some usable tools

Vision

- Large-scale management of decision rights
- When to invest in creating them
- How they are produced as side effects of other decisions
- How to value them
- When to exercise them
- As an integral part of software development process

Reading

- Kevin Sullivan et al., “Software Design as an Investment Activity,” in Trigeorgis, ed. *Real Options and Business Strategy*, Risk Books, 1999.

- Carliss Baldwin and Kim Clark, *Design Rules*, MIT Press, 2000.

More Information

sullivan@virginia.edu

Modeling Assumptions

- N is number of design parameters. $N = 13$ or $N = 16$.
- Given a module of p parameters, its complexity is $n = p/N$.
- The visibility cost of a module i of size n is $Z_i = \Sigma_{\text{j sees i}} cn$.
- Value of one experiment on an unmodularized design, $\sigma N^{1/2} Q(1) = 1$, is the value of the original system.
- The design cost $c = 1/N$ of each design parameter is the same, and the cost to redesign the whole system is $cN = 1$.
- One experiment on unmodularized system just breaks even: $\sigma N^{1/2} Q(1) - cN = 0$.

Generalized Valuation

- Model multi-period uncertainty as deeper tree
- Option value at time t node v is maximum of immediate payoff and expected t+1 payoff
- With finite time horizon can use backwards recursion approach to compute option value

Prior Work

- Baldwin & Clark options value of modularity
- Lots of work on real options in capital budgeting [Dixit & Pindyck, Trigeorgis]
- Kumar and others valuing flexible manufacturing
- Withey 96, Favaro 98 reuse investment analysis

Basis for Economic Reasoning about Risky Decisions

- Effects of uncertainty over benefits
- Effects of direct cost to exercise option
- Effects of uncertainty over direct cost
- Effect of probability of favorable outcome