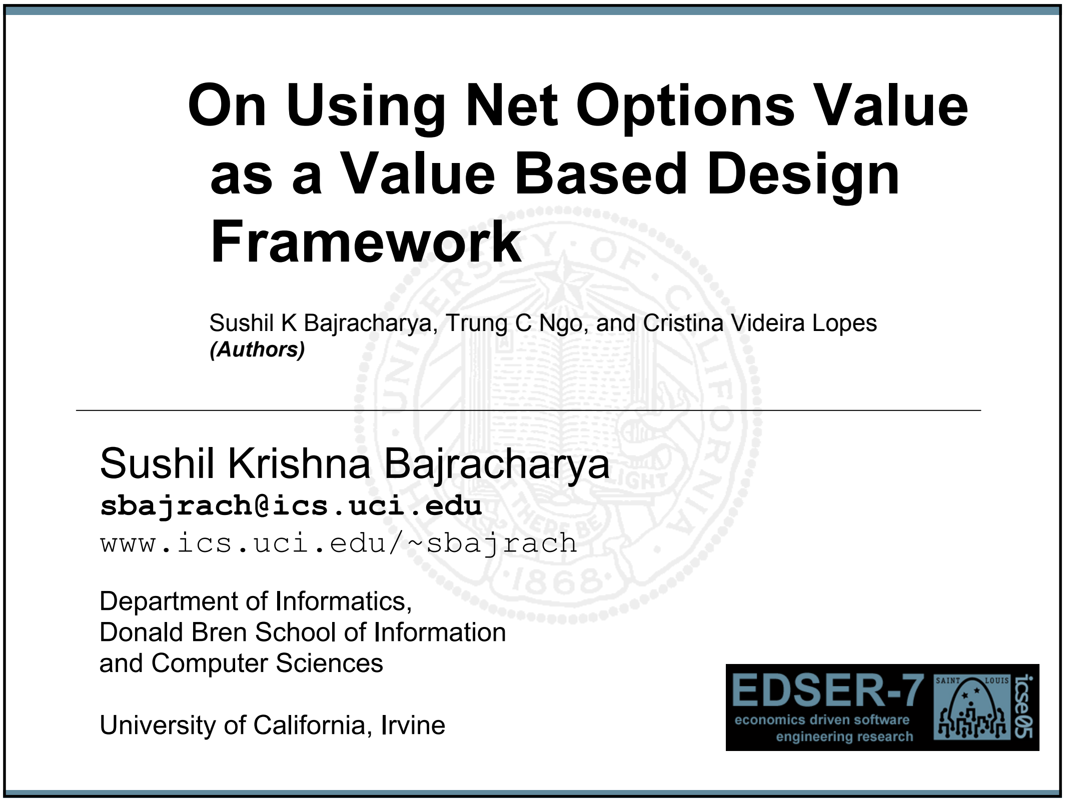


On Using Net Options Value as a Value Based Design Framework

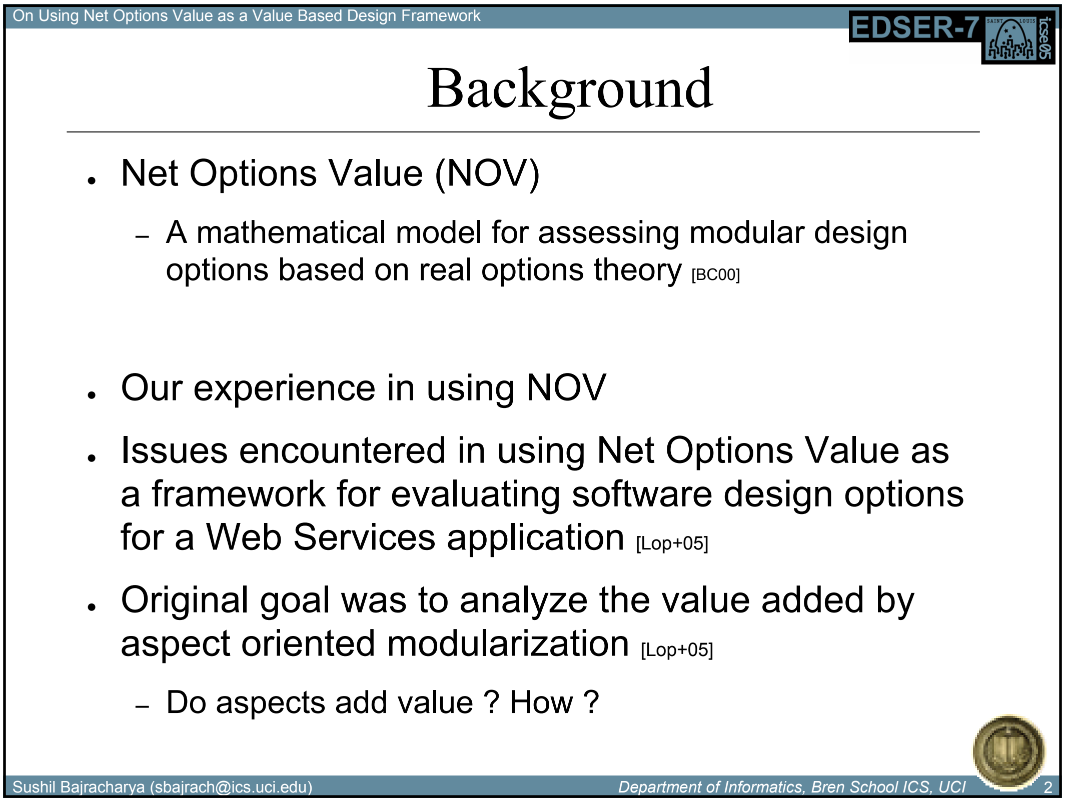
Sushil K Bajracharya, Trung C Ngo, and Cristina Videira Lopes
(Authors)

Sushil Krishna Bajracharya
sbajrach@ics.uci.edu
www.ics.uci.edu/~sbajrach

Department of Informatics,
Donald Bren School of Information
and Computer Sciences

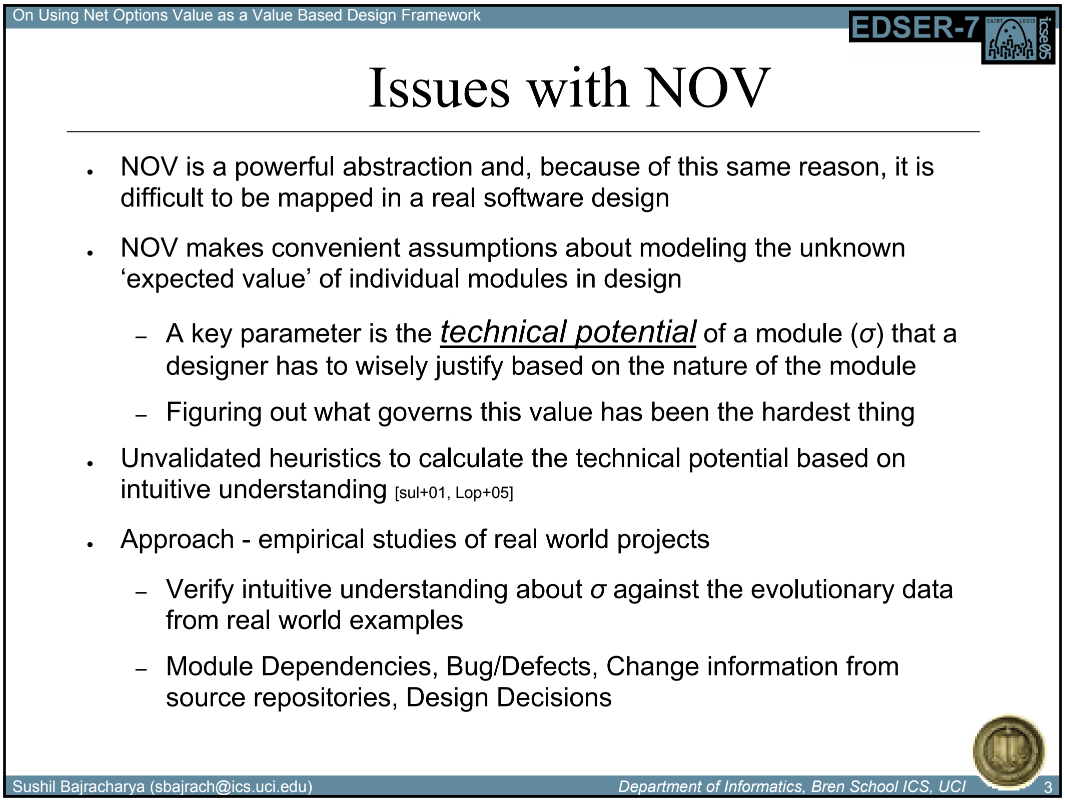
University of California, Irvine

EDSER-7 economics driven software engineering research icse05



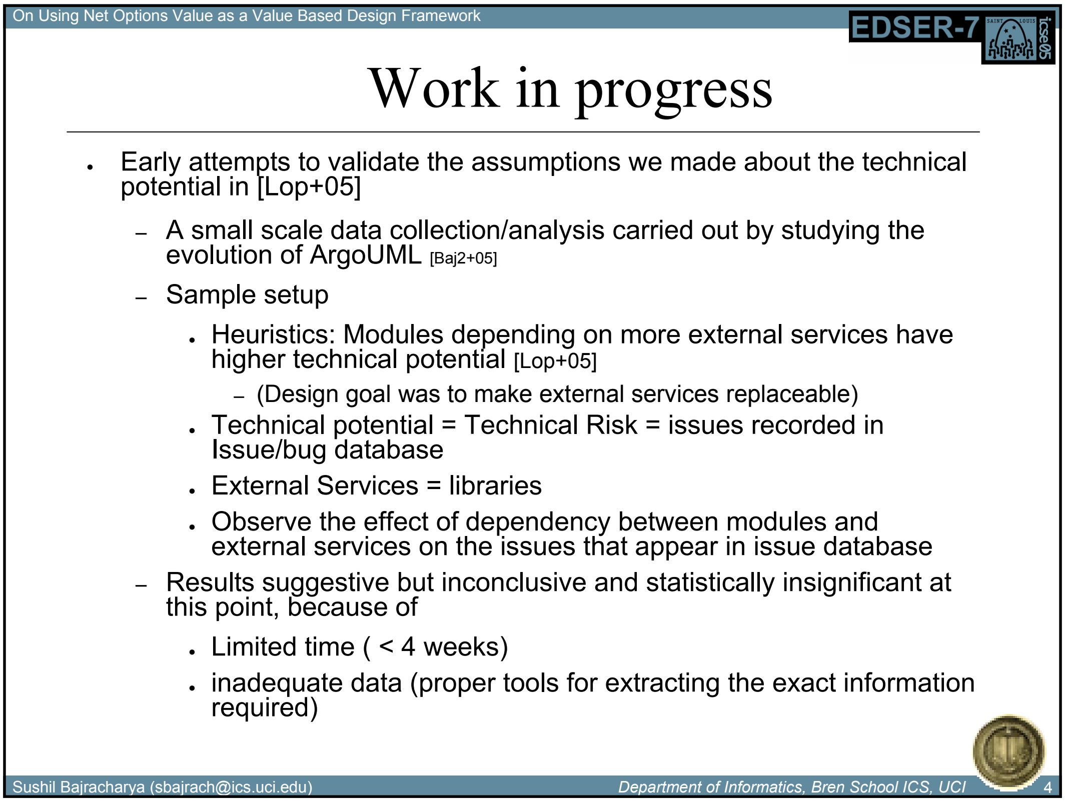
Background

- Net Options Value (NOV)
  - A mathematical model for assessing modular design options based on real options theory [BC00]
- Our experience in using NOV
- Issues encountered in using Net Options Value as a framework for evaluating software design options for a Web Services application [Lop+05]
- Original goal was to analyze the value added by aspect oriented modularization [Lop+05]
  - Do aspects add value ? How ?



Issues with NOV

- NOV is a powerful abstraction and, because of this same reason, it is difficult to be mapped in a real software design
- NOV makes convenient assumptions about modeling the unknown 'expected value' of individual modules in design
  - A key parameter is the *technical potential* of a module (σ) that a designer has to wisely justify based on the nature of the module
  - Figuring out what governs this value has been the hardest thing
- Unvalidated heuristics to calculate the technical potential based on intuitive understanding [sul+01, Lop+05]
- Approach - empirical studies of real world projects
  - Verify intuitive understanding about σ against the evolutionary data from real world examples
  - Module Dependencies, Bug/Defects, Change information from source repositories, Design Decisions



Work in progress

- Early attempts to validate the assumptions we made about the technical potential in [Lop+05]
  - A small scale data collection/analysis carried out by studying the evolution of ArgoUML [Baj2+05]
  - Sample setup
    - Heuristics: Modules depending on more external services have higher technical potential [Lop+05]
      - (Design goal was to make external services replaceable)
    - Technical potential = Technical Risk = issues recorded in Issue/bug database
    - External Services = libraries
    - Observe the effect of dependency between modules and external services on the issues that appear in issue database
  - Results suggestive but inconclusive and statistically insignificant at this point, because of
    - Limited time (< 4 weeks)
    - inadequate data (proper tools for extracting the exact information required)

Challenges

- Challenges
  - The real problem in finding good example systems, especially with interesting evolution history
  - Issues with data collected from open source projects
  - Requires expertise in experimentation and statistical techniques

Reference

- [BC00] C. Y. Baldwin and K. B. Clark. Design Rules vol I, The Power of Modularity. MIT Press, 2000
- [sul+01] K. J. Sullivan, W. G. Griswold, Y. Cai, and B. Hallen. The structure and value of modularity in software design. ACM SIGSOFT 2001
- [Lop+05] C. V. Lopes and S. K. Bajracharya. An analysis of modularity in aspect oriented design. In *AOSD '05: Proceedings of the 4th international conference on Aspect-oriented software development*, pages 15 - 26,New York, NY, USA, 2005. ACM Press.
- [Baj+05] Sushil Krishna Bajracharya, Trung Chi Ngo and Cristina Vidiera Lopes. On Using Net Options Value as a Value Based Design Framework. *EDSER-7*, ICSE 2005
- [Baj2+05] Sushil Krishna Bajracharya and Trung Chi Ngo. Characterizing the technical potential of a Software Module. Manuscript*, March 2005, Bren School ICS*. (available online: http://www.ics.uci.edu/~sbajrach/Research/course-writings/w05-ics225.pdf)