CSE 4621 Software Metrics and Models (3 credits)

Primary instructor: Cem Kaner **Supporting instructor:** Keith Gallagher

Textbooks and references:

N. Fenton, Software Metrics: A Rigorous and Practical Approach. 3rd edition. CRC Press, 2014. (T)

L. Bossavit, The Leprechauns of Software Engineering. 2013. (T)

R. Austin, Measuring and Managing Performance in Organizations. Dorset House Publishing, 1996. (T)

L. H. Putnam, Five Core Metrics: The Intelligence Behind Successful Software Management. Dorset House Publishing, 2003. (R)

W. Shadish, Experimental & Quasi-Experimental Designs for Generalized Causal Inference. Wadsworth Publishing, 2001. (R)

Course information:

2014–2015 Catalog description: CSE 4621 Software Metrics and Modeling (3 credits). Examines common software metrics, axiomatic foundations of measurement, validity of measurements and measurement dysfunction, and some statistical and modeling approaches to help students make their software measurements meaningful. Prerequisites: CSE 2400, CSE 2410.

Prerequisites by topic: Experience with or knowledge of the software development life-cycle, familiarity with basic descriptive statistics, e.g., sample mean and standard deviation, histograms, discrete mathematics, including study of the properties of real numbers, familiarity with Poisson process models.

Place in program:

Computer Science Program: Advanced elective

Software Engineering Program: Required

Course outcomes & related student outcomes: The student will be able to

1. Be aware of connections of software engineering measurements with the multidisciplinary history of measurement theory. (2: Scientific, computing, and engineering problem solving & 4: Apply mathematical, scientific, algorithmic, and theoretical principles, to model, design and evaluate software systems and processes)
2. Be aware of the social and ethical issues associated with human performance measurement (which includes most of the measures taken by software engineers). (5: Awareness of professional issues and responsibilities)
3. Assess the quality of a proposed metric. (4: Apply mathematical, scientific, algorithmic, and theoretical principles, to model, design and evaluate software systems and processes)

4. Understand the commercial and organizational contexts of any metric. (5: Awareness of professional issues and responsibilities & 6: Analyze computing's impact)
5. Appreciate fact-based cost/benefit analysis and optimization. (2: Scientific, computing, and engineering problem solving)
6. Become familiar with several common measures, including the ability to calculate some of them. (2: Scientific, computing, and engineering problem solving)
7. Use idea generators, such as Goal-Question-Metric, to develop a measurement strategy for answering questions of interest. (2: Scientific, computing, and engineering problem solving)
8. Describe uses of converging measures and contrasting measures (such as balanced scorecards) for reducing the severity of measurement distortion. (2: Scientific, computing, and engineering problem solving & 4: Apply mathematical, scientific, algorithmic, and theoretical principles, to model, design and evaluate software systems and processes)
9. 9. Write a critical and thorough review of the research literature associated with a specific metric. (7: Communicate effectively)

Topics covered:

1. The nature of problems addressed by software metrics (3 hours)
2. Introduction to measurement theory (3 hours)
3. Goal-based framework for software metrics (3 hours)
4. Review of statistical methods and experimental design (3 hours)
5. Analysis of software measurement data (3 hours)
6. Measurement of internal product attributes, such as size and structure; external product attributes, such as value and reliability; project attributes, such as cost, schedule, resource, and productivity; distortion and dysfunction (15 hours)
7. Validity of metrics (3 hours)
8. How to navigate and critique the metrics-related research literature (3 hours)
9. Metrics challenges and practices in other disciplines (3 hours)
10. Evaluating patterns instead of single data points (1 hours)

Approved by: Cem Kaner, Professor & Heather Crawford, Assistant Professor

Signature: _____ **Date:** 2/2/15

Signature: _____ **Date:** 10F15