

# Software Metrics and Quality Engineering

CSE 8314 — Fall 2015

Prof. Jeff Tian, [tian@lyle.smu.edu](mailto:tian@lyle.smu.edu)  
CSE, SMU, Dallas, TX 75275  
(214) 768-2861; Fax: (214) 768-3085  
[www.lyle.smu.edu/~tian/class/8314.15f](http://www.lyle.smu.edu/~tian/class/8314.15f)

## **Module I: Metrics/QE Overview**

- About CSE 8314
- Software Measurement
- ESE, GQM, etc.
- Comparison, Evaluation and M. Theory

## Goal of Software Measurement

---

To achieve the goal of controlled software development, we need to:

- Develop an *engineering* discipline;
- Measure and evaluate the working product;
- Construct a *scientific* model for program measurement:
  - ▷ Techniques from other disciplines;
  - ▷ Develop new techniques if necessary;
  - ▷ Basic questions:
    - What to measure: goal & environ.
    - How to measure it: metrics & tools
    - Selection and validation
- See also measurements and models from Tian/SQE Chapter 19.

---

## How Does CSE 8314 Fit In?

---

- (Area I) M/QE fundamentals:
  - ▷ Generic concepts, important ideas
  - ▷ Overall framework.
  
- (Areas II&III) basic metrics:
  - ▷ External metrics
  - ▷ Internal metrics
  - ▷ Relations, classification, usage
  
- (Area IV) Metrics evaluation:
  - ▷ Empirical  $\Rightarrow$  formal model.
  - ▷ Formal models for metrics evaluation.
  
- (Area V) New frontier:
  - ▷ Hypothesis testing using metrics
  - ▷ Bigger picture (ESE) + new applications/frontier

## CSE 8314 Overview

---

- (Area I) M/QE fundamentals:
  - ▷ Generic concepts, important ideas
  - ▷ Overall framework.
  
- General concepts
  - ▷ Measurement, ESE, and SE
  - ▷ Measurement of software vs. measurement of other objects
  - ▷ Measurement "maturity" and spectrum
  
- Overall framework.
  - ▷ GQM/QIP/EF
  - ▷ Other frameworks
  
- Mathematical foundation
  - ▷ Measurement theory
  - ▷ Types and levels of measurement

---

## CSE 8314 Overview

---

- (Areas II&III) basic metrics:
  - ▷ External metrics
  - ▷ Internal metrics
  - ▷ Relations, classification, usage
  
- External metrics
  - ▷ Quality: reliability, safety, dependability, usability, etc.
  - ▷ Cost related
  - ▷ Time/schedule/activity/environment/etc.
  - ▷ Areas and contexts: PPP
  
- Internal metrics
  - ▷ Complexity
  - ▷ Dimensions and classification of complexity metrics

---

## CSE 8314 Overview

---

- (Area IV) Metrics evaluation:
  - ▷ Empirical  $\Rightarrow$  formal model.
  - ▷ Formal models for metrics evaluation.
  
- Empirical evaluation
  - ▷ Data and statistical analysis
  - ▷ Other empirical evidence/corroborations
  
- Formal models for metrics evaluation.
  - ▷ Historical development
  - ▷ Tian-Zelkowitz model
  - ▷ Other recent developments

## CSE 8314 Overview

---

- (Area V) New frontier:
  - ▷ Hypothesis testing using metrics
  - ▷ Bigger picture (ESE) + new applications/frontier
  
- Hypothesis testing using metrics:  
Koru-Tian and other works
  
- ESE, the bigger picture
  - ▷ ESE ideas and guidelines
  - ▷ Applications and examples
  
- New applications/frontier/development
  - ▷ Traditional: commercial, telecom, etc.
  - ▷ New: net-centric, SOA, cloud, etc.

---

## Complexity and Other Measurement

---

**Basic assumption:** The lower the complexity, the more desirable:

- ▷ cheaper to build;
- ▷ easier to maintain;
- ▷ more reliable;
- ▷ ...

### Usage of Complexity Measurement:

| activity   | time         | nature  |
|------------|--------------|---------|
| assessment | a posteriori | passive |
| prediction | a priori     | passive |
| control    | persistent   | active  |

**Other Measurement** : Internal and external ones (next).

---

## Internal/External Measures

---

**Internal Measures:** depend on programs only.  
complexity measures  $\subset$  internal measures;

**External Measures:** depend also on other external factors — so called *-lities*.

**Relations:** correlated but not uniquely determined. To use internal measures to predict external measure, we need:

- ▷ Discover *appropriate* internal measures;
- ▷ Establish *predictive* relations;
- ▷ Use and validate predictions.

## Measures and Dimensions

---

Complexity measures are multi-dimensional because of:

1. Multi-facet *internal* organization:

- ▷ Presentation;
- ▷ Control;
- ▷ Data.

2. Multi-purpose *external* usage under different activities.

- ▷ Assessment: Basili's GQM;
- ▷ Prediction: Boehm's COCOMO;
- ▷ Control: Boehm's spiral.

---

## Software Measurement

---

- In the measurement spectrum: "maturing"
  
- Example: testing evaluation
  - ▷ Test results and expenditure.
  - ▷ Test cases and measurement.
  - ▷ Internal measurements: size/complexity/etc.
  - ▷ Environmental data: process/people/setup
  - ▷ Evaluation results: reliability.
  
- Data/analysis from other phases:
  - ▷ Product: code, documents etc.
    - external: quality, cost, schedule etc.
  - ▷ Process: entities/relations/environment
  - ▷ People: experience etc.
  - ▷ Various assessment/prediction/improvement.

## Software Product Measurement

---

- Product specific (static):
  - ▷ Code, test case, document
  - ▷ Structure vs. information flow
  - ▷ Control/data/presentation
  - ▷ Metrics and data collection
  - ▷ ESE: product quality/etc. questions?
  
- Execution specific (dynamic):
  - ▷ Path verification (white-box)
  - ▷ Usage to component mapping (black-box)
  - ▷ Measurement along the path
  - ▷ Usage of the measurement data
  - ▷ ESE: performance/reliability/etc.?

## Other Software Measurement

---

- Process characteristics
  - ▷ Entities, relationships, and integration
  - ▷ Preparation, execution and followup
  
- People characteristics
  - ▷ Skills and experience
  - ▷ Roles: planners/developers/testers
  - ▷ Process management and teams
  
- Environmental characteristics
  - ▷ Hardware/software environment
  - ▷ Product/market environment

---

## Measurement and ESE

---

- *Empirical Software Engineering (ESE)*:  
Applying empirical techniques/methods to solve software engineering problems.
  
- Objects of study:
  - ▷ Observation of SE activities.
  - ▷ Case studies in SE.
  - ▷ Controlled experiments.
  
- Analysis and conclusions:
  - ▷ Data from the above activities.
  - ▷ Statistical and other analyses.
  - ▷ Conclusions draw based on data/analyses.
  
- Measurement plays a central role in ESE.

## Software Engineering Perspective

---

- Key components of S/W Eng.
  - ▷ Methods and processes
  - ▷ Formal foundations (math/theory)
  - ▷ Experimentation (scientific)
  
- Methods and process
  - ▷ Methods and methodologies
    - structured programming, OO, SOA
    - specialized methods
      - specification: formal vs informal
      - testing: black-box/white-box/random
  - ▷ Process models (and measurement)
  - ▷ Mixing method and process
    - agile, XP, TDD, etc.
    - clean room example
  - ▷ 7313, 7314 and other MS/CS courses.

---

## Software Engineering Perspective

---

- Formal foundations
  - ▷ Mathematics/logic/statistics
    - formal specifications
    - program verification
    - statistical models
  - ▷ Computer science
    - language and ADT  $\Rightarrow$  OO
    - systems/tools/CASE
  - ▷ Formal models on metrics: Area IV.
  
- Experimentation (scientific)
  - ▷ Trace/case studies
  - ▷ Controlled experiment
  - ▷ Measurement and analysis
  - ▷ Empirical validation
  - ▷ Observation-based vs. goal-oriented

---

## Software Processes

---

- Mega-Process: Initiation, Development, Maintenance, Termination.
  
- Components: Requirement, Specification, Design, Coding, Testing, Release.
  
- Process Variations:
  - ▷ Waterfall: sequence and dependencies;
  - ▷ Iterative: incremental, divide&conquer;
  - ▷ Spiral: risk management;
  - ▷ Mixed/synthesized.
  
- Measurement and analysis throughout different components of the products and processes.
  
  
- Relation to CSE 7313, 7314, etc.

---

## ESE in SE Activities

---

- Observational studies:
  - ▷ Passive observations of industrial practice, etc.
  - ▷ Try to draw preliminary conclusions based on observations and related data.
  - ▷ Multiple observations  $\Rightarrow$  validation.
  
- Case studies:
  - ▷ Semi-active.
  - ▷ Pre-set study goals.
  - ▷ Conclusions need further validation.
  
- Controlled experiments:
  - ▷ Active design and experimentation.
  - ▷ Closest to scientific experiments.
  - ▷ Solid conclusions.

---

## Measurement Framework: GQM

---

- Background:
  - ▷ Software Engineering Laboratory
  - ▷ TAME projects
  - ▷ Key personal: Basili et al.
  
- Software Engineering Laboratory
  - ▷ NASA/GSFC
  - ▷ University of Maryland
  - ▷ Computer Sciences Corp.
  - ▷ 1st SEI process award recipient
  - ▷ Software measurement and ESE:
    - among the first ESE studies
    - software measurement and analysis
    - goal-question-metric (GQM) paradigm
    - experience factory (EF)

## GQM

---

- GQM: what is it?
  - ▷ Goal: goal of the (measurement) study.
  - ▷ Questions: questions related to goals.
  - ▷ Metrics: metrics answering questions.
  
- GQM background/foundations:
  - ▷ Goal oriented approach.
  - ▷ Measurement based.
  - ▷ Scientific experimentation.
  - ▷ Hierarchy or paradigm: diagram.
  
- Relation to ESE:
  - ▷ Can serve as general guidelines for ESE.
  - ▷ Related EF: similar to scientific labs in ESE.

## GQM and EF

---

- EF: What is it?
  - ▷ Experience Factory
  - ▷ Separation of concerns
  - ▷ In connection with GQM/TAME
  - ▷ In ESE: Similar to scientific labs that conducts scientific experiments.
  
- Experience Factory
  - ▷ Input from product organization
  - ▷ Output to product development
  - ▷ Internal organization
  - ▷ Implementation in NASA/SEL

---

## GQM/EF Recent Development

---

- Research activities:
  - ▷ New NSF-funded Center:
    - Univ. Maryland and USC (Boehm)
    - GMQM and other activities
  - ▷ Fraunhofer Institute and Centers
  - ▷ Others
  
- GQM extensions:
  - ▷ GMQM: success model
  - ▷ Specialized guidelines
    - Kitchenham et al.
    - Tian measurement/model, etc.
  - ▷ More emphasis on scientific experimentation
  
- EF beyond NASA/SEL.

---

## GQM/EF Work at SMU

---

- NSF Net-Centric and Cloud Software and Systems I/UCRC
  - ▷ SMU/UNT/UTD founding members
  - ▷ EF for industrial partners of I/UCRC
  - ▷ Quality/dependability/performance
  - ▷ Application domain:
    - net-centric, service-oriented, and
    - more recently cloud computing
  
- MRI and other projects
  - ▷ Instruments for dependability evaluation for CCS (cloud computing systems)
  - ▷ Again, a kind of EF
  
- J. Tian, An Emerging Experience Factory to Support High-quality Applications Based on Software Components and Services. *Journal of Software*, Vol.6, No.2, pp.289-297, Feb., 2011.

## Measurement: Comparison

---

- Physics(and other physical sciences)
  - ▷ Motion: static and dynamic aspects
    - distance metrics
    - time metrics
    - energy, force, etc.
    - other metrics: speed, acceleration etc.
  - ▷ Similar for other areas in physics:
    - heat, sound, electricity, atomic/nuclear
  
- Some common characteristics
  - ▷ Well-defined, quantitative metrics:
    - usually interval or ratio type (later)
    - "unit": important standard of references
  - ▷ Importance role of measurement and data in "scientific" experiment: observation, hypothesis testing.
  - ▷ Theory and models: both basis and guide

## Measurement: Comparison

---

- Other "hard" sciences:
  - ▷ Chemistry: mostly quantities (other focus: reactions, pathways, etc.)
  - ▷ Biological/life sciences:
    - similar role of measurement
  - ▷ Geo-/astro-/etc.
    - important role of measurement
  
- Engineering:
  - ▷ Mechanical/civil/chemical/electrical/etc.
  - ▷ Emerging: measurement also maturing.
  - ▷ Measurement: Similar to foundational scientific disciplines
  
- Software measurement: "maturing" towards these

## Measurement: Comparison

---

- Psychology:
  - ▷ Example: IQ test and IQ score
  - ▷ Less well-defined
    - subjective vs objective
    - data validity and interpretation
    - usually unit-less
  - ▷ Other quantitative measurements and statistics
  - ▷ Non-quantitative: classification/type
  
- Other "soft" sciences and disciplines:
  - ▷ Social sciences, humanities, arts
  - ▷ Type of measurement as a distinguishing factor
  
- Software measurement: more "mature" than these?

## Measurement in ESE

---

- Measurement: central activity in ESE
  - ▷ context of measurement/expr/study
  - ▷ measurements associated with different experimental designs
  - ▷ measurement and data collection
  - ▷ measurement result analysis
  - ▷ measurement/analysis result presentation, interpretation, and drawing conclusions
  
- Interpreted as measurement activities:
  - ▷ definition: context, design
  - ▷ gathering: data collection
  - ▷ analysis/lookup:  
analysis, presentation, interpretation

## Measurement: Evaluation

---

- Measurement typically used to evaluation SE artifacts/activities.
  
- Also need to evaluate measurements/metrics themselves:
  - ▷ properly defined?
  - ▷ properly used?
  - ▷ lead to useful results?
  
- Use of evaluation results:
  - ▷ selecting existing measures/metrics
  - ▷ proposing new ones
  - ▷ under what context?

## Measurement: Evaluation

---

- Types of metrics evaluation:
  - ▷ self evaluation
  - ▷ empirical evaluation
  - ▷ formal model based evaluation
  
- Self evaluation of new metrics:
  - ▷ when proposed/defined
  - ▷ demonstrate the use & usefulness
  - ▷ possible subjective bias
  - ▷ limited scope & validity

## Measurement: Evaluation

---

- Empirical evaluation of metrics:
  - ▷ a set of given metrics
  - ▷ empirical study set up
  - ▷ focus: how these metrics work
  - ▷ other performance measures not subjected to evaluation
  - ▷ typical evaluation objects:  
internal (complexity) metrics
  
- Evaluation based on formal models:
  - ▷ based on empirical studies/evidences
  - ▷ generalized theory/models
  - ▷ development: after many empirical evaluation studies
  
- More later (Area IV of CSE 8314).

---

## Measurement Theory

---

- Best book on the subject:  
Fred S. Roberts, "Measurement Theory, with applications to decision making, utility, and the social sciences", Addison-Wesley, 1979.
  
- Formalization of measurement:
  - ▷ R: relation  
reflexive, symmetric, transitive, complete?
  - ▷  $f(x)$ : measurement as functional mapping
  - ▷  $aRb \leftrightarrow f(a) \leq f(b)$
  
- Basic questions
  - ▷ representation: defines mapping/scale
  - ▷ meaningfulness: truth unchanged by admissible transformations

## Measurement Theory

---

- Scale types defined by admissible transformations
- Some common scale types (from strongest to weakest):
  - ▷ absolute:  $\phi(x) = x$ , e.g., counting
  - ▷ ratio:  $\phi(x) = \alpha x$ ,  $\alpha > 0$ , e.g., mass, temperature (K), time interval
  - ▷ interval:  $\phi(x) = \alpha x + \beta$ ,  $\alpha > 0$ , e.g., temperature (C, F), time (calendar), IQ standardized score
  - ▷ ordinal:  $x \geq y$  iff  $\phi(x) \geq \phi(y)$ , e.g., preference, hardness, air quality, IQ raw score
  - ▷ nominal: any one-to-one, label alt plans, CSE/EMIS/EE/etc course code