CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Object Oriented Software Engineering

LECTURE 17

I. METRICS FOR PROCESS & PROJECT

II. METRICS FOR SOFTWARE QUALITY

Introduction

- Within the context of the software process and the projects that are conducted using the process, a software team is concerned primarily with productivity and quality metrics—measures of software development “output” as a function of effort and time applied and measures of the “fitness for use” of the work products that are produced.

Software Measurement and Metrics

- **Software Measurement:** A measurement is an manifestation of the size, quantity, amount or dimension of a particular attributes of a product or process. Software measurement is a titrate impute of a characteristic of a software product or the software process. It is an authority within software engineering. Software measurement process is defined and governed by ISO Standard.

Classification of Software Measurement:

- There are 2 types of software measurement:
- **Direct Measurement:**
  In direct measurement the product, process or thing is measured directly using standard scale.
- **Indirect Measurement:**
  In indirect measurement the quantity or quality to be measured is measured using related parameter i.e. by use of reference.

Classification of Software Metrics:

- **Product Metrics:**
  Product metrics are used to evaluate the state of the product, tracing risks and undercovering prospective problem areas. The ability of team to control quality is evaluated.
- **Process Metrics:**
  Process metrics pay particular attention on enhancing the long term process of the team or organisation.
- **Project Metrics:**
  Project matrix is describes the project characteristic and execution process.
  - Number of software developer
  - Staffing pattern over the life cycle of software
  - Cost and schedule
  - Productivity

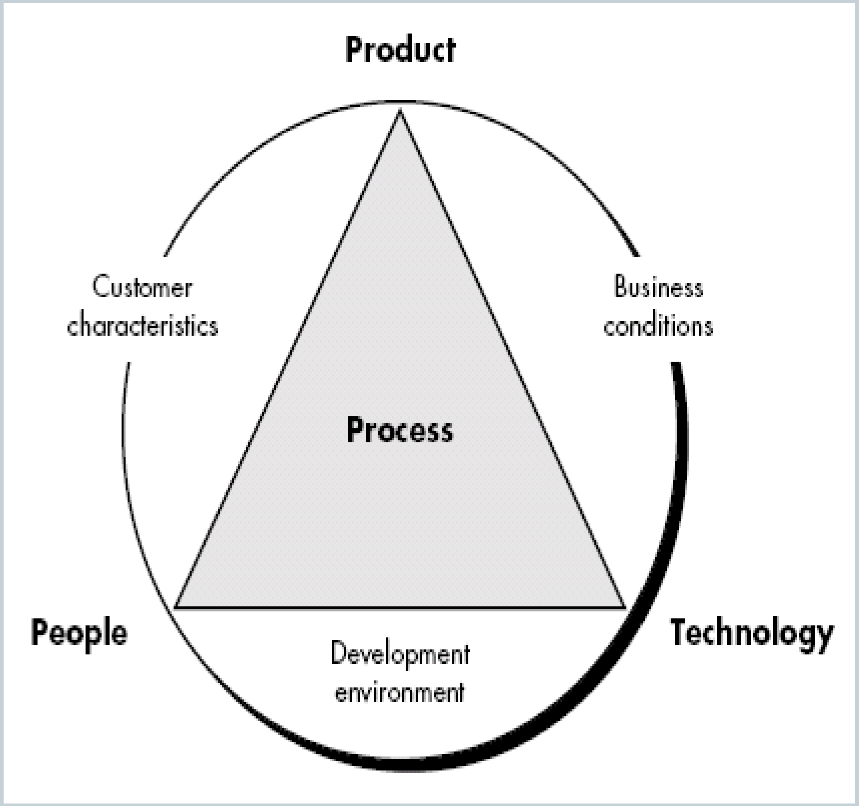
Process and Project Metrics

- **Process metrics** are collected across all projects and over long periods of time. Their intent is to provide a set of process indicators that lead to long-term software process improvement.
- **Project metrics** enable a software project manager to (1) assess the status of an ongoing project, (2) track potential risks, (3) uncover problem areas before they go “critical,” (4) adjust work flow or tasks, and (5) evaluate the project team’s ability to control quality of software work products.

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Process Metrics

- The only rational way to improve any process is to measure specific attributes of the process, develop a set of meaningful metrics based on these attributes, and then use the metrics to provide indicators that will lead to a strategy for improvement



Process Metrics

- The skill and motivation of people has been shown to be the single most influential factor in quality and performance. The complexity of the product can have a substantial impact on quality and team performance. The technology (i.e., the software engineering methods and tools) that populates the process also has an impact. In addition, the process triangle exists within a circle of environmental conditions that include the development environment (e.g., integrated software tools), business conditions (e.g., deadlines, business rules), and customer characteristics (e.g., ease of communication and collaboration).

Project Metrics

- Unlike software process metrics that are used for strategic purposes, software project measures are tactical. That is, project metrics and the indicators derived from them are used by a project manager and a software team to adapt project workflow and technical activities. The first application of project metrics on most software projects occurs during estimation. Metrics collected from past projects are used as a basis from which effort and time estimates are made for current software work. As a project proceeds, measures of effort and calendar time expended are compared to original estimates (and the project schedule). The project manager uses these data to monitor and control progress.

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Project Metrics

- The intent of project metrics is twofold. First, these metrics are used to minimize the development schedule by making the adjustments necessary to avoid delays and mitigate potential problems and risks. Second, project metrics are used to assess product quality on an ongoing basis and, when necessary, modify the technical approach to improve quality. As quality improves, defects are minimized, and as the defect count goes down, the amount of rework required during the project is also reduced. This leads to a reduction in overall project cost.

Software Measurement

Measurements in the physical world can be categorized in two ways: direct measures (e.g., the length of a bolt) and indirect measures (e.g., the "quality" of bolts produced, measured by counting rejects). Software metrics can be categorized similarly. Direct measures of the software process include cost and effort applied. Direct measures of the product include lines of code (LOC) produced, execution speed, memory size, and defects reported over some set period of time. Indirect measures of the product include functionality, quality, complexity, efficiency, reliability, maintainability, and many other "–abilities"

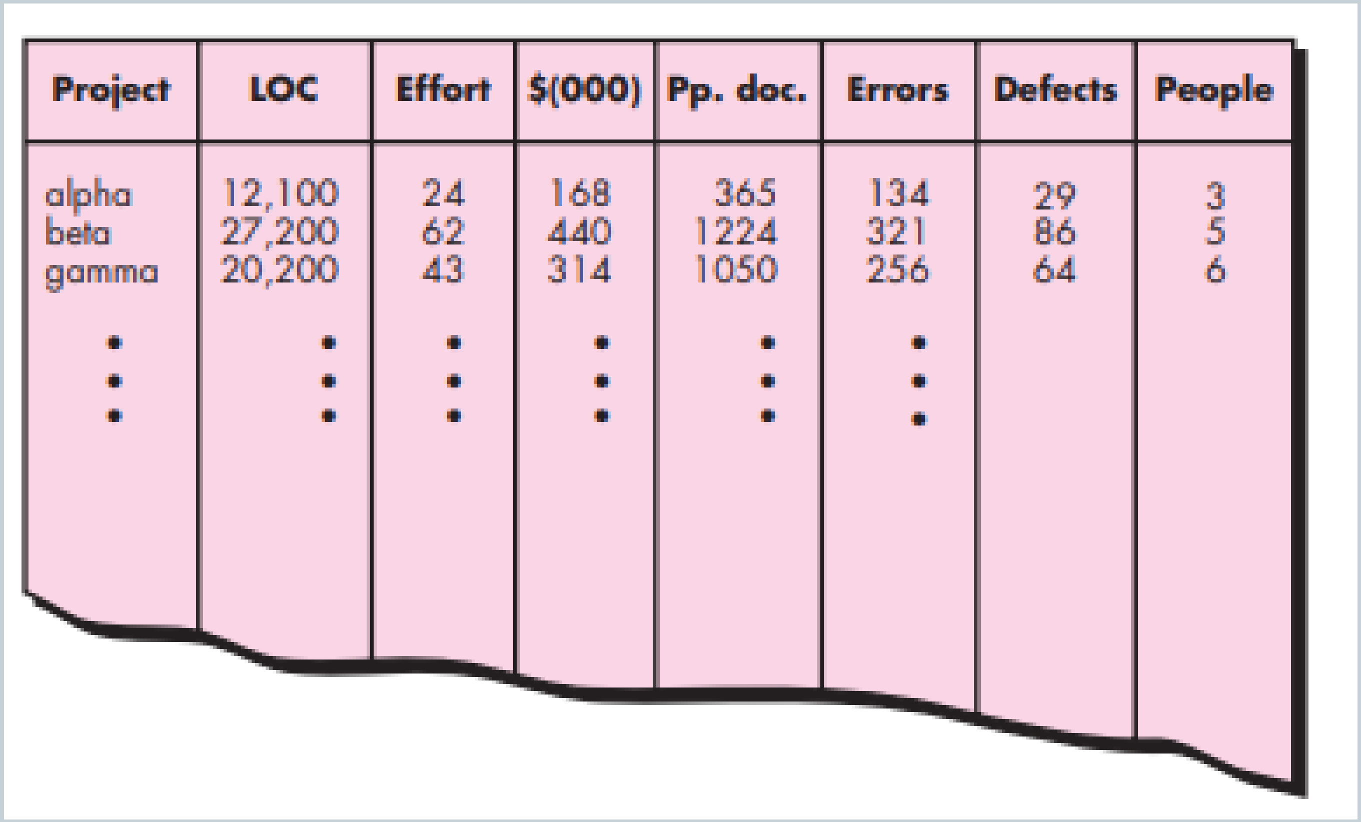
Software Measurement

Consider a simple example. Individuals on two different project teams record and categorize all errors that they find during the software process. Individual measures are then combined to develop team measures. Team A found 342 errors during the software process prior to release. Team B found 184 errors. All other things being equal, which team is more effective in uncovering errors throughout the process? Because you do not know the size or complexity of the projects, you cannot answer this question. However, if the measures are normalized, it is possible to create software metrics that enable comparison to broader organizational averages.

Size-Oriented Metrics

Size-oriented software metrics are derived by normalizing quality and/or productivity measures by considering the size of the software that has been produced. If a software organization maintains simple records, a table of size-oriented measures, such as the one shown in Figure (next page), can be created. The table lists each software development project that has been completed over the past few years and corresponding measures for that project. Referring to the table entry for project alpha: 12,100 lines of code were developed with 24 person-months of effort at a cost of $168,000. It should be noted that the effort and cost recorded in the table represent all software engineering activities (analysis, design, code, and test), not just coding. Further, information for project alpha indicates that 365 pages of documentation were developed, 134 errors were recorded before the software was released, and 29 defects were encountered after release to the customer within the first year of operation. Three people worked on the development of software for project alpha.

Size-Oriented Metrics



Project	LOC	Effort	$(000)	Pp. doc.	Errors	Defects	People
alpha	12,100	24	168	365	134	29	3
beta	27,200	62	440	1224	321	86	5
gamma	20,200	43	314	1050	256	64	6
⋮	⋮	⋮	⋮	⋮	⋮		

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Size-Oriented Metrics

- In order to develop metrics that can be assimilated with similar metrics from other projects, you can choose lines of code as a normalization value. From the rudimentary data contained in the table, a set of simple size-oriented metrics can be developed for each project:

• Errors per KLOC (thousand lines of code)

• Defects per KLOC

• $ per KLOC

• Pages of documentation per KLOC

- In addition, other interesting metrics can be computed:

• Errors per person-month

• KLOC per person-month

• $ per page of documentation

Function-Oriented Metrics

- Function-oriented software metrics use a measure of the functionality delivered by the application as a normalization value. The most widely used function-oriented metric is the function point (FP). Computation of the function point is based on characteristics of the software’s information domain and complexity.

Metrics for Software Quality

- The overriding goal of software engineering is to produce a high-quality system, application, or product within a time frame that satisfies a market need.
- The quality of a system, application, or product is only as good as the requirements that describe the problem, the design that models the solution, the code that leads to an executable program, and the tests that exercise the software to uncover errors. You can use measurement to assess the quality of the requirements and design models, the source code, and the test cases that have been created as the software is engineered. To accomplish this real-time assessment, you apply product metrics to evaluate the quality of software engineering work products in objective, rather than subjective ways.

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Measuring Quality

- **Correctness.** A program must operate correctly or it provides little value to its users. Correctness is the degree to which the software performs its required function. The most common measure for correctness is defects per KLOC, where a defect is defined as a verified lack of conformance to requirements.
- **Maintainability.** Software maintenance and support accounts for more effort than any other software engineering activity. Maintainability is the ease with which a program can be corrected if an error is encountered, adapted if its environment changes, or enhanced if the customer desires a change in requirements.

Measuring Quality

- **Integrity.** Software integrity has become increasingly important in the age of cyber terrorists and hackers. This attribute measures a system's ability to withstand attacks (both accidental and intentional) to its security. To measure integrity, two additional attributes must be defined: threat and security. Threat is the probability (which can be estimated or derived from empirical evidence) that an attack of a specific type will occur within a given time. Security is the probability (which can be estimated or derived from empirical evidence) that the attack of a specific type will be repelled.

Measuring Quality

- **Usability.** If a program is not easy to use, it is often doomed to failure, even if the functions that it performs are valuable. Usability is an attempt to quantify ease of use and can be measured in terms of the characteristics

Defect Removal Efficiency

- A quality metric that provides benefit at both the project and process level is defect removal efficiency (DRE). In essence, DRE is a measure of the filtering ability of quality assurance and control actions as they are applied throughout all process framework activities. When considered for a project as a whole, DRE is defined in the following manner:

$$\text{DRE} = \frac{E}{E + D}$$

where E is the number of errors found before delivery of the software to the end user and D is the number of defects found after delivery.

The ideal value for DRE is 1. That is, no defects are found in the software

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

THANK YOU