Software Project Management – Metrics for Project and Product Domains Software Measures, Metrics for Software Quality

Marwadi University

Computer Engineering Diploma

Unit 2:-
Project Management Concepts, Requirement Engineering & Metrics

Software Engineering (09CE2402)

Software Metrics

- Software metrics refers to a broad range of measurements for computer software.
- Measurement can be applied to the software process with the intent of improving it on a continuous basis.
- Measurement can be used throughout a software project to assist in estimation, quality control and project control.
- Measurement can be used by software engineers to help assess the quality of technical work products and to assist in tactical decision making as a project proceeds.

Reasons to Measure

- **To <u>characterize</u> in order to**
  - Gain an understanding of processes, products, resources, and environments
- **To <u>evaluate</u> in order to**
  - Determine status with respect to plans
- **To <u>predict</u> in order to**
  - Gain understanding of relationships among processes and products
  - Build models of these relationships
- **To <u>improve</u> in order to**
  - Identify inefficiencies and other opportunities for improving product quality and process performance

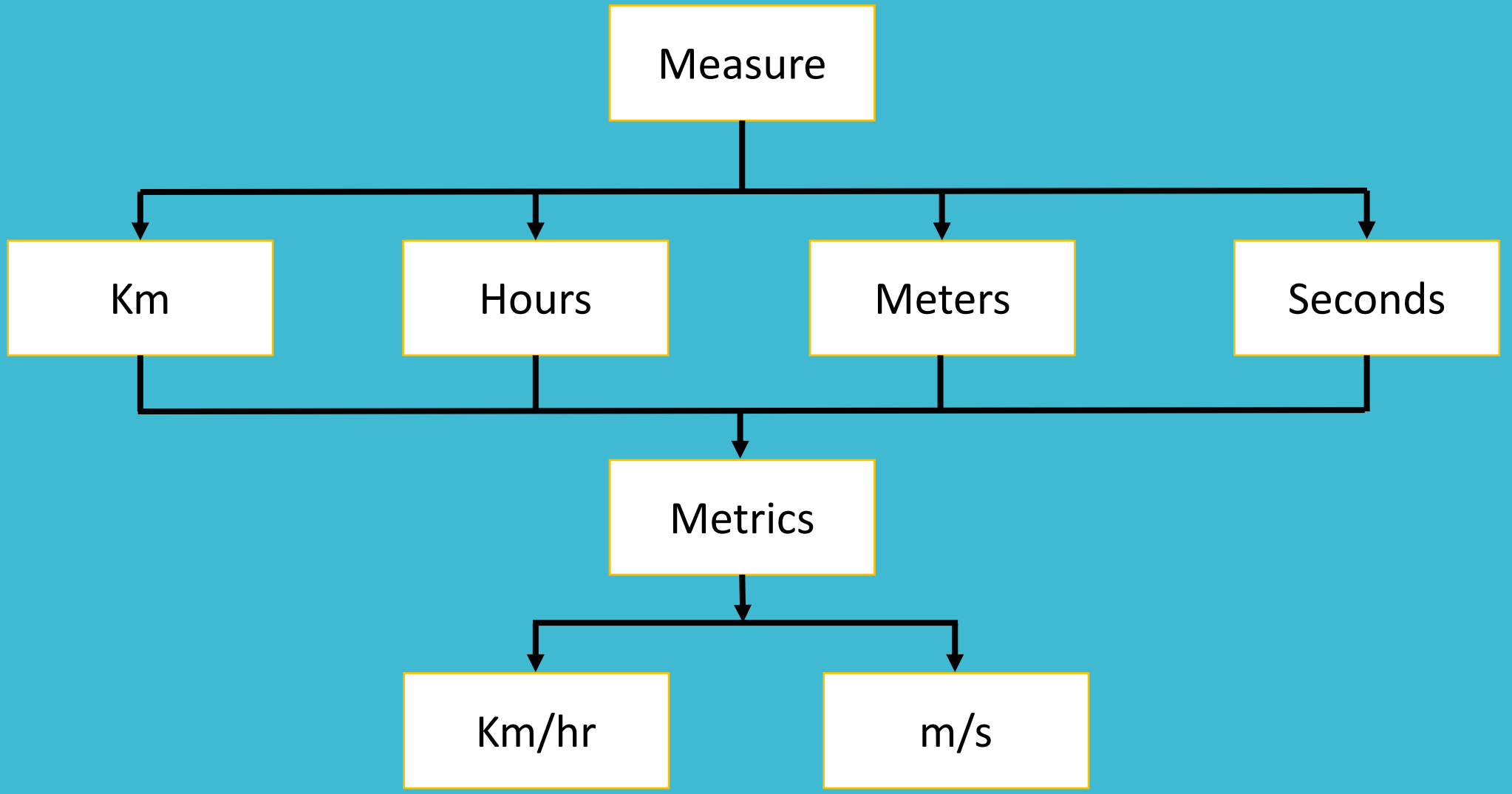
Measure and Metrics

- ***Measure*** A measure is a number or a quantity that records a directly observable value or performance.
- All measures are composed of a value (a number) and a unit of measure.
- The number provides magnitude for the measure (how much), while the unit gives number meaning (what is measured).
  - 1,234,567 Pageviews
  - 8,901,234 Sessions
  - 567,890 Facebook Likes

Measure and Metrics

- **Metrics:-** Metrics represent the different methods we employ to understand change over time across a number of dimensions or criteria. .

Measure and Metrics



INDICATOR

- An ***indicator*** is a metric or combination of metrics that provide insight into the software process, a software project, or the product itself.
- An indicator provides insight that enables the project manager or software engineers to adjust the process, the project, or the process to make things better.

Metrics for Process and Project Domain

Process Metrics

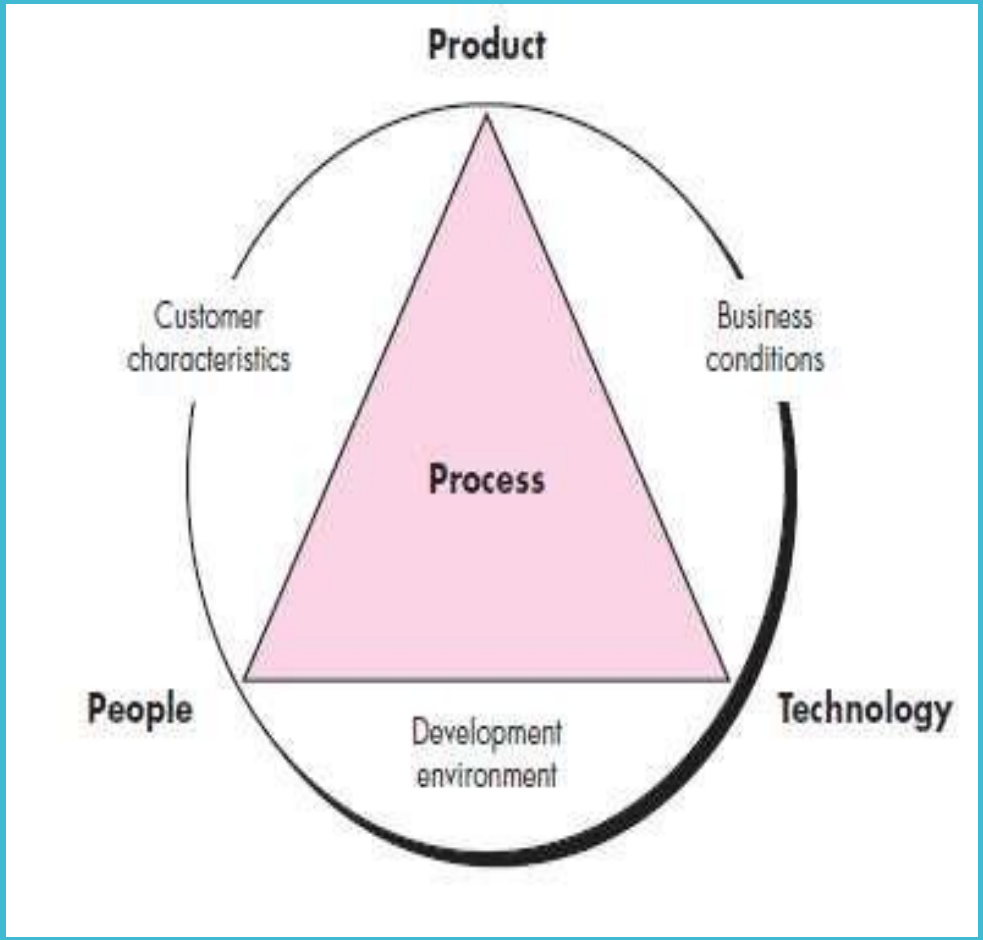
Process Metrics

Process metrics are collected across all projects and over long periods of time.

- They enable managers and practitioners to assess what works and what doesn't.
- Process metrics are measures of the software development process, such as
  - ❑ Overall development time
  - ❑ Type of methodology used
- Their intent is to provide indicators that lead to long- term software process improvement.

Process Metrics and Software Process Improvement:-

- Software process improvement, it is important to note that process is only one of a number of "controllable factors in improving software quality and organizational performance".
- Process sits at the center of a triangle connecting three factors that have a profound influence on software quality and organizational performance.



Process Metrics and Software Process Improvement:-

To improve any process, the rational way is:

- Measure Specific attributes of the process
- Derive meaningful metrics from these attributes.
- Use these metrics to provide indicators.
- The indicators lead to a strategy for improvement.

Process Metrics and Software Process Improvement

Statistical Software Process Improvement (SSPI):

- Errors uncovered before release of the software
- Defects delivered to and reported by end-users
- Work products delivered (productivity)
- Human effort expended
- Calendar time expended etc.
- Conformance to schedule

Project Metrics

Metrics In The Project Domains

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,

(5) evaluate the project team’s ability to control quality of software work products

Project Metrics

- Used by a project manager and software team to adapt project work flow and technical activities.
- Tactical and Short Term.

Purpose:

- Minimize the development time by making the necessary adjustments to avoid delays and *mitigate (less)* problems
- Assess product quality on an ongoing basis & modify the technical approach to improve quality.

Project Metrics

- Metrics collected from past projects are used as a basis from which effort and time estimates are made for current software project.
- As a project proceeds, actual values of human effort & calendar time expended are compared to the original estimates.
- This data is used by the project manager to monitor & control the project.

Project Metrics

Metrics:

- Effort or time per SE task
- Errors uncovered per review hour
- Scheduled vs. Actual milestone dates
- Number of changes and their characteristics
- Distribution of effort on SE tasks

Product Metrics

Product Metrics

- Product metrics are measures of the software product at any stage of its development, from requirements to installed system.
- Product metrics may measure:
  - the complexity of the software design
  - the size of the final program
  - the number of pages of documentation produced

Software Measurements

Software Measurements

- A software metric is a measure of software characteristics which are measurable or countable.
- Software metrics are valuable for many reasons, including measuring software performance, planning work items, measuring productivity, and many other uses.

Types of Measures

(1) Direct Measures

(2) Indirect Measures

(1) Direct measures of SE process include cost and effort.

Direct measures of product include LOC (Lines of code) produced, execution speed, memory size and defects reported over some set period of time.

(2) Indirect measures of product include **functionality, quality, complexity, efficiency, reliability, maintainability.**

EXAMPLE

- 2 different project teams are working to record errors in a software process
- **Team A – Finds 342 errors** during software process before release
- **Team B- Finds 184 errors**

Which team do you think is more effective in finding errors?

Normalization for Metrics

- To answer this we need to know the size & complexity of the projects.
- But if we normalize the measures, it is possible to compare the two
- For normalization we have 2 ways-
  - **Size-Oriented Metrics**
  - **Function Oriented Metrics**

Size-Oriented Metrics

Size-oriented Metrics

- Based on the size of the software that has been produced.

For example: choose LOC (Line of Code) as normalization value.

Size Oriented Metrics

Project	LOC	FP	Effort (P/M)	R(000)	Doc(page)	Errors	Defects	People
alpha	12100	189	24	168	365	134	29	3
beta	27200	388	62	440	1224	321	86	5
gamma	20200	631	43	314	1050	256	64	6

- LOC = Line of Code
- FP = Function points
- Efforts per/month
- R(000) = Project cost
- Doc (page) = Document size in page

Size-oriented Metrics

Set of simple size-oriented metrics:

- Errors per KLOC
- Defects per KLOC
- $ per LOC
- Page of documentation per KLOC
- LOC per person-month,
- Errors per person-month
- $ per page of documentation.

Size-oriented Metrics

Advantages :

- LOC can be easily counted
- Many software estimation models use LOC or KLOC as input.

Disadvantages :

- LOC measures are language dependent, programmer dependent

Function-Oriented Metrics

Function-Oriented Metrics

- Use a measure of the **functionality** delivered by the application as a normalization value.
- Functionality can not be measured directly, it must be derived indirectly using other direct measures.
- A measure called the **function point.**
- Function points are derived using an empirical relationship based on countable (direct) measures of software's information domain and assessments of software complexity.

Metrics for Function Point

- errors per FP (A **mistake** in coding is called **Error)**
- defects per FP **(Errors** found by tester is called **Defect)**
- $ per FP
- pages of documentation per FP
- FP per person-month

Function Point

Advantages:

1. language independent
2. based on data known early in project
3. good for estimation

Disadvantages:

1. calculation complexity
2. subjective assessments

Information Criteria for Webapp Project

- Number of Static Web pages
- Number of dynamic web pages
- Number of internal Link pages
- Number of Static content objects
- Number of dynamic content objects
- Number of executable functions

Customization Index

$$C=N_{dp}/N_{sp}+N_{dp}$$

N_{dp} =number of dynamic web pages

N_{sp}=number of static web pages

Metrics for Webapp Project

- Efforts expended
- Errors and defects uncovered per page
- Documentation per page

Metrics for Software Quality

METRICS FOR SOFTWARE QUALITY

- Must **use technical measures to evaluate quality** in objective, rather than subjective ways.
- Must evaluate quality as the project progresses.
- The primary thrust is to **measure errors and defects**
- metrics provide indication of the effectiveness software quality assurance and control activities.

Note: Faults:

Errors: Faults found by the practitioners during software development

Defects: Faults found by the customers after release

Metrics for Software Quality

- Correctness
- Maintainability
- Integrity
- Usability

Measuring Quality

(1) Correctness: A system or software must function correctly.

- Correctness can be defined as the degree to which software performs its specified function.
- defects per KLOC

(2) Maintainability: Maintainability can be defined as the ease with which a software product can be modified to correct errors(correct), to meet new requirements(enhance), to adapt to the changed environment(adapt).

Measuring Quality

Maintainability can be measured by Time/cost.

- **Time-oriented metrics:** Mean-time-to-change (MTTC)

  time taken to analyze change request, design modifications, implement changes, testing, and distribute changes to all users.

- **Cost-oriented metrics**: – cost to correct defects encountered.

Measuring Quality

(3) Integrity: ability to withstand attacks

- Software integrity can be defined as the degree to which unauthorized access to the components of software (program, data, and documents) can be controlled.
  - Threat: the probability that an attack of a specific type will occur within a given time.
  - Security: the probability that the attack of a specific type will be repelled(resist).

Integrity = sum [(1 – threat)x(1 – security)]

Measuring Quality

(4) Usability: Software, which is easy to understand and easy to use is always preferred by the user.

- Usability can be defined as the capability of the software to be understood, learned, and used under specified conditions.

Defect Removal Efficiency

- A quality metric that provides benefit at both the project and process level.
- DRE is a measure of filtering ability of quality assurance and control activities as they applied throughout all process framework activities.

Defect Removal Efficiency

DRE = (errors) / (errors + defects)

Where,

errors = problems found before release

defects = problems found after release

The ideal value for DRE is 1, no defects found.

THANK YOU!!!!