

Lecture : 05

Software Process Measurement

Why do we measure?

IF YOU CAN'T MEASURE IT, YOU CAN'T MANAGE IT (Tom DeMarco, 1982)

1. Determine quality of piece of software or documentation
2. Determine the quality work of people such software engineers, programmers, database admin, and most importantly MANAGERS
3. Improve quality of a product/project/ process
4. To uncover problem areas before they go critical
5. To close the gap of any problems (E.g training)
6. To help justify requests for new tools or additional training
7. Track potential risks
8. To adjust workflow or tasks

Definitions

MEASUREMENT: is the act of obtaining a measure.

MEASURE: provides a quantitative indication of the size of some product or process attribute e.g. number of errors

METRIC: is a quantitative measure of the degree to which a system, component or process possesses a given attribute e.g. number of errors found per person hours expected.

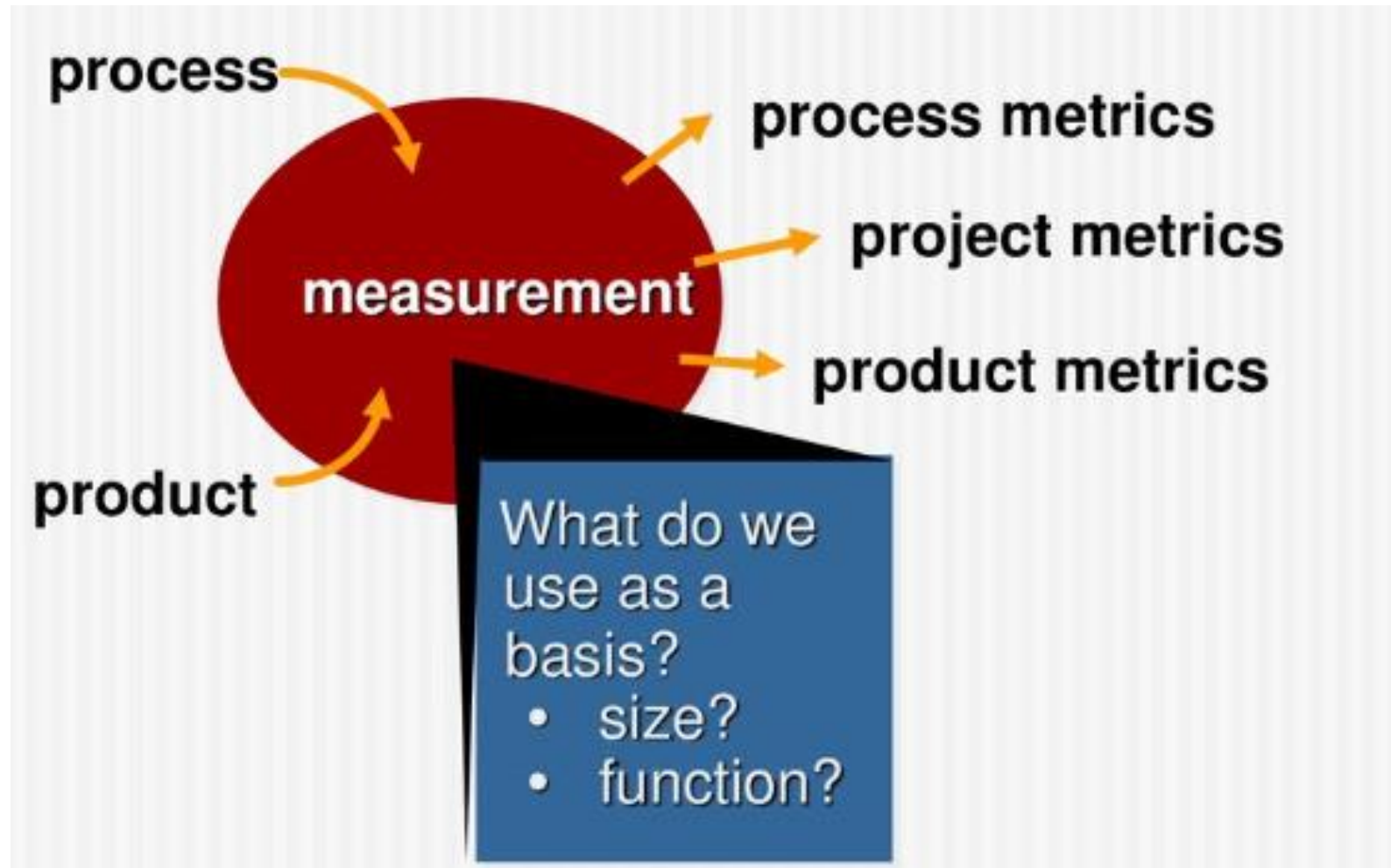
What to Measure

Process: Measure the efficacy of processes. What works.. what doesn't

Project: Assess the status of the project. Track risks Identify problem areas. Adjust workflows

Product: Measure predefined product attributes

A Good Manager Measures



What to measure (Process)

- Code quality

- Programmer productivity

- Software engineer productivity

 - Requirements,
design,
testing
and all other tasks done by software engineers

- Software

 - Maintainability

 - Usability

 - And all other quality factors

- Management

 - Cost estimation

 - Schedule estimation, Duration, time

 - Staffing

Process Metrics Vs Project Metrics

Process metrics are measures of the software development process, such as

Overall development time

Type of methodology used

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

Their intent is to provide indicators that lead to long-term software process improvement.

Project Metrics are the measures of Software Project and are used to monitor and control the project.

They usually show how project manager can estimate schedule and cost

They enable a software project manager to:

- Minimize the development time by making the adjustments necessary to avoid delays and potential problems and risks.
- Assess product cost on an ongoing basis & modify the technical approach to improve cost estimation.

Process Metrics Vs Product Metrics

Process metrics are measures of the software development process, such as

- Overall development time

- Type of methodology used

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

Their intent is to provide indicators that lead to long-term software process improvement.

Product metrics are measures of the software product at any stage of its development, from requirements to installed system. Product metrics may measure:

- How easy is the software to use

- How easy is the user to maintain

- The quality of software documentation

- And more ..

.

Integrating Metrics within the Software Process

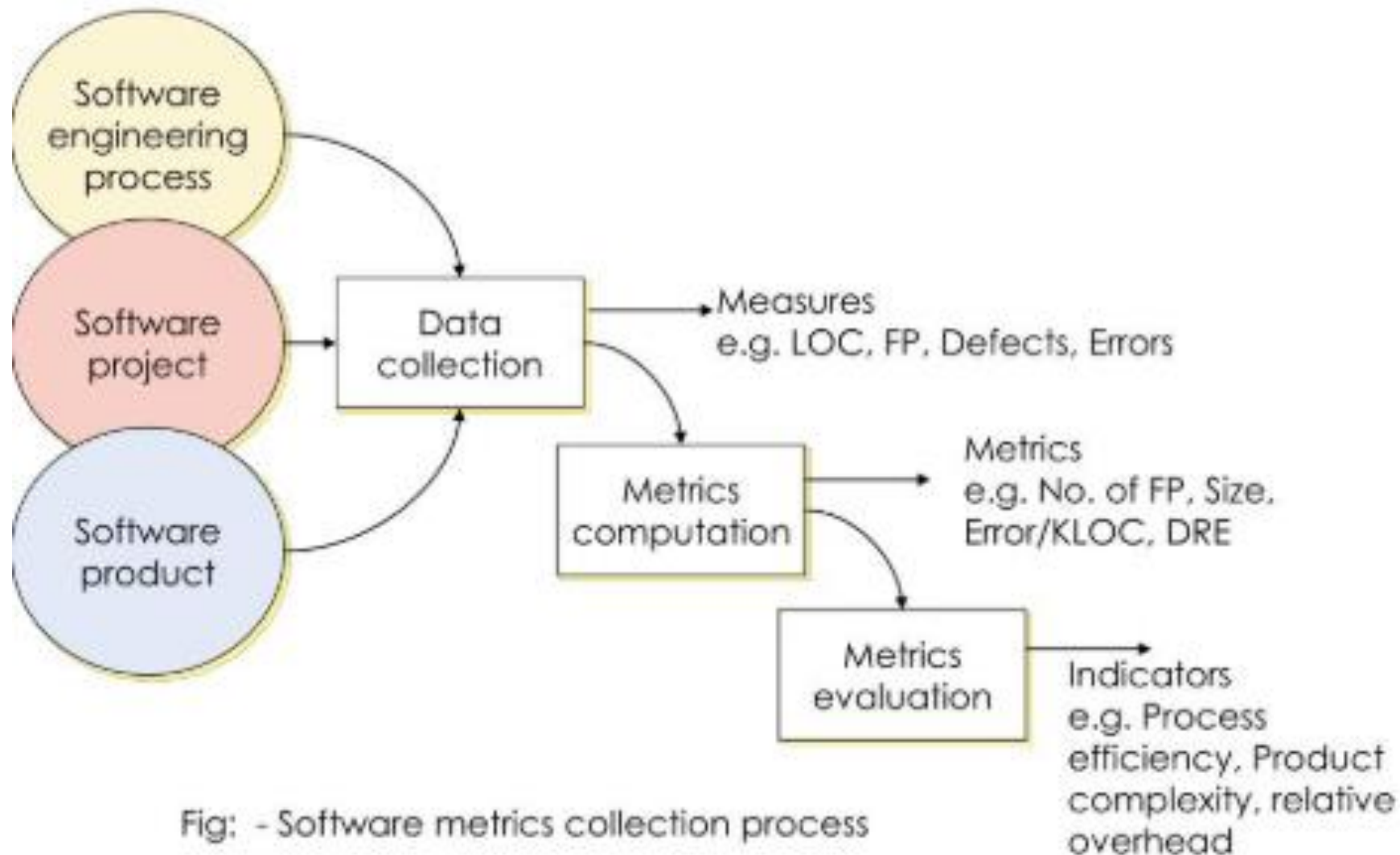


Fig: - Software metrics collection process

Process Measurement

We measure the efficacy of the software process indirectly

That is we drive a set of metrics based on the outcomes that can be derived from the process

Outcomes include

measures of errors uncovered before release of the software

defects delivered to and reported by the end users

work products delivered (productivity)

human effort expended

calendar time expended

schedule conformance

We also derive process metrics by measuring the characteristics of specific software engineering tasks.

Process Metrics

Quality-Related:

Focus on quality of work products and deliverables

Productivity-Related:

Production of work-products related to effort expended

Statistical SQA data:

Error categorization & analysis

Defect Removal Efficiency

Propagation of errors from process activity to activity

Reuse Data

The number of components produced and their degree of reusability

Software Metrics etiquettes

- ☐ Do not use metrics to appraise people
- ☐ Provide regular feedback to the individuals and teams who collect measure and metrics
- ☐ Use common sense and organizational sensitivity when analyzing and interpreting software metrics data.
- ☐ Don't obsess on a single metric to the exclusion of other important metrics
- ☐ Metrics area that indicate a problem area should not be considered negative. These data are merely an indicator for process improvement.
- ☐ Never use metrics to threaten individuals or teams
- ☐ Work with practitioners and teams to set clear goals and metrics that will be used to achieve them

Statistical Software Process Improvement

ERROR

Some flaw in the software engineering work product that is uncovered before the s/w is delivered to the end-user

DEFECT:

A flaw that is uncovered after the delivery to the end user

TASK:

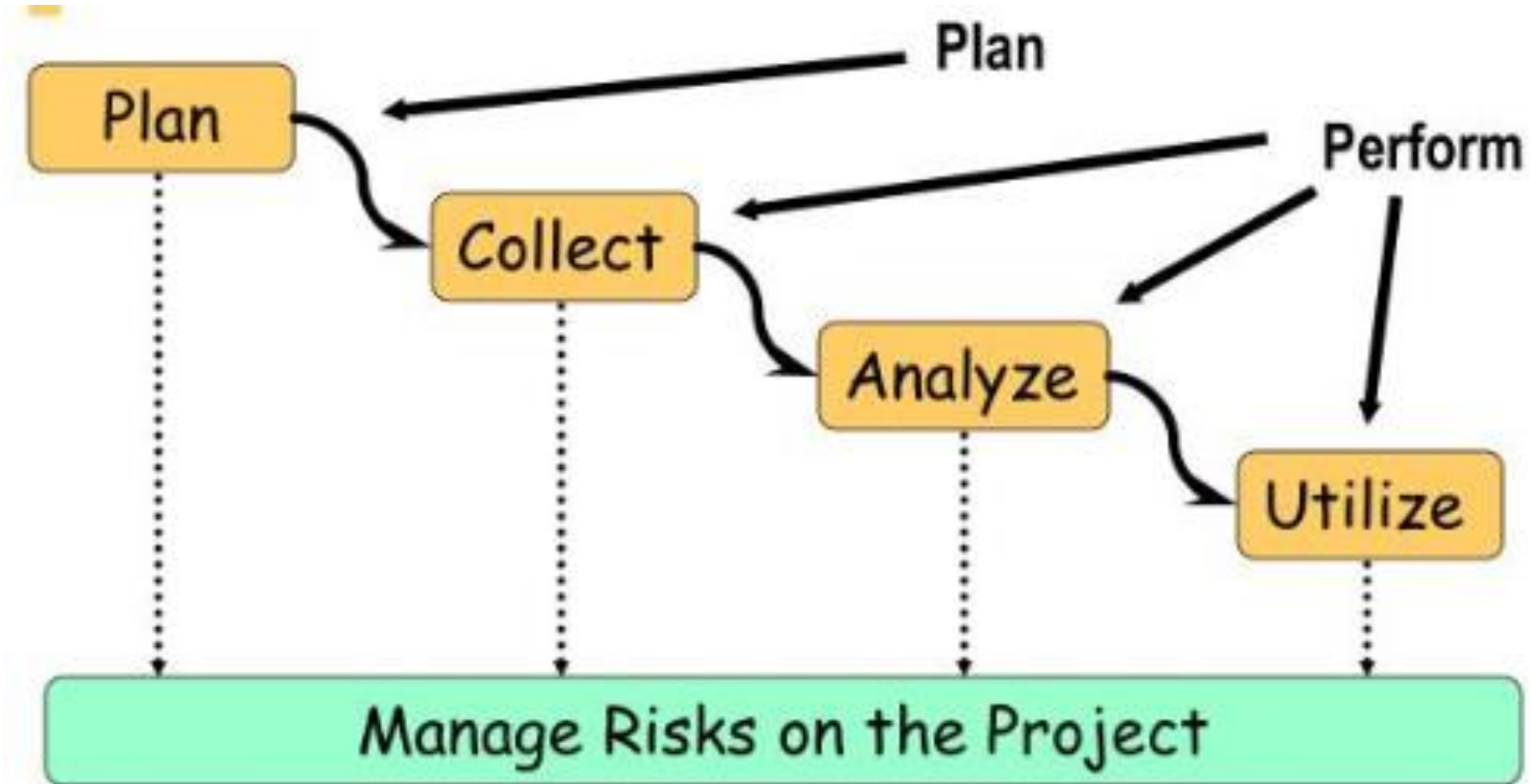
Identify the metrics parameters in a Web Engineering project

TASK:

Identify the metrics parameters in a Web Engineering project

- Number of static Web pages
- Number of dynamic Web pages
- Number of internal page links
- Number of persistent data objects
- Number of external systems interfaced
- Number of static content objects
- Number of dynamic content objects
- Number of executable functions

Core Measurement Process



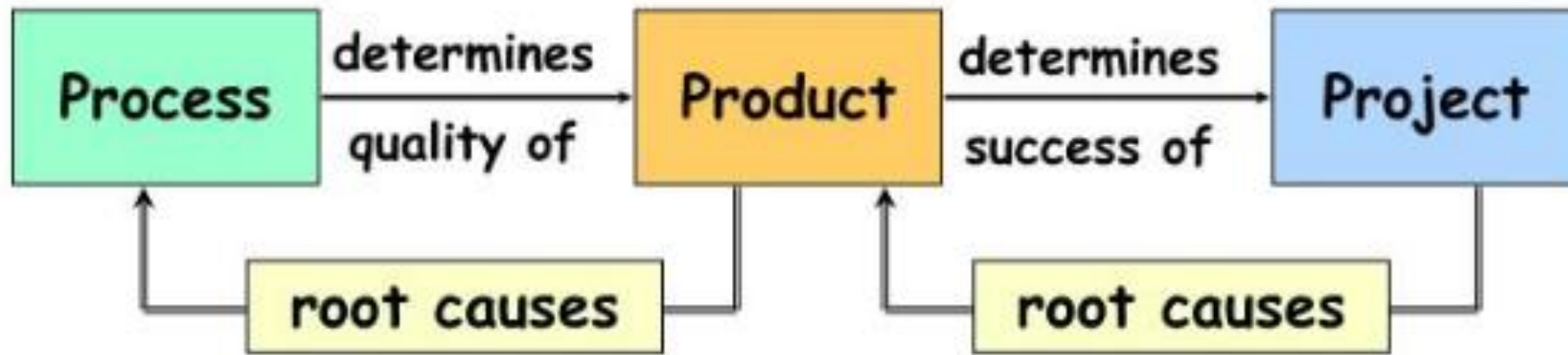
Who Cares about what?

Managers – *Project Measures*

Developers – *Product Measures*

Both should care about the *Process measures*.

What should we measure?



Process Measure

- how effective is the process?
- how well are we following the process?
- risk monitoring

Product Measure

- performance and quality
- How well is the product meeting its requirements

Project Measure

- used to monitor the state of the project
- how are we doing relative to cost, schedule and staffing)