Chapter 3
Software Project Management Metrics

3.1. SW Metrics

a) <u>What is a Metric</u>?

- Measure : is a quantitative indication of the extent, amount, dimension, capacity, or size of some attributes of a product or process.
- Measurement : is the act of determining a measure.
- Metric : is a quantitative measure of the degree to which a system, component, or process possesses a given attribute … A software metric relates the individual measures in some way … (IEEE Standard Glossary of SE terms)
- Indicator : is a metric or combination of metrics that provides insight into the software process, a software project, or the product itself.
- Metrics assign values to quantitative factors.
- Metrics facilitate project planning, scheduling, and improving the SDLC process and product quality.
- A metric enables you to measure the quality of a factor.
- You can measure and quantify a factor only if the factor has numeric values

3.1. SW Metrics

a) What is a Metric?

* What do we measure?

- Process
- Project
- Product
- Design
- Maintenance

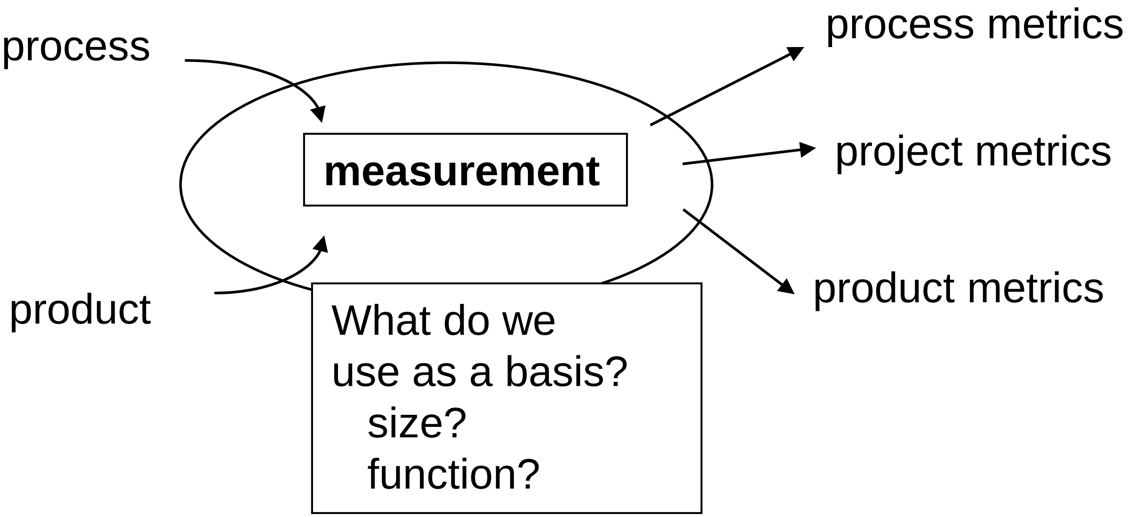
* Why do we measure?

- To characterize
- To evaluate
- To predict
- To improve

3.1. SW Metrics

a) What is a Metric?

* A Good Manager Measures



What do we use as a basis?
size?
function?

3.1. SW Metrics

b) Characteristics of a Metric

- Goal-oriented approach:

+ A metric should be goal-oriented.

For example, the goal of metric can be to reduce the expenses incurred in different phases of a SW project or to reduce the time spent on it.

+ The goal of the metric is used to define a baseline value. (A baseline value is a specification that is formally reviewed and agreed upon)

3.1. SW Metrics

b) Characteristics of a Metric

- Measurable :

+ Measurability denotes that a metric can be used to measure a SW entity to a high degree of accuracy, if not completely accurately.

+ Measurability of a metric ensure consistent results for all processes in a project.

- Analyzable: It should be suitable for analysis

- Programming language-independent: A metric should also be independent of the programming language used for SW development

- Timely: this means that the data to produce results using the metric should always be available when it is needed.

3.1. SW Metrics

c) <u>Steps to Create a Metric</u>

- Define the goal of the metric

- Identify the requirements of the metric: Human resources, data collection techniques, and methodologies used to process the data.

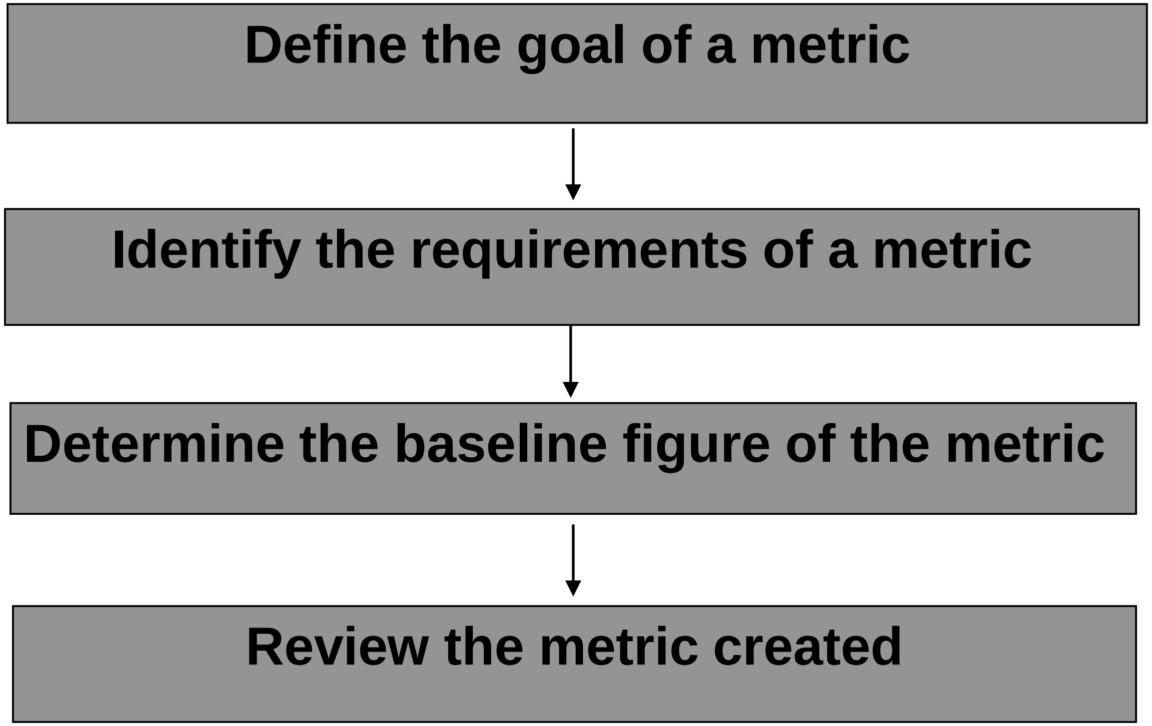
- Identify the organizational baseline value for the metric

+ A baseline value is an average value that an organization may have identified based on prior experience.

- Review the metric for its usability

3.1. SW Metrics

c) Steps to Create a Metric



Define the goal of a metric

Identify the requirements of a metric

Determine the baseline figure of the metric

Review the metric created

3.2. Types of Software Metrics

a) Design Metrics

- Design metrics help measure the design and architecture of a SW project.
- These are used to record design issues, which correspond to the requirements document.
- Enable you to decide how much you have deviated from the requirements of the project (Lesser the deviation, fewer the number of defects).
- Design metric is the measure of the complexity of a SW design.
  - Complexity can be measured for the structure, data component, and the interface design of a SW program.
  - Complexity can affect the size, testability, and effort spent on developing and testing the modules of a project.

3.2. Types of Software Metrics

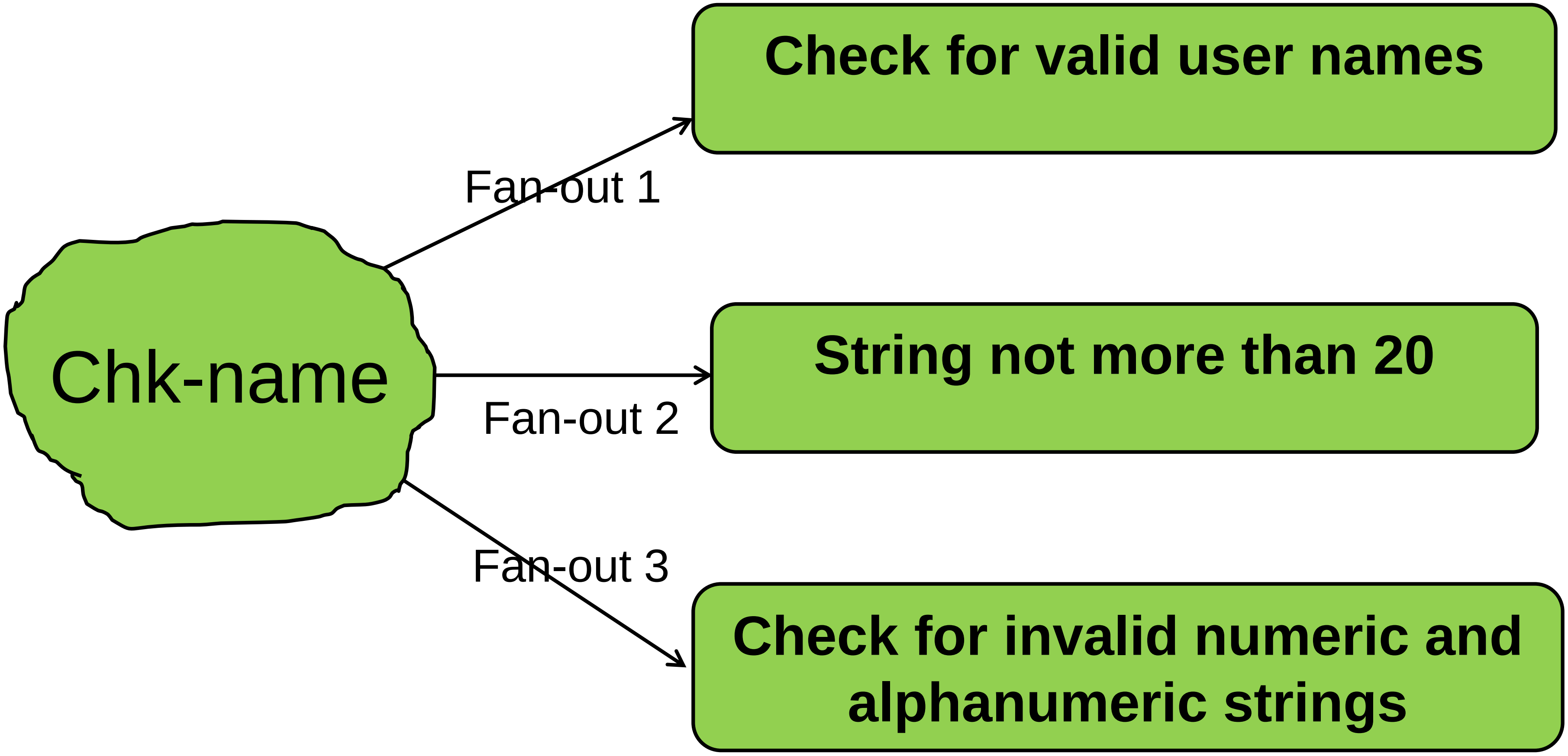
a) Design Metrics

* **Architecture Design Metrics**

There are three types of architecture design metrics:

- Structural complexity (the number of fan-out modules)
- Data complexity (the complexity of the internal interface of a SW program)
- System complexity (structural complexity + data complexity)

3.2. Types of Software Metrics(➡)



Chk-name

Fan-out 1

Check for valid user names

Fan-out 2

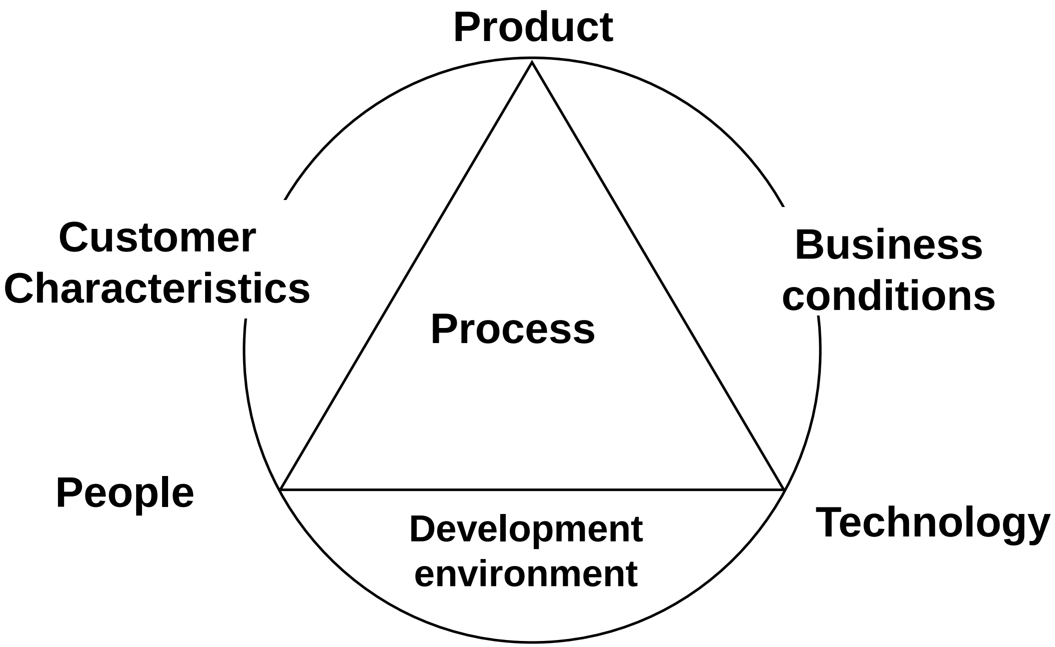
String not more than 20

Fan-out 3

Check for invalid numeric and alphanumeric strings

3.2. Types of Software Metrics

b) Process Metrics



Measure process indirectly

3.2. Types of Software Metrics

b) Process Metrics

- People: មានឥទ្ធិពលយ៉ាងខ្លាំងទៅលើ process ពិសេសអ្នកដែលមាន ជំនាញខ្ពស់ៗ
- Product: មានឥទ្ធិពលយ៉ាងសំខាន់ទៅលើគុណភាព និង team performance.
- Technology (ដូចជា SE method): ក៏មានឥទ្ធិពលយ៉ាងខ្លាំងទៅលើ process ដែរ

ក្រៅពីនេះនៅមាន៖

- Development environment: CASE Tool (Computer-Aided SE)
- Business conditions: deadlines, business rules
- Customer characteristics: ease of communication

3.2. Types of Software Metrics

c) Project Metrics

- Project metrics are specific to the actual execution of a project.
- They can help you avoid project delays.
- They measure the effectiveness of the important factors for a project.
- They are implemented in every phase of the SDLC.
- They comprise effort, productivity in FP (function point), cost, size, defects, and testing.

3.2. Types of Software Metrics(➡)

c) Project Metrics

1) Effort Metrics

- The effort metric enable you to determine the amount of effort to complete a project.
- It is calculated in person-months (man-months). A person-month is the amount of effort required to complete work in a month.
- The planned value for effort is known as the baseline value

	Metrics	**Analysis**	**Design**	**Coding**	**Testing**	**Total**
Effort	Planned effort (person-months)	5	12	25	10	52
	Actual effort (person-months)	8	14	35	8	65
	Percentage of increase in effort	60	16.5	40	-20	25

3.2. Types of Software Metrics

c) <u>Project Metrics</u>

2) <u>Productivity Metrics</u>

- The effort in performing an FP of work in an hour is called productivity.
- To calculate productivity, you first determine the total amount of FP for the SW project. Then, distribute the total FP among the phases in the SDLC.
- To calculate accurate productivity for a particular phase, you need two types of data:
  - Actual effort expressed in person hours
  - Actual size of the project in FP

Metrics	**Analysis**	**Design**	**Coding**	**Testing**	**Total**
Productivity in hours/FP					
Planned	1.2	1.6	2.8	2.4	8.0
Actual	1.0	1.4	3.0	2.5	7.9
Difference(+/-)	0.2	0.2	-0.2	-0.1	0.1

3.2. Types of Software Metrics

c) Project Metrics

3) Cost Metric

- A cost metric measures the planned versus actual expense incurred on a project.
- An important component of the cost metric is the cost of resources (human resources and material resources)

Metric		***Analysis***	***Design***	***Coding***	***Testing***	***Total***
Cost in USD		40,000	100,000	200,000	100,000	440,000
Resources	Planned					
	Actual	55,000	130,000	300,000	250,000	735,000
Communication	Planned	10,000	7,000	1,000	1,000	19,000
	Actual	20,000	4,500	700	500	25,700
	Total planned	50,000	107,000	201,000	101,000	459,000
	Total actual	75,000	134,500	300,700	250,500	760,700
	Total deviation in percentage (actual-planned/planned)*100	50	25	50	148	66

3.2. Types of Software Metrics

c) <u>Project Metrics</u>

4) Size Metrics

- Size metric is calculated for an entire project and not for any particular phase of a project.
- The size of a project may vary with respect to the changes required by the customer.
- The size metric directly affects effort, testing, and productivity metrics of a project.

  There are two main components of the size metric:

  - Change in the required effort due to change in project size
  - Percentage growth in project size

3.2. Types of Software Metrics

c) <u>Project Metrics</u>

4) <u>Size Metrics</u>

Metric		***Total***
Size in FP		
	Size Planned	100
	Size Actual	120
	Change in size	+20
	Growth of size in %	20

3.2. Types of Software Metrics

c) Project Metrics

5) Defects Metrics

- The number of defects defines the quality of a project.
- If the number of defects is large, the quality of the product is inferior. In contrast, if the number of defects is small, the quality of the product is superior.
- Defects can occur during any phase of the SDLC.
- Defects not detected in a particular phase continue to be present during the subsequent phases in the SDLC.

3.2. Types of Software Metrics

c) Project Metrics

5) Defects Metrics

* Severe defects critically affect the functionality of a product.

* Major defects logically affect the functionality of a product.

* A minor defect is a slight defect that may act as an irritant for users but does not disturb the functionality of the application.

Metric		***Analysis***	***Design***	***Coding***	***Testing***	***Total***
Defects per KLOC						
Severe	Planned (less than equal to)	0	0	0	0	0
	Actual	5	3	15	8	31
Major	Planned (less than equal to)	5	2	5	10	22
	Actual	3	6	12	18	39
Minor	Planned (less than equal to)	20	15	15	20	70
	Actual	12	10	12	25	59

3.2. Types of Software Metrics

c) Project Metrics

6) Testing Metrics

- The testing metric is used to measure the number of test cases required to test SW.
- Test case is a specification that needs to be executed to test a particular module in a SW program.
- There are separate test cases for integration testing and unit testing:
  - Integration testing refers to the overall black box testing of the entire SW project.
  - In contrast, unit testing refers to testing individual modules of a SW project

Metric(s)		***Coding***
Testing	Test cases per FP	
	Planned	5
	Actual	4

3.2. Types of Software Metrics

d) Product Metrics

- Product metrics measure the quality and conformance to the requirements of the deliverables of a phase.
- They also measure the timeliness in the delivery of project deliverables.

	Deliverable name	***Effectiveness(%)***	***Conformance to requirements (%)***
Source code	Planned	100	100
	Actual	72	90

3.2. Types of Software Metrics

d) <u>Product Metrics</u>

Two factors are measured:

- Effectiveness

Formula to calculate effectiveness of a deliverable:

Effectiveness = (Quantity in % * Quality in %) / 100

Ex: 80% of source code is completed and 90% of the quality goal are achieved in developing that source code.

Effectiveness = (80 * 90) / 100 = 72

- Conformance to requirements

3.2. Types of Software Metrics

e) **Maintenance Metrics**

- Maintenance metrics are a set of metrics similar to the metrics for a development project.
- They are used to measure the cost, effort, productivity, and defects associated with a maintenance project.
- You use maintenance metrics to track the number of required changes implemented in a maintenance project.
- There are two types of measurement performed for maintenance project:
  - Extent of change required
  - Type of maintenance requested by the customer

3.2. Types of Software Metrics

e) **Maintenance Metrics**

No. of modules remaining unmodified	6
No. of modules added to a released SW program	3
No. of modules modified in a released SW program	10
No. of modules deleted from a released SW program	2
Total	21

- There are two metrics used to measure maintenance activities in an organization:
  - Corrective
  - Upgrades

Type of maintenance	***Planned in %***	***Actual in %***
Corrective	17	70
Upgrades	83	30

3.3. Direct and Indirect Measures

- Direct Measures
  - Cost and effort applied
  - Lines of Code
  - Execution speed
  - Memory size
  - Defects per unit time

are relative easy to collect

- Indirect Measures
  - Quality
  - Complexity
  - Efficiency
  - Functionality
  - Reliability
  - Other “–abilities”

are more difficult to assess

3.2. Types of Software Metrics(⬅)

* **ជាទូទៅ**

បើការវាស់វែងតាមបែប Metrics ទៅលើដំណាក់កាលណាមួយនៃ SDLC ឃើញ៖

1- AV(Actual Value)> PV(Planned Value) ឬ BV(Baseline Value)

⇒ អត់ល្អ

⇒ យើងត្រូវត្រួតពិនិត្យឡើងវិញទៅលើការងារគ្រប់គ្រង (Management)។

⇒ កែលម្អ

2- AV ≤ PV ⇒ ល្អ

3.2. Types of Software Metrics(⇦)

* Fan-out Module

Fan-out module ជា module កូនទាំងឡាយណាដែលត្រូវបាន module មេហៅ(Access) ទៅកាន់ៗ