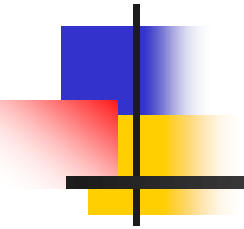
SOEN 384
Management, Measurement and Quality Control

http://users.encs.concordia.ca/~s384_2/

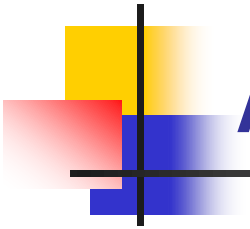


Instructor: Dr. Olga Ormandjieva

Lecture 4:

Perform the Measurement Process

1. *ISO 15939*
2. *SWEBOK chapter 7*

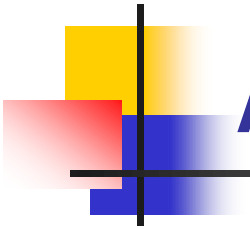


Agenda

- Homework 1 (GQM)
- Review
- Plan Measurement Process. Examples
- Perform Measurement Process. Examples
- Homework 2 (LOC)
- Next?

Homework 1 (1%, individual)

- Propose and document a *GQM* model for the following goal: **Improve Student Performance in soen384 course**
- Follow the GQM steps seen in tutorial 1
- Submit your GQM electronically through the EAS as **theory assignment 1** by September 15 midnight (individual).
- WRITE YOU NAME and ID ON THE FIRST PAGE OF YOUR SUBMISSION



Agenda

- Homework 1 (GQM)
- Review
- Plan Measurement Process. Examples
- Perform Measurement Process. Examples
- Homework 2 (LOC)
- Next?

Project Management Process

(Managing the Development of Software-Intensive Systems)

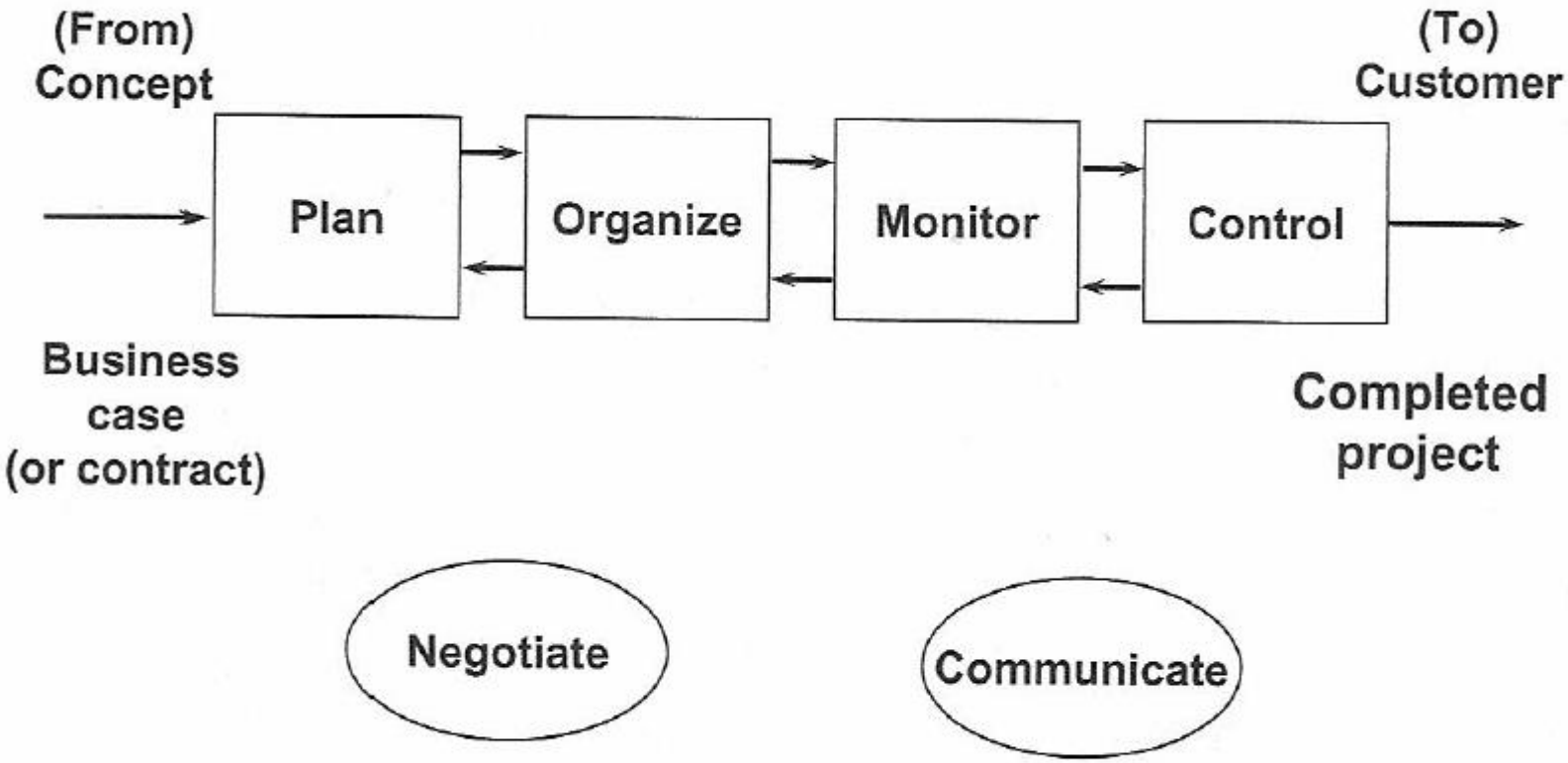
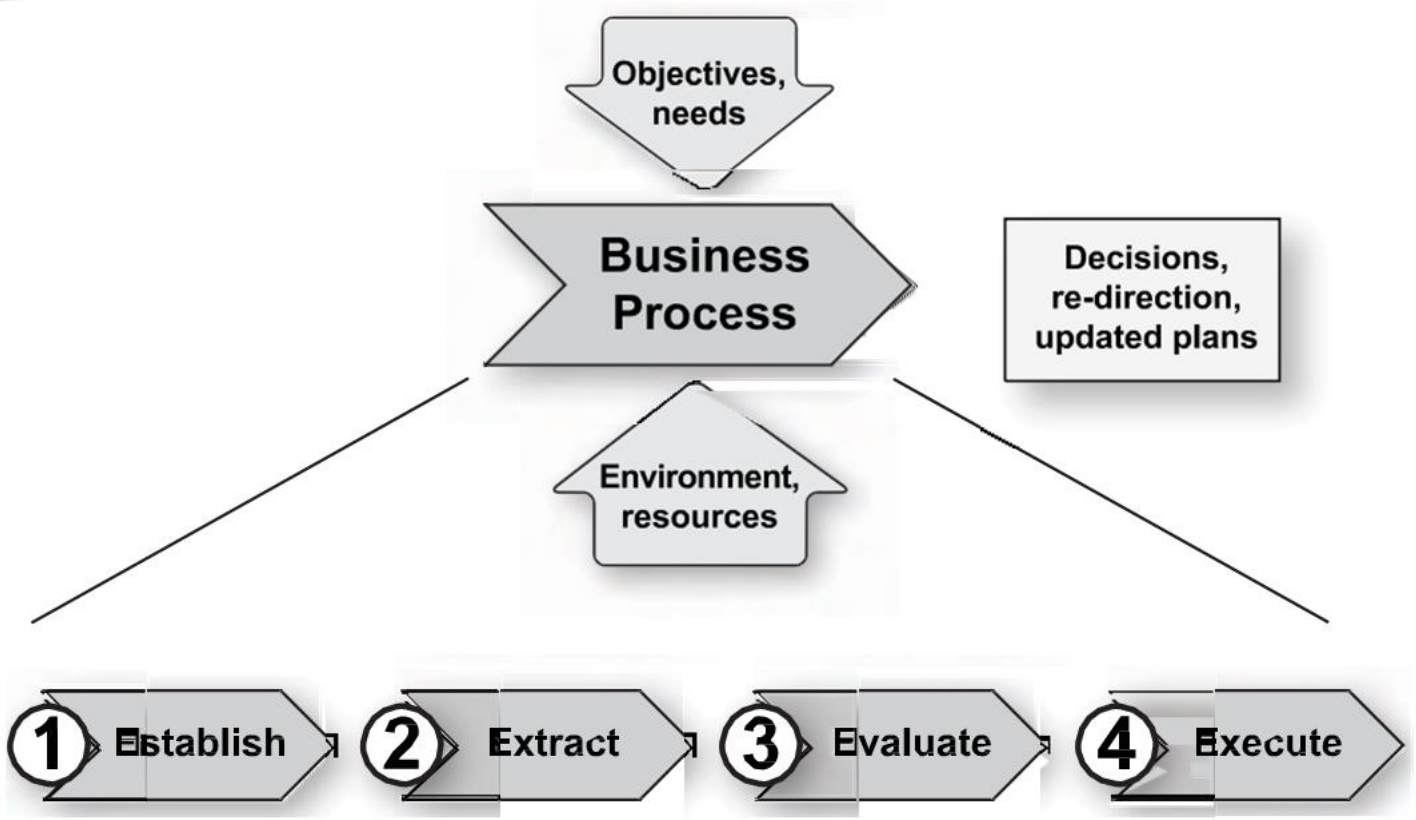


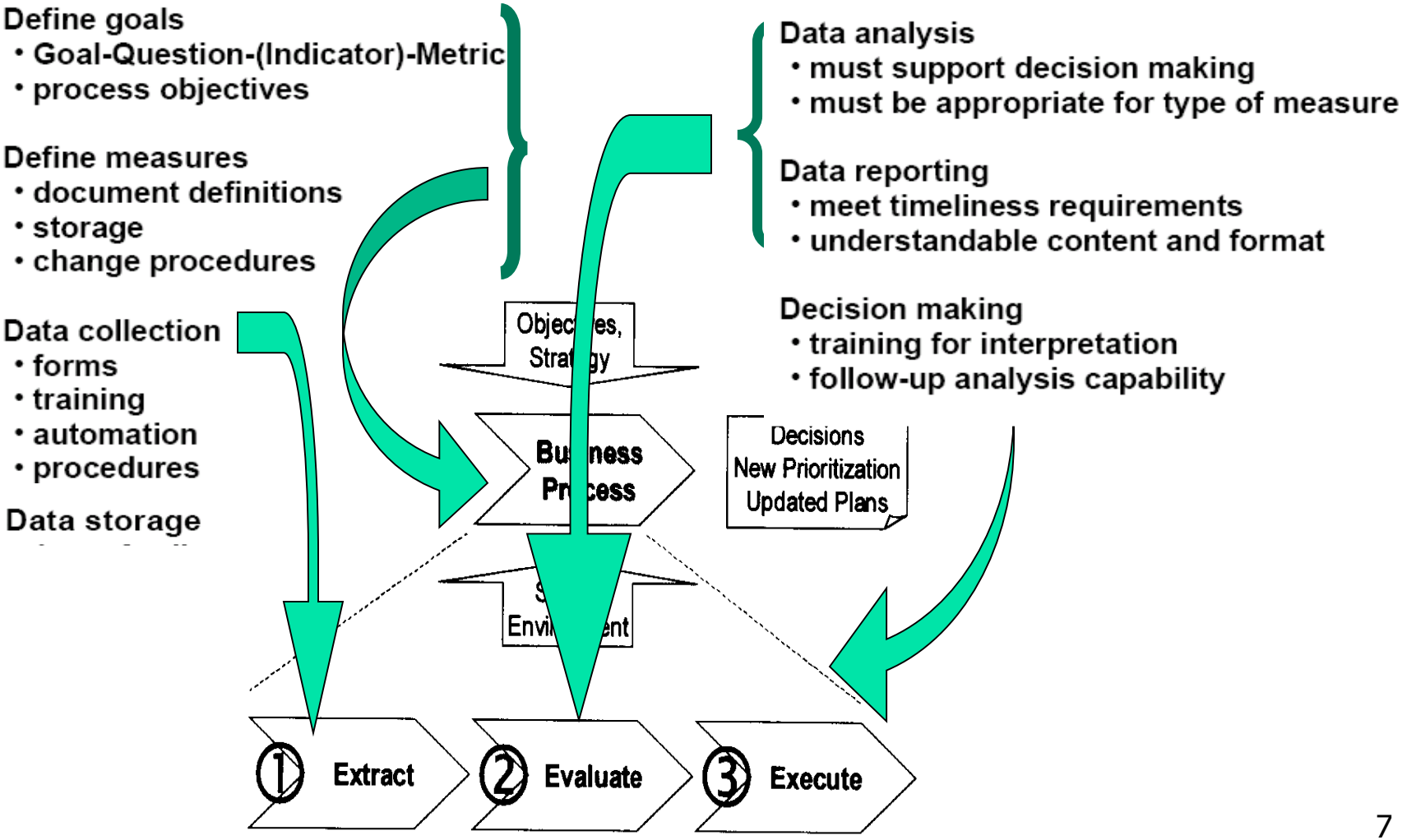
Figure 1.1 Project management processes.

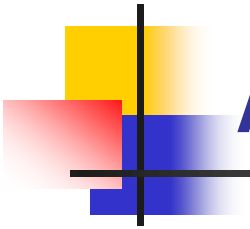
A Generic Measurement Feedback Loop

Software Project and Process Measurement, Dr. Christof Ebert
Vector Consulting Services



Software Measurement Key Tasks





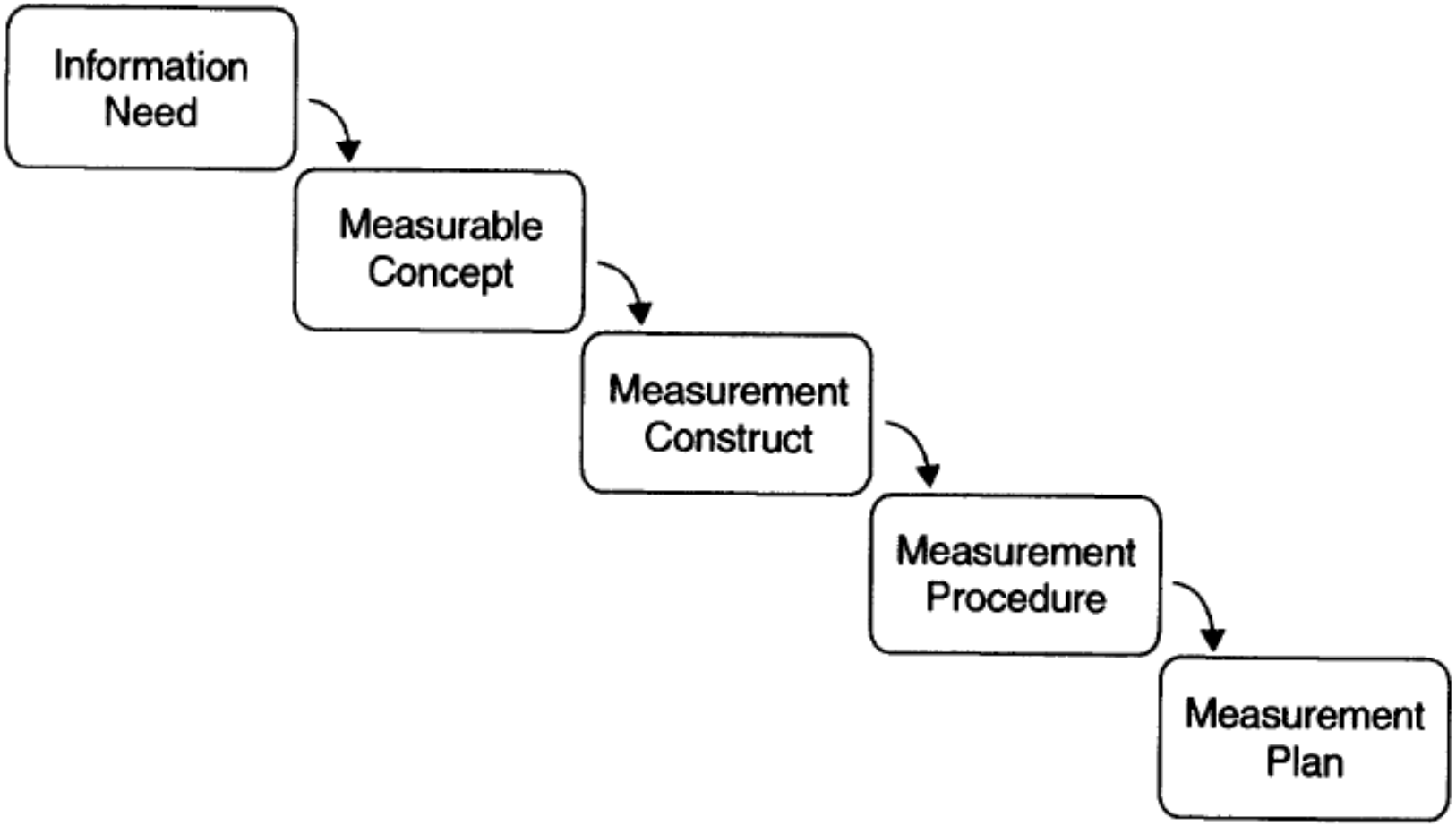
Agenda

- Homework 1 (GQM)
- Review
- Plan Measurement Process. Examples
- Perform Measurement Process. Examples
- Homework 2 (LOC)
- Next?

Plan the measurement process (Chapter 7, section 6.2, SWEBOK)

- Characterize the organizational unit
  - Application domain, org. structure and processes, technology…
- Identify information needs
  - Goals, problems, constraints
- Select measures
- Define data collection, analysis, and reporting procedures
  - Storage, scheduled, verification, stats, reporting
- Provide resources for measurement tasks

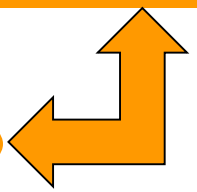
Measurement Information Model: foundation for consistent measurement terminology (ISO 15939)



Information needs: Common Categories for project management

- Schedule and progress
- Resources and cost
- Product size and stability
- Product quality
- Process performance
- Technology effectiveness
- Customer satisfaction

Productivity analysis



Information need and

Information Need	Estimate productivity of future project
Measurable Concept	Project productivity

- **Goal:** defines the information need
  - A decision maker is concerned with improving the project performance by allocating the right budget and associated resources to software tasks
- **Question:** What should be measured in order to satisfy the information need?
  - He/she may believe that **productivity** is related to the type of software task that will be performed

Productivity is the measurable concept that addresses the defined information need

From Measurable Concept to Measurement construct (GQM approach)

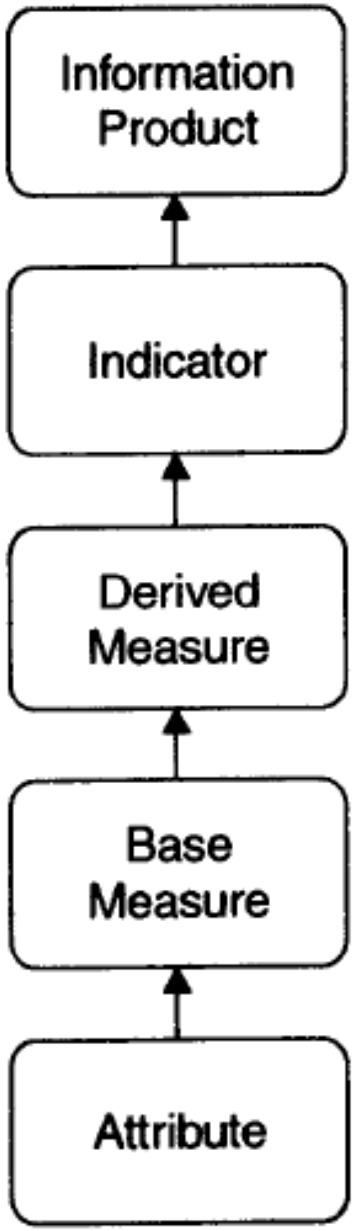
- **M. Construct:** Specifies exactly what will be measured and how the data will be combined to produce results that satisfy information need.
  - **Metrics:** amount of software produced and effort expended for each project in the measurement data store
  - ***Assumption:*** *productivity is estimated based on past performance*

Measurement construct

Measurement Construct:

- describes how the relevant software attributes are quantified and converted to indicators that provide a basis for decision making

Levels:



Relevant entities and (Measurable) attributes

- Things that can actually be measured in a software project:
  - Size, effort, number of defects, schedule

Relevant Entities	1.	Code produced by past projects
	2.	Developers' effort expended by past projects
Attributes	1.	C++ language statements (size)
	2.	Timecard entries (recording effort)

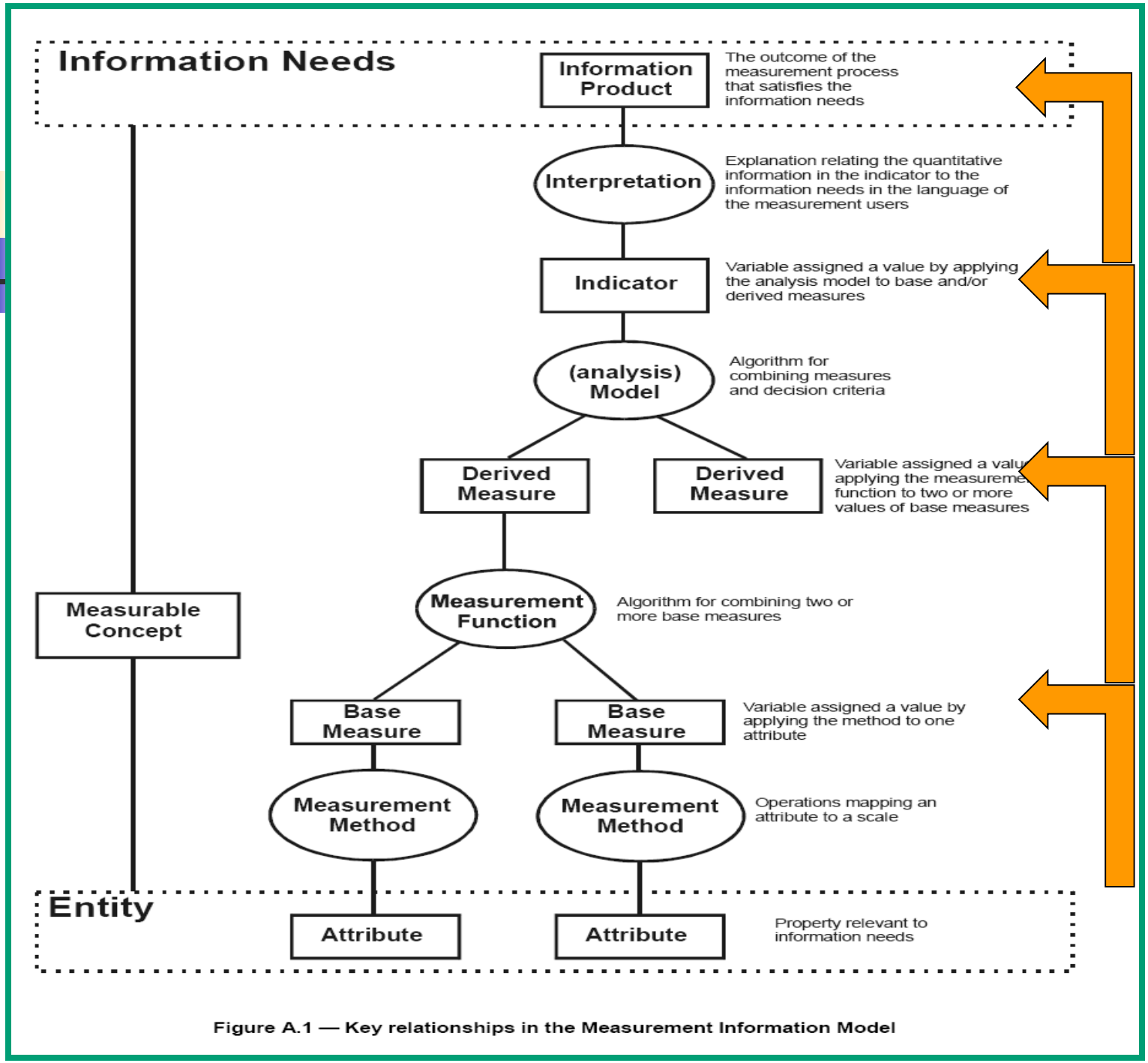


Figure A.1 — Key relationships in the Measurement Information Model

Levels of measurement construct

Relevant Entities	1. Code produced by past projects 2. Effort expended by past projects
Attributes	1. C++ language statements (in code) 2. Timecard entries (recording effort)
Base Measures	1. Project X Lines of code 2. Project X Hours of effort
Derived Measure	Project X Productivity
Indicator	Average productivity

Information product: Estimated productivity based on historical data (for similar projects).

Base Measure: measurement method, Scale, Scale Type, Unit)

Base Measures	1. Project X Lines of code	
	2. Project X Hours of effort	
Measurement Method	1. Count semicolons in Project X code	
Ty		2. Objective
Scale	1. Integers from zero to infinity	
	2. Real numbers from zero to infinity	
Type of Scale		1. Ratio
		2. Ratio
Unit of Measurement		1. Line
		2. Hour

Will be discussed in detail
(RE: representational theory of measurement)

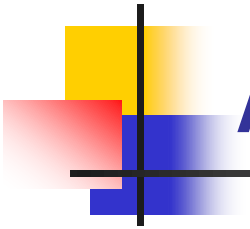
Derived measure: Measurement function

Derived Measure	Project X Productivity
Measurement Function	Divide Project X Lines of Code by Project X Hours of Effort

Indicator: Analysis model, Decision criteria

Indicator	Average productivity
Model	Compute mean and standard deviation of all project productivity values
Decision Criteria	Computed confidence limits based on the standard deviation indicate the likelihood that an actual result close to the average productivity will be achieved. Very wide confidence limits suggest a potentially large departure and the need for contingency planning to deal with this outcome.

Information Need	Estimate productivity of future project
Measurable Concept	Project productivity
Relevant Entities	1. Code produced by past projects 2. Effort expended by past projects
Attributes	1. C++ language statements (in code) 2. Timecard entries (recording effort)
Base Measures	1. Project X Lines of code 2. Project X Hours of effort
Measurement Method	1. Count semicolons in Project X code 2. Add timecard entries together for Project X
Type of Measurement Method	1. Objective 2. Objective
Scale	1. Integers from zero to infinity 2. Real numbers from zero to infinity
Type of Scale	1. Ratio 2. Ratio
Unit of Measurement	1. Line 2. Hour
Derived Measure	Project X Productivity
Measurement Function	Divide Project X Lines of Code by Project X Hours of Effort
Indicator	Average productivity
Model	Compute mean and standard deviation of all project productivity values
Decision Criteria	Computed confidence limits based on the standard deviation indicate the likelihood that an actual result close to the average productivity will be achieved. Very wide confidence limits suggest a potentially large departure and the need for contingency planning to deal with this outcome.



Agenda

- Homework 1 (GQM)
- Review
- Plan Measurement Process. Examples
- Perform Measurement Process.
- Examples
- Homework 2 (LOC)
- Next?

SWEBOK Chapter 7, 6.3 Perform the Measurement Process

- Integrate measurement tasks with relevant software processes they are measuring
- Collect data (a.k.a. Extract)
- Analyze data (a.k.a. Evaluate)
- Communicate results

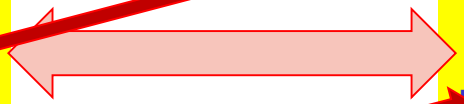
Integrating project management & measurement

Set of management processes and techniques:

- Planning
- Organizing
- Monitoring
- Controlling

To be integrated with a measurement process:

- Establish
- Extract
- Evaluate
- Execute



How to integrate them?

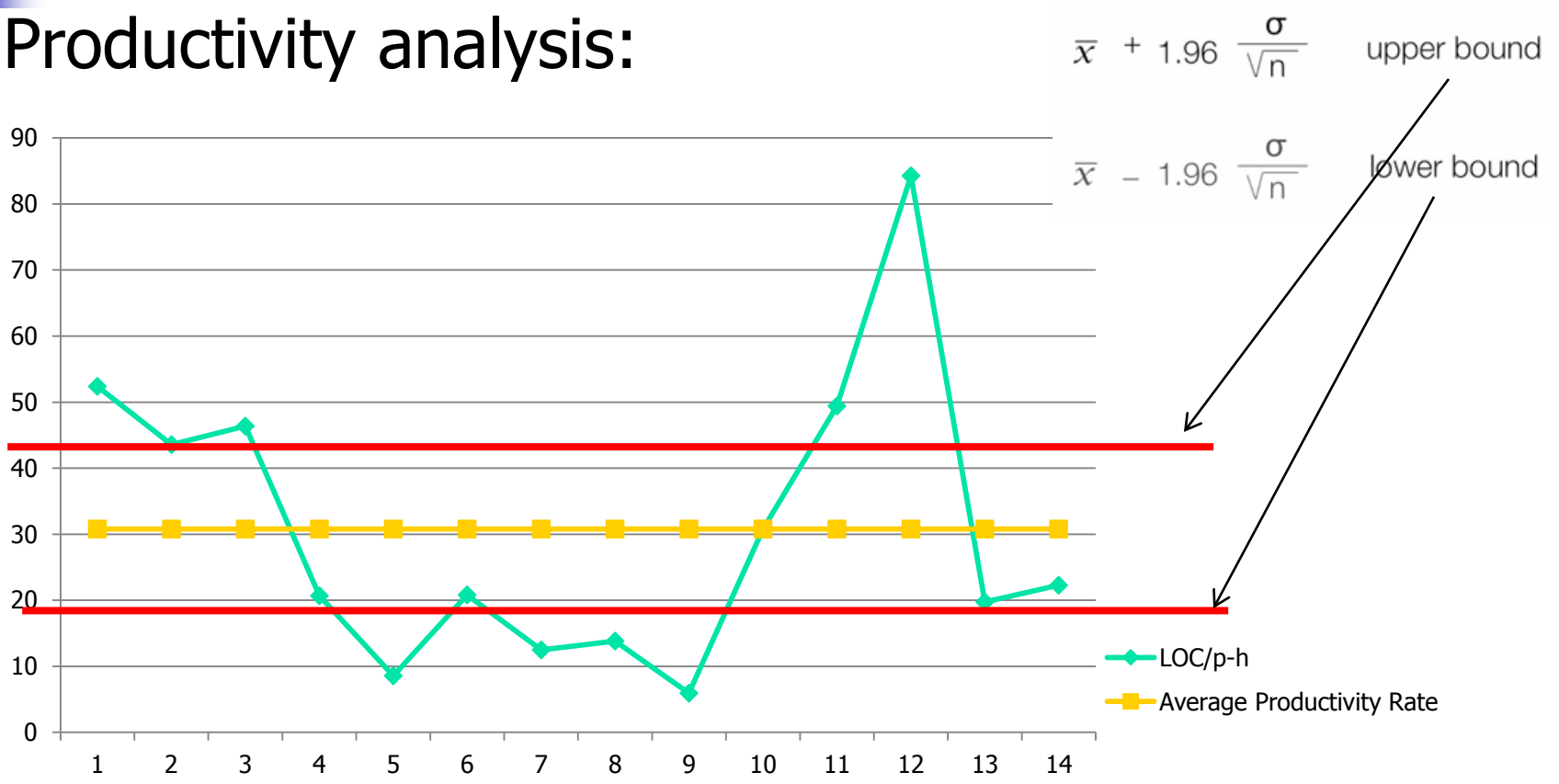
General Steps of Perform Measurement Process

- **Extract**
  - Extract information (collect measurement data) for a specific need and record it for potential further usage.
- **Evaluate**
  - Use the extracted measurement data on the spot!
  - Statistical analysis and evaluation
- **Execute**
  - Execute decis n status and objectives.
  - Manageme ecific need of the stakeh

project	average productivity
1	52
2	43
3	48
4	21
5	9
6	20.5
7	11.5
8	13.5
9	7.5
19	50
11	85
12	19.5
13	21.5

Productivity Indicator (analysis report) confidence limits

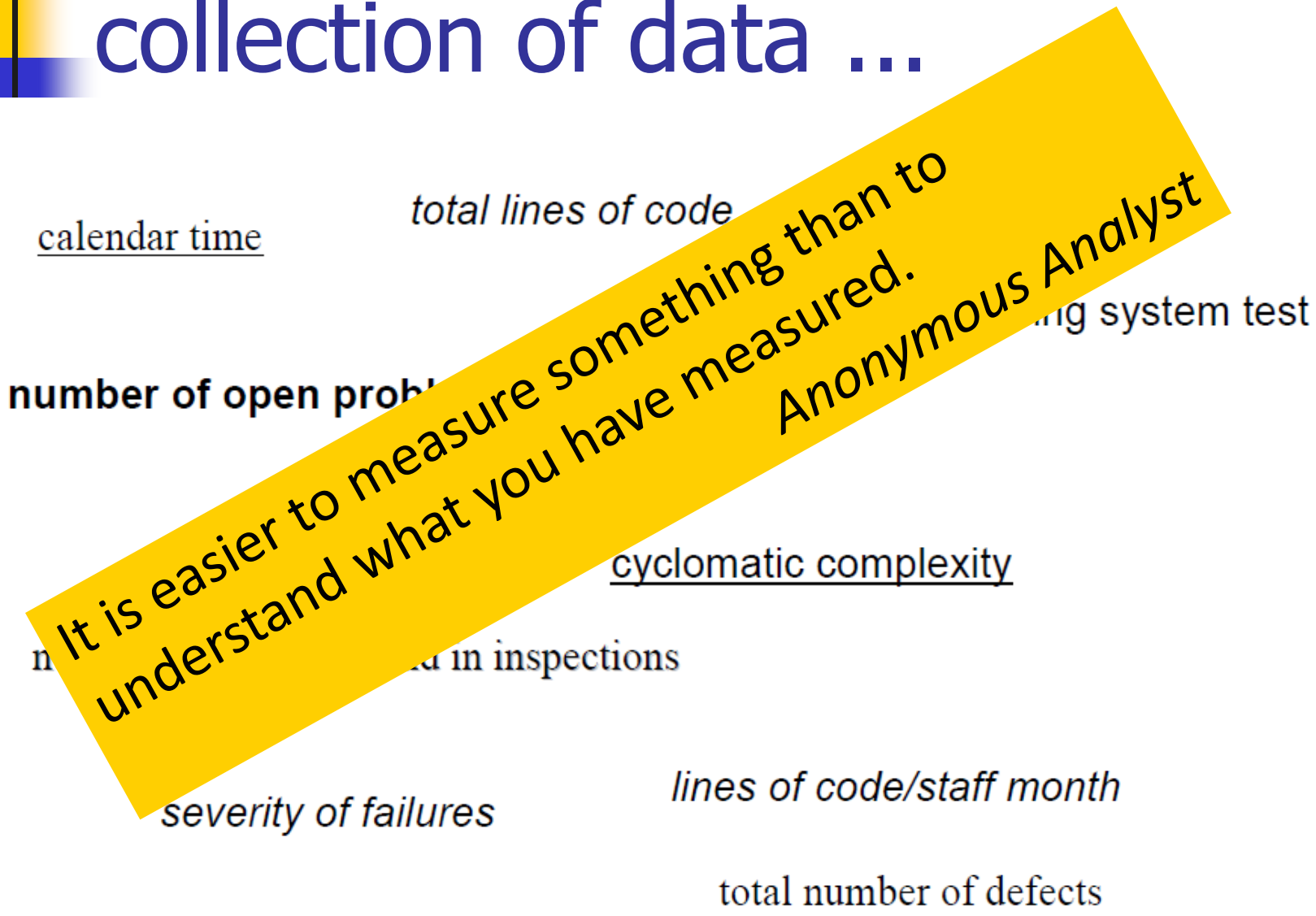
- Productivity analysis:



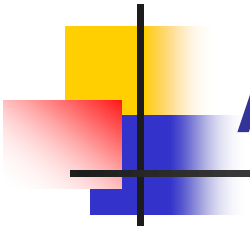
General Steps of Perform Measurement Process

- **Extract**
  - Extract information (collect measurement data) for a specific need and record it for potential further usage.
- **Evaluate**
  - Use the extracted measurement data on the spot!
  - Statistical analysis and evaluation
- **Execute**
  - Execute decisions and actions to reduce the differences between status and objectives.
  - Management decisions are directly linked to the specific need of the stakeholder

Measurement is NOT just a collection of data …



Measurement data needs interpretation ...

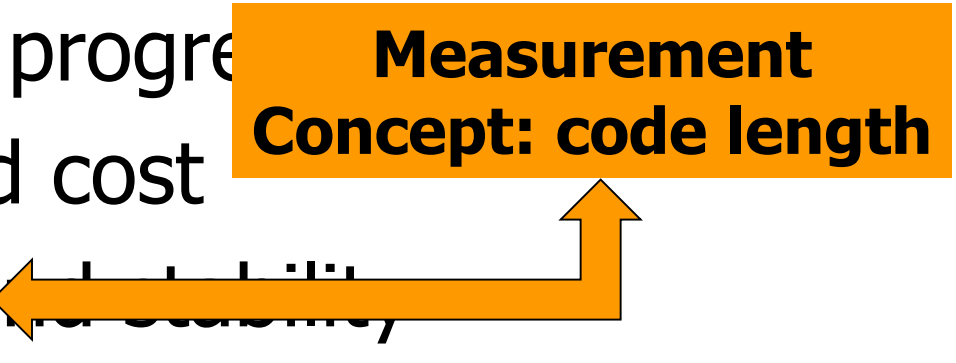


Agenda

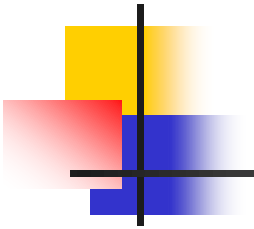
- Homework 1 (GQM)
- Review
- Plan Measurement Process. Examples
- Perform Measurement Process.
- Examples
- Homework 2 (LOC)
- Next?

Information needs: Common Categories for project management

- Schedule and progre
- Resources and cost
- Product size a
- Product quality
- Process performance
- Technology effectiveness
- Customer satisfaction



Perform Measurement Process: Code size trends example (RE: A1 soen377, 2009)

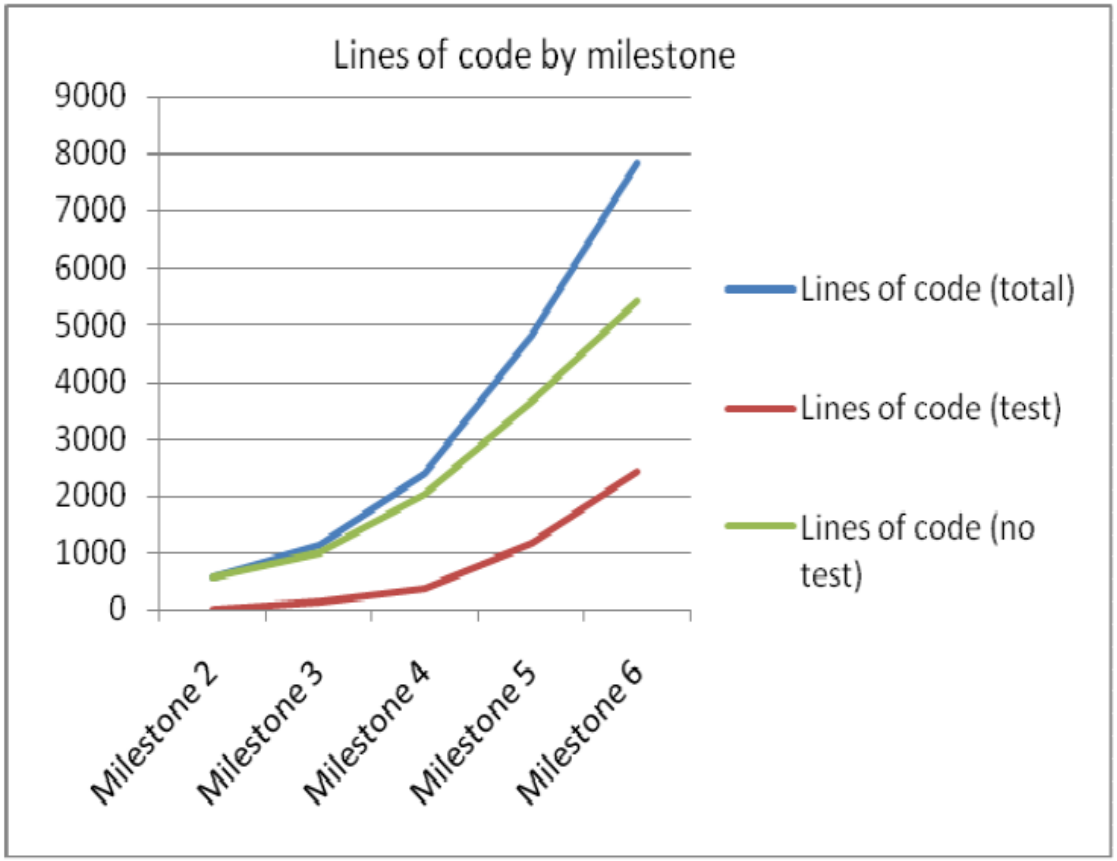


- Examining the lines of code in each milestone can give us some insight on the status of the project development
- Measures:
  - LOC
  - Percentage of comment lines of code
  - Average number of lines of code in functions
- Collect and Analyze data
  - For each measure: Show the size trends. Is the size of the code you have measured increasing or decreasing? Comment on the major size changes that occur in different milestones.

Perform Measurement Process: Code size trends analysis (RE: A1 soen377, 2009)

Communicate results:

- in Milestones 2 and 3, there is about the same amount of lines of code created for each milestone. The amount of lines of code created in Milestone 4 is slightly more, but then that amount seems to double for Milestone 5 & 6.
- the amount of code produced per milestone is a result of the team becoming more comfortable with the development of the application and becoming more efficient in their work.



Perform Measurement Process: Code size trends analysis (RE: A1 soen377, 2009)

Milestone 2	Milestone 3	Milestone 4	Milestone 5	Milestone 6
1.04%	4.33%	2.82%	3.91%	3.84%

Communicate results:

- If we consider that the average function has 10 lines (given that the median for the number of lines of codes per function is between 2 and 5 for each milestone) and that on average there should be 2 lines of comments, then:

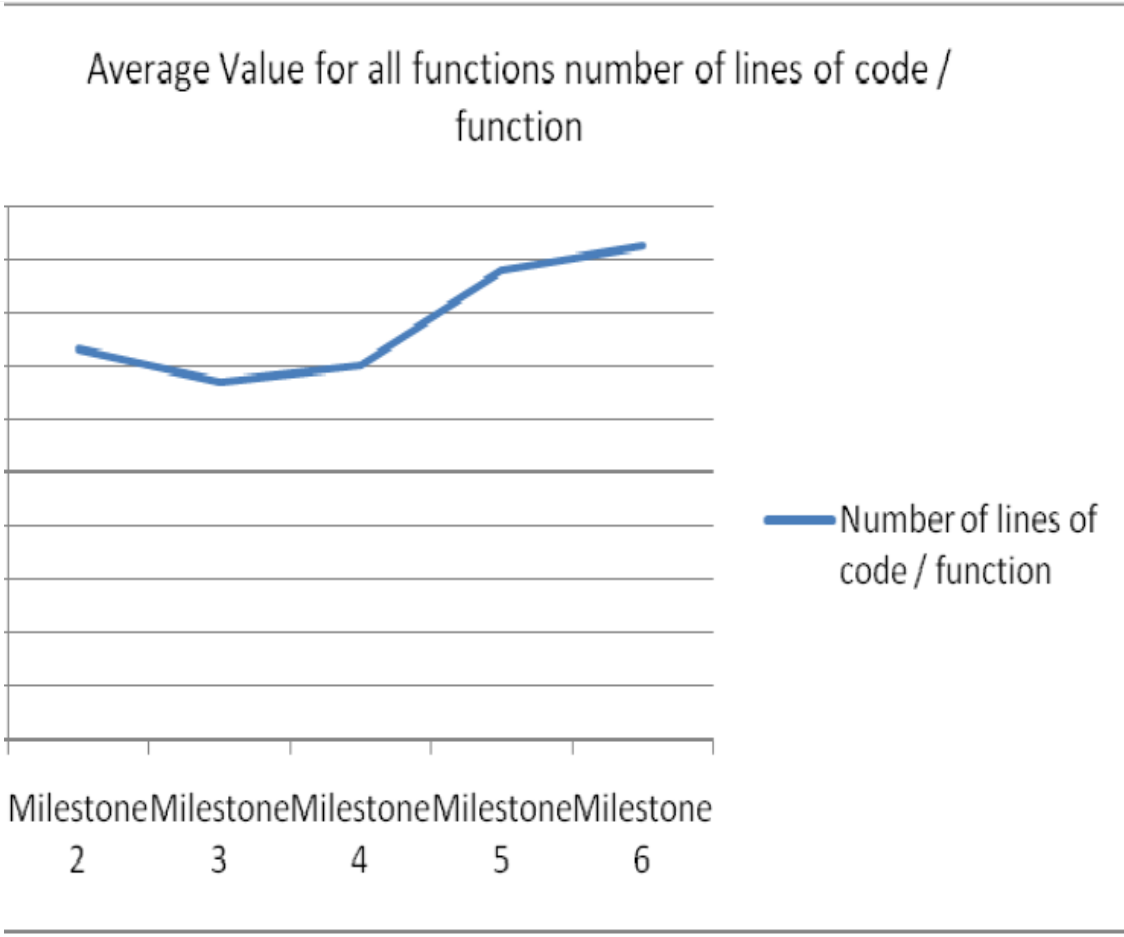
  Expected % comment = 2 / (10 + 2) = 2 / 12 = 16.7%

- no milestone has more than 5% comment lines => code documentation has to be improved

Perform Measurement Process: Code size trends analysis (RE: A1 soen377, 2009)

Communicate results:

- throughout the milestones there is a relatively low average of number of lines of code per function => the code is easy to maintain or reuse.
- There is a slight increase (approximately 25%) in the number of lines of code in Milestones 5 and 6. We hypothesize that this could be caused by an increase in the complexity of the domain logic that needs to be handled by these later milestones.



Discussion on Code Size Measures

Physical LOC [corresponds] to one line starting with the first character and ending by a carriage return or an end-of-file marker of the same line, and which excludes the blank and comment line.

Physical SLOC?

```
sum = a + b + c +
      d + e + f +
      g + h + i;
```

Discussion on Code Length Measures

Logical LOC is given by the number of "statements" in a program

- Language dependent!
- => the counting of logical LOC is ambiguous

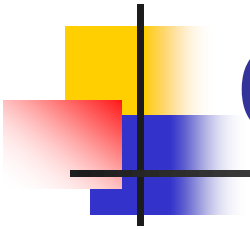
SLOC?

```
sum = a + b + c +
      d + e + f +
      g + h + i;
```

```
/* The following has a semantic error. */
if (x < 0) {
  printf("x is a positive number");
}
```

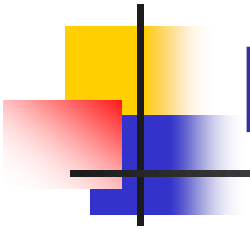
Homework 2 (1%, individual)

- Read the document LENGTH OF SOURCE CODE **BY Dr. PANKAJ KAMTHAN**
- Calculate Source LOC (SLOC) (physical or logical, blank lines or comment lines, etc.) of the provided C++ code (download from the course web page) using two different size measurement tools
- Compare the differences, interpret the results
- Submit your results and analysis electronically through the EAS as **programming assignment 1** by September 19 midnight (individual).
- WRITE YOU NAME and ID ON THE FIRST PAGE OF YOUR SUBMISSION



Questions?

- …



Next?

- Evaluate Measurement