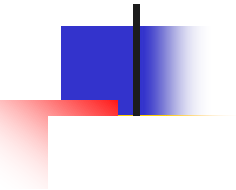
SOEN 384
Management, Measurement and Quality Control

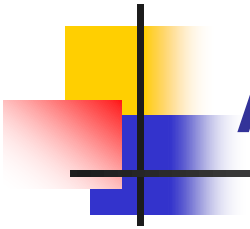
http://users.encs.concordia.ca/~s384_2/



Lecture 8:

Software Engineering Management (SEM) : A goal-based framework for software measurement

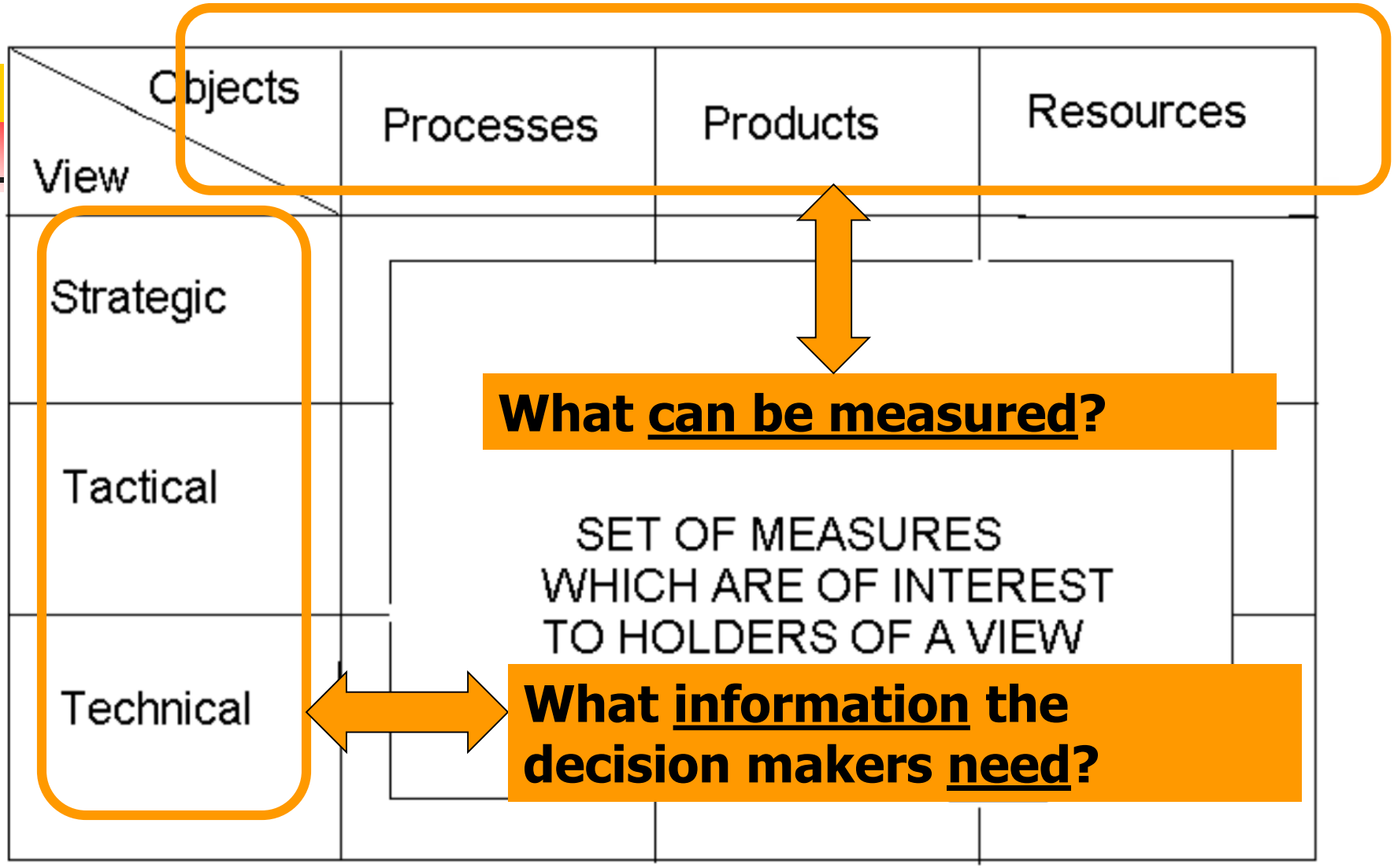
- *Fenton, chapter 3 – section 3.1*
- *Sylvie Trudel "Measurement for improving accuracy of estimates: the case study of a small software organisation"*



Agenda

- goal-based framework for software measurement program
  - Identifying the views of measurement
  - identifying the entities and attributes that we want to measure
  - internal v.s. external attributes
  - examples
- Simple measurement plan

A goal-based framework for software measurement



Measurement Program: Technical View

Emphasis on a particular process or product

- **Technical measures** are focused upon a set of **internal** attributes of a single product or process
- Measurement data supports **technical decisions** (choice of design, trade-offs, data structure, algorithms selection, etc.)
- **Primary user** of technical measurement data: **software engineer**

Measurement Program: Tactical View

Emphasis on individual project

- Goals are stated in the **estimated** or **planned** values
- **Measurement data** is used to
  - **compare** actual results to target (estimated or planned) results. Any variances are noted and investigated.
    - Examples:
      - Defect discovery rate during inspection or testing activities
      - Progress to date (percentage completion measure)
  - **predict** values of certain indirect project measures
    - Examples:
      - Product measures to predict project measures (Using project size to predict effort and schedule)
      - Resource consumption (cost of materials, consumable resources required for the project)
- **Primary user** of tactical measurement data: **project manager**

Measurement Program: Strategic View

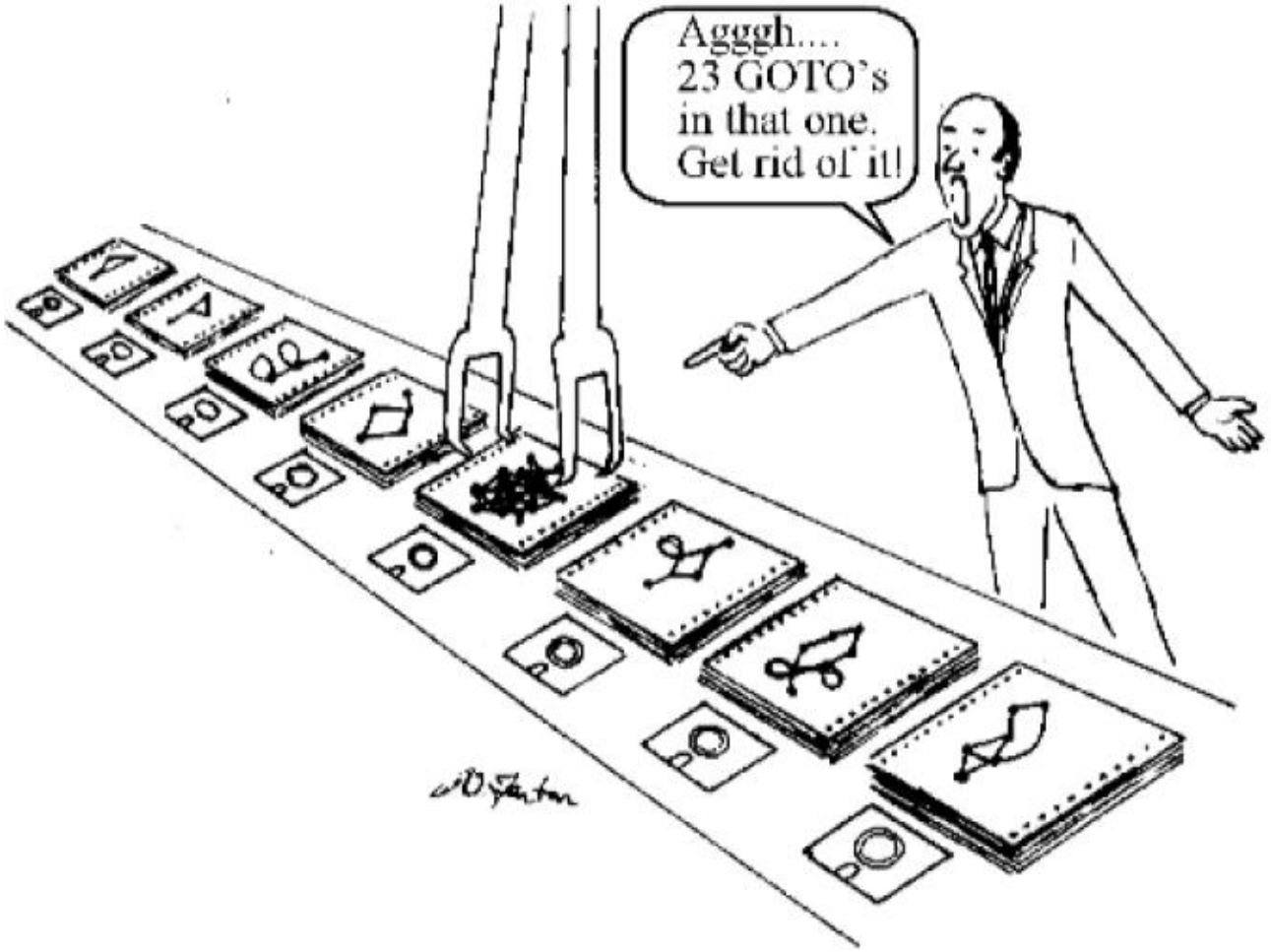
Emphasis on business processes

- Organization's goals are stated in measurable terms
- Strategic View tracks trends of summary statistics.
- **Primary user** of strategic measurement data: *strategic manager*

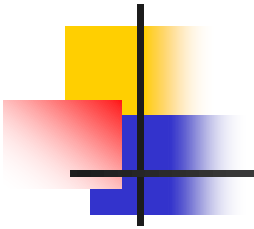
Classifying software measures

- **Software measures** are used to quantify characteristics of:
  - Software Products
  - Software Development Resources
  - Software Development Process
- This includes items which are **directly measurable**, such as *lines of code*, as well as items which are calculated **indirectly** from measurements, such as *software quality*.

Using internal measures for quality control

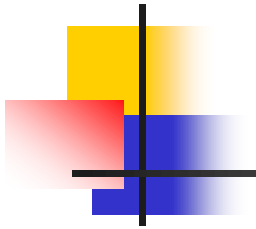


Internal Attributes v.s. External Attributes



- **Internal:** measured purely in terms of the process, project, product or resource itself
- **External:** can be measured only with respect to how the process, project, product or resource relates to its environment

Internal v.s. External Attributes



- **Examples of Internal product attributes**:
  - Size, complexity, reuse, modularity, coupling, cohesion
- **Examples of External product attributes**:
  - Understandability, usability, reusability, maintainability, Quality

Fenton's list

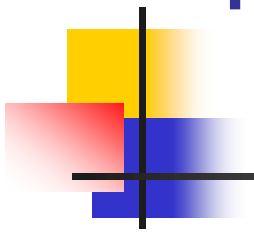
ENTITIES	ATTRIBUTES	
Products	*Internal*	*External*
Specifications	size, reuse, modularity, redundancy, functionality, syntactic correctness, ...	comprehensibility, maintainability, ...
Designs	size, reuse, modularity, coupling, cohesiveness, functionality, ...	quality, complexity, maintainability, ...
Code	size, reuse, modularity, coupling, functionality, algorithmic complexity, control-flow structuredness, ...	reliability, usability, maintainability, ...
Test data	size, coverage level, ...	quality, ...
...	...	...
Processes		
Constructing specification	time, effort, number of requirements changes, ...	quality, cost, stability, ...
Detailed design	time, effort, number of specification faults found, ...	cost, cost-effectiveness, ...
Testing	time, effort, number of coding faults found, ...	cost, cost-effectiveness, stability, ...
...	...	...
Resources		
Personnel	age, price, ...	productivity, experience, intelligence, ...
Teams	size, communication level, structuredness, ...	productivity, quality, ...
Software	price, size, ...	usability, reliability, ...
Hardware	price, speed, memory size, ...	reliability, ...
Offices	size, temperature, light, ...	comfort, quality, ...
	...	...

Entities (Objects) of Measurement: Products

- **Product:** deliverable created during the course of a project
  - Not all products are delivered to the customer!
- **Examples of Internal product attributes**:
  - Size, Complexity, Functionality, reuse, modularity, coupling, cohesion
- **Examples of External product attributes**:
  - Understandability, usability, reusability, maintainability.

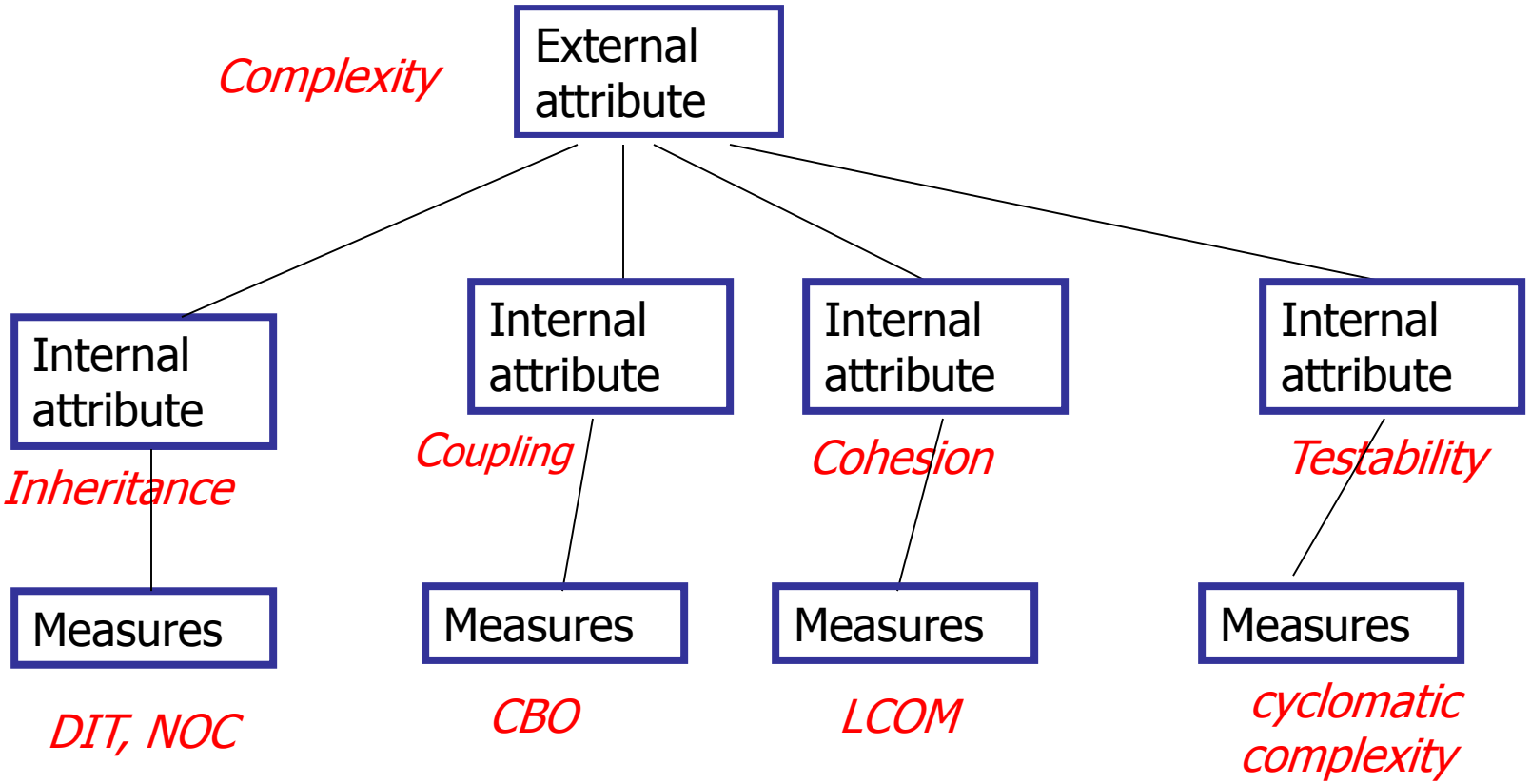
Entities of Object Oriented Software	Software Measures	The measure is characterizing the OO Software Entity
System	Weighted Methods per Class (WMC)	True / False
	% of reused "as is" classes	True / False
	Number of local methods	True / False
	Count of objects in the system	True / False
	Average Inheritance Depth	True / False
	Number of Children (NOC)	True / False
Class	Operation coupling measure (count of the number of operations that access other classes, and cooperate with other classes)	True / False
	Weighted methods per class (WMC)	True / False
	Lack of cohesion of methods (LCOM)	True / False
	Depth of inheritance tree (DIT)	True / False
Method	Cyclomatic complexity	True / False
	Attribute Inheritance Factor	True / False
	% of inherited methods that are overloaded	True / False
	Method Inheritance Factor	True / False

How do we measure OO internal attributes?



- COUPLING:
  - coupling between object classes
  - response for a class
- INHERITANCE:
  - depth of the inheritance tree
  - number of children
- COHESION:
  - lack of cohesion in methods
- TESTABILITY:
  - cyclomatic complexity

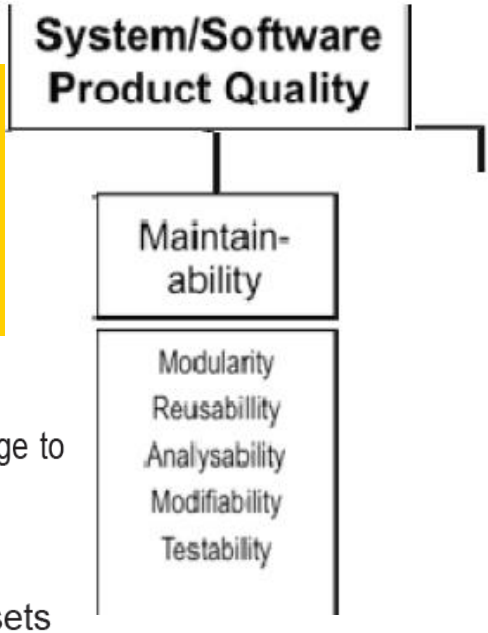
OOD Internal/External Attributes Example



ISO 25010: external attribute Maintainability

ISO/IEC 25010:2011(E)

Definition: degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers



4.2.7.1
modularity
degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components

4.2.7.2
reusability
degree to which an asset can be used in more than one system, or in building other assets

4.2.7.3
analysability
degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified

4.2.7.4
modifiability
degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality

Modifiability is a combination of changeability and stability.

4.2.7.5
testability
degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met

LOGISCOPE: Maintainability

MAINTAINABILITY = ANALYZABILITY + CHANGEABILITY + STABILITY + TESTABILITY

4.2.7.3
analysability
degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified

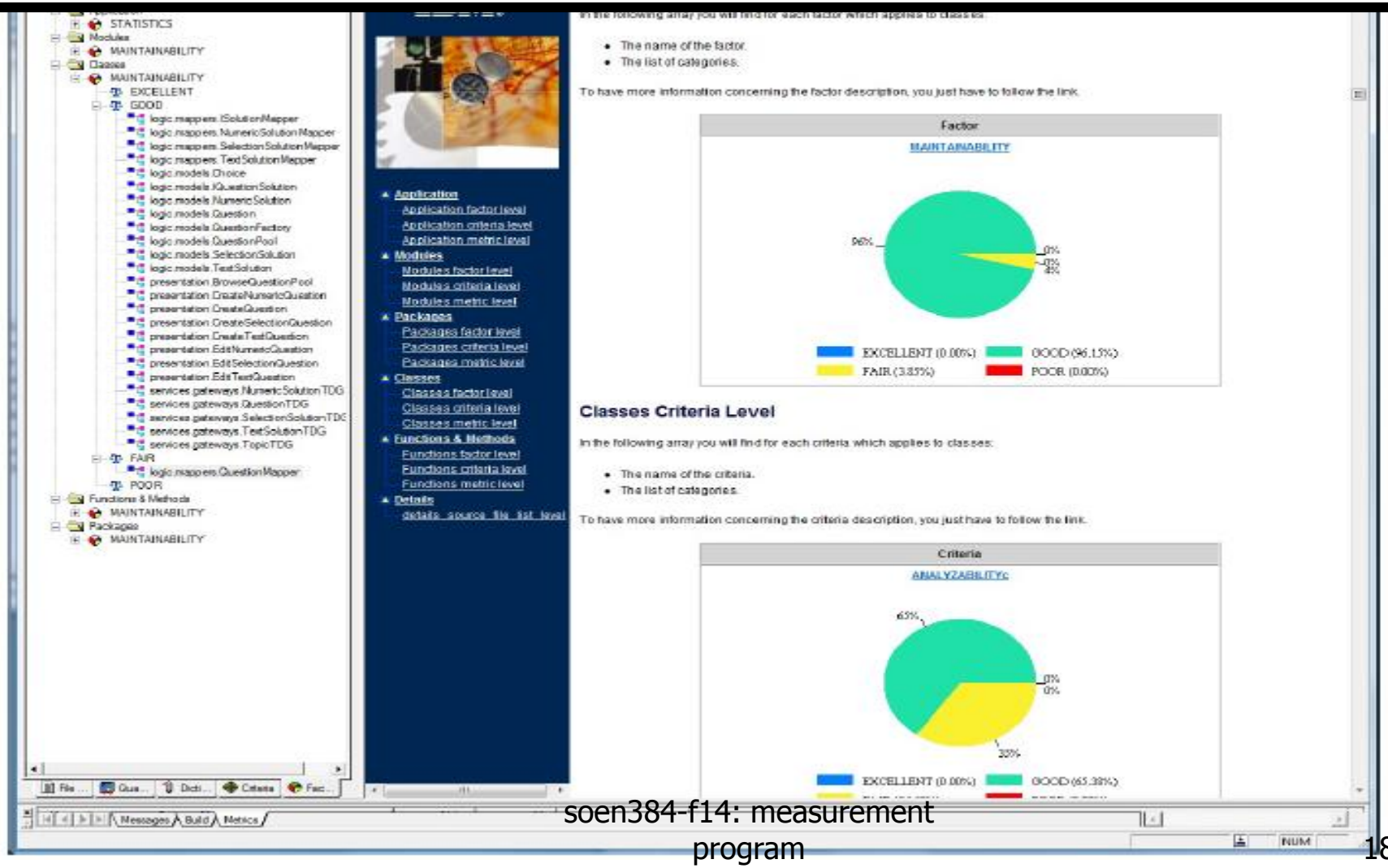
4.2.7.4
modifiability
degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality

Modifiability is a combination of changeability and stability.

4.2.7.5
testability
degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met

Logiscope tool

$$MAINTAINABILITY = ANALYZABILITY + CHANGEABILITY + STABILITY + TESTABILITY$$



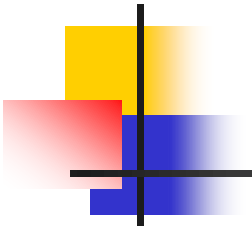
Entities (Objects) of Measurement: Resources

- **Resources** are those objects that serve as input to the processes:
  - People, tools, materials, methods, time, money, training
  - **Internal attributes:**
    - Cost, capability, constraints on use, consumption
  - **External attributes:**
    - Performance, productivity

Entities (Objects) of Measurement: Processes

- **Process:** software-related activity associated with some time-scale development process or
  - Examples: maintenance process, testing process, and so on
- **Examples of Internal process attributes**:
  - the duration of the process or one of its activities
  - the effort associated with the process or one of its activities
  - the number of incidents of a specified type arising during the process or one of its activities
- **Examples of External process attributes**:
  - Cost

Simple measurement plan – tactical view (source: Sylvie Trudel)



- Small Canadian software development organisation
- "**Not to exceed**" estimate business model, guarantees that fixing all defects found by their customer are free of charge.
- **Motivation for a measurement program**: the inaccuracy of initial estimates (half of the projects ended up exceeding estimates)
- Measurement results were used to improve the accuracy of estimation models
- With more accurate estimates, several sound business decisions were made regarding future projects

Simple measurement plan (source: Sylvie Trudel)

ID	Objectives	Reason
G1	Deliver projects within effort estimates	To reach corporate goal of 30% gross margin.
G2	Deliver defect free versions into production	Ensure product quality and customer satisfaction, minimise rework.

Identifying Quantifiable Questions and Indicators

Measurement Goals

- Object of interest
- Purpose
- Perspective
- Environment and constraints

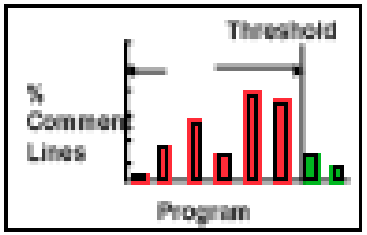
Question 1

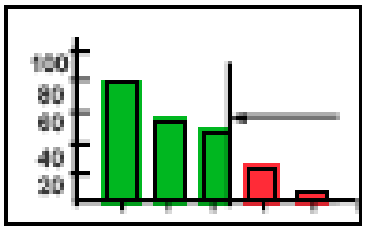
Question 2

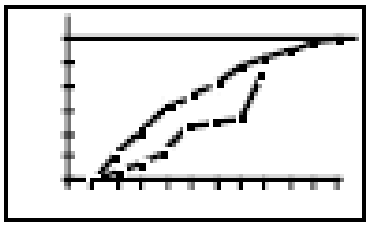
Question 3

Question 4

Indicators



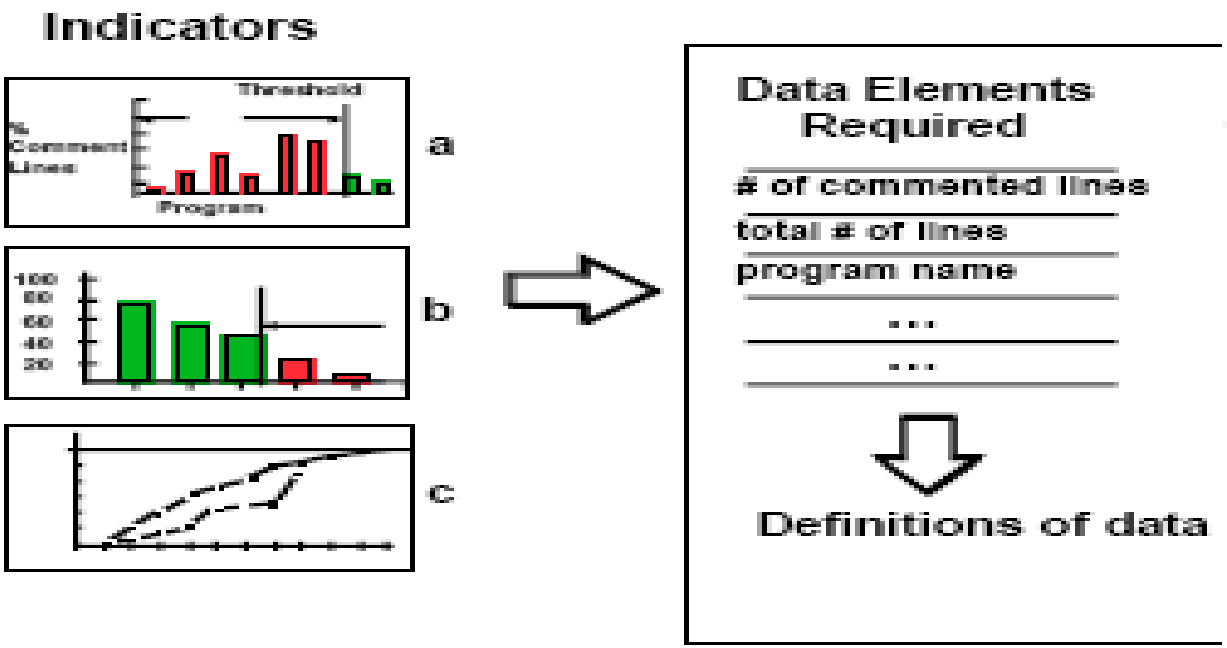




Simple measurement plan: Questions & Indicators (source: Sylvie Trudel)

Q1	Q2	Q3	Q4
For each project, what is the difference between actual effort and planned effort?	What project proportion has an overrun > 5%?	What are the differences between the planned effort and the initial Scrum detailed estimate?	How many defects do we have per year and per release?
(Actual effort - (planned effort + CRs))*100 /(planned effort + CRs)	(Number of projects of overrun>+5%)*100/ total number of projects	Planned effort – Scrum initial effort	Number of defects per release and total
G1	G1	G1	G2
- Verify that the process was applied, especially on CRs. - Verify any encountered issue.	When > 15%, adjust estimation model	- Re-estimate either plan or Scrum. - If appropriate, advise customer of an estimate change prior to beginning project.	When > 1, do a retrospective.

Identifying Measurement Data

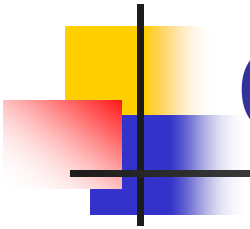


1. the data elements that you must collect to construct the indicators that you identified in the previous exercise
2. how you want the data elements to be defined, so that the indicators will show what they purport to show

Simple measurement plan: some base measures related to goal G1

ID	**M1**	**M2**	**M3**	**M4**
Measures	Actual effort	Planned effort	Total effort for all CRs	Scrum initial effort
Scope	Per project	Per project	Per project	Per project
U of M	Hours	Hours	Hours	Hours
Precision	1 hr	1 hr	1 hr	1 hr
Measured by	Employees	PM	PM	Employees
Data source	Anatime	Project plan	CR files	Scrum Works
Data collection procedure	Timesheet must be entered every day	Project < 50 hrs = manual only Project > 50 hrs = FSM	As soon as a CR is approved, enter it in the CR Follow-up table in the project plan.	As soon as Scrum initial effort is completed, the PM copies the effort value in the project portfolio file.

Simple measurement plan: Goal 2



- See "Quality-Management-SylvieTrudel.xls" file