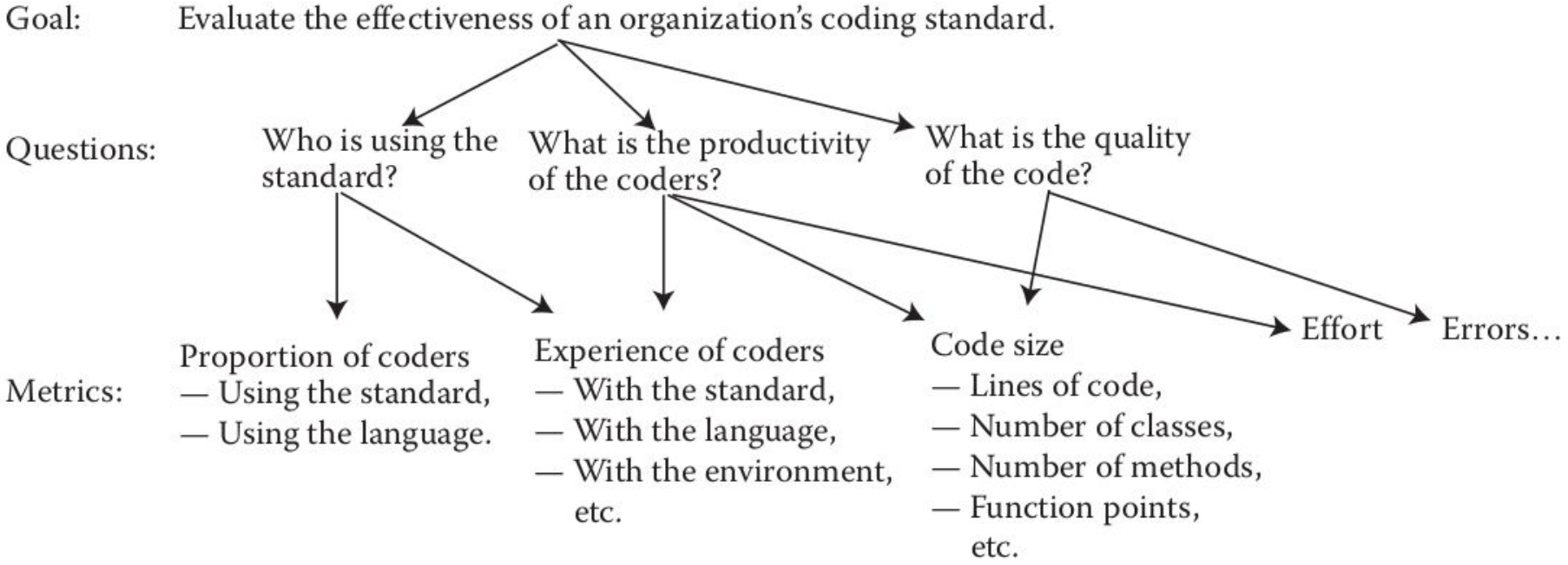
A Goal-Based Framework for Software Measurement

Determining What to Measure : GQM Paradigm

- Stands for Goal-Question-Metric
- Steps -
  - List the major goals of the development or maintenance project.
  - Derive from each goal the questions that must be answered to determine whether the goals are being met.
  - Decide what must be measured in order to be able to answer the questions adequately.

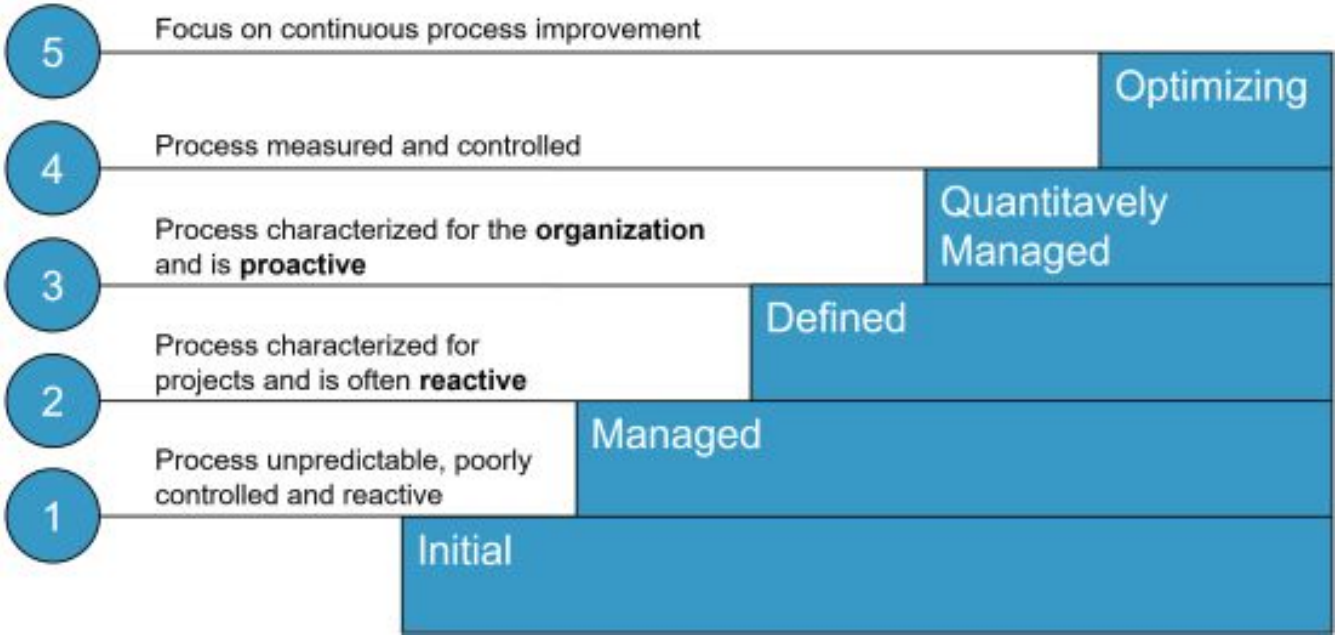
Example of GQM



Templates for Goal Definition

- **Purpose:** To (characterize, evaluate, predict, motivate, etc.) the (process, product, model, metric, etc.) in order to (understand, assess,manage, engineer, learn, improve, etc.) it.
  - Example: To evaluate the maintenance process in order to improve it.
- **Perspective:** Examine the (cost, effectiveness, correctness, defects, changes, product measures, etc.) from the viewpoint of the (developer, manager, customer, etc.)
  - Example: Examine the cost from the viewpoint of the manager.
- **Environment:** The environment consists of the following: process factors, people factors, problem factors, methods, tools, constraints,etc.
  - Example: The maintenance staff consists of poorly motivated programmers who have limited access to tools.

Measurement for Process Improvement - CMMI



***Mario, H. I. R. Z. "An approach supporting integrated modeling and design of complex mechatronics products by the example of automotive applications." (2018).*

Measurement for Process Improvement - CMMI

- Goals are defined for process areas
- Questions are asked for each goal (typically yes no questions)
- Level decided by number of responses

Combining GQM with Process Maturity

- High-level goals:
  - Improving productivity
  - Improving quality
  - Reducing risk
- the goal of improving productivity can be interpreted as several subgoals affecting resources:
  - Assuring adequate staff skills
  - Assuring adequate managerial skills
  - Assuring adequate host software engineering technology
- Similarly, improving productivity with products can mean
  - Identifying problems early in the life cycle
  - Using appropriate technology
  - Reusing previously built products

Combining GQM with Process Maturity

- if you have chosen improving quality with a subgoal of improving the quality of the requirements, then the related questions might include:
  - Is the set of requirements clear and understandable?
  - Is the set of requirements testable?
  - Is the set of requirements reusable? etc.
- Depends on CMMI level
- suppose you want to answer the question: Is the set of requirements maintainable?

Combining GQM with Process Maturity

- Level 1, then the project is likely to have ill-defined requirements. Measuring requirements characteristics is difficult at this level, so you may choose to count the number of requirements and changes to those requirements to establish a baseline
- Level 2 : the requirements are well defined you can collect addi-
- tional information: the type of each requirement (database requirements, interface requirements, performance requirements, etc.) and the number of changes to each type.
- Level 3, your visibility into the process is improved, and intermediate activities are defined, with entry and exit criteria for each activity. For this level, you can collect a richer type of measurement: e.g. - traceability