Dipartimento di Scienze Teoriche e Applicate
Università degli Studi dell'Insubria

GQM support for data-centric projects

Luigi Lavazza

Università degli Studi dell'Insubria, Varese, Italy

luigi.lavazza@uninsubria.it

The 16th International Conference on Software Engineering Advances
October 3-7, 2021 – Barcelona, Spain

Luigi Lavazza

- Luigi Lavazza is associate professor of Computer Science at the University of Insubria at Varese, Italy. Formerly he was assistant professor at Politecnico di Milano, Italy. Since 1990 he cooperates with the Software Engineering group at CEFRIEL, where he acts as a scientific consultant in digital innovation projects.
- His research interests include: Empirical software engineering, software metrics and software quality evaluation; Software project management and effort estimation; Software process modeling, measurement and improvement; Open Source Software.
- He was involved in several international research projects, and he also served as reviewer of EU funded projects.
- He is co-author of over 180 scientific articles, published in international journals, or in the proceedings of international conferences or in books.
- He has served on the PC of a number of international Software Engineering conferences; from 2013 to 2018 he was the editor in chief of the IARIA International Journal On Advances in Software.
- He is a IARIA fellow since 2011

Luigi Lavazza: research interests

- Empirical software engineering
  - Evaluation of estimation models’ accuracy
- Software metrics and software quality evaluation
- Software project management and effort estimation
- Software process modeling, measurement and improvement
- Open Source Software.

Data-driven projects: two opposite situations

1. we have a lot of data (mainly automatically collected): let's try and get some value from them. The implicit objective is getting value.
2. we have a well-defined objective, but we do not have the data to pursue it. Quite often, we do not even know IF we have the data, because we do not know exactly what data are needed to pursue the objective.

   ➔ GQM is helpful in this case.

Reasoning on objectives

- In any company/organization there is always a *hierarchy* of objectives
  - E.g., the top objective is usually “make more money” in commercial settings, “provide a better service” or “save money without decreasing the quality of the provided services” in public administrations, etc.
- At some point in the hierarchy, we have
  - the current situation (E)
  - an objective (R) which is not currently achieved (i.e., R does not hold in E)
  - a strategy (S) that applied in E is expected to cause the achievent of R
    - Applying strategy S involves achieving a set of sub-objectives, which can require specific strategies, and so on and so on.

Example

- E: software development
- R: increase the quality of code wrt defectiveness, maintainability, etc.
- There are several strategies that may lead to achieve the objective:
  - Perform code inspections
  - Search for problems via static analysis tools (like SpotBugs)
  - Improve developers' abilities via training etc.
  - Monitor software measures (SPC)
  - …

The target objective

- At some point, we are given a specific objective.
  - This may be functional to higher-level goals, but this is not our concern.
- To pursue the given objective, we often need to collect and analyse data in order to derive the knowledge we need to achieve the objective.
  - To this end, we can use the GQM method

Types of objectives

- Informative
  - Understand some phenomenon, often in terms of cause-effect relationship
  - Characterize (usually quantitatively) a situation, a product, an instrument, an activity, etc.
- Operational
  - Timely discovery of problems
    - E.g., monitor activities and get warnings when “problematic” conditions are identified
  - Guarantee consistency
    - E.g., if activity A feeds activity B, than the rate of production of A must be coherent with the expected input flow of B.
  - Improvement
    - Do something better, cheaper, more quickly, etc.

Types of objectives to which the GQM is applicable

- The GQM applies to objectives (R)
  - E.g., How much was the code improved?
- The GQM applies to strategies (S)
  - Were inspections effective?
  - How many relevant issues were found via static analysis?
  - …

Types of objectives

- Meta-objectives
  - Is a strategy working?
  - Why did a given strategy fail?
  - …

GQM

- Attaining specific objectives related to the corporate and project objectives often requires process or product measurement
  - measurement for its own sake makes no sense
- Existing models and measures cannot be reused as-is
  - it is necessary to verify if they are applicable to the specific case
- In the GQM paradigm
  - Top-down refinement of goals in metrics, via questions
  - Bottom-up interpretation and validation of the collected data, in the context at hand

Metrics and measurement

- Metrics
  - Quantification of attributes or properties of a given (possibly abstract) entity
- Measurement
  - A techniques and activity to measure the attributes of interesting entities (e.g., software processes and products)
- NOTE: the principles of metrology and metrics theory apply!

Bottom-up approaches do not work

- We cannot adopt an approach that involves
  1) Measuring all what is possibly useful
  2) Extracting the needed knowledge from the measure base on-demand when needed
- Because
  - There is no standard set of metrics.
  - The usefulness of metrics cannot be assessed out of context.
  - Measuring costs, thus it is not a good idea to spend money for collecting data that will be never be used.
  - When an objective is set, it is not automatically known what measures are actually relevant

GQM: Measurement must be defined top-down

- Metrics have to be chosen, customized and used according to goals of interest and characteristics of the context.
- Goals change: the useful metrics change accordingly.

Counterpoint

- What about "opportunistic" exploitation of existing data?
- Quite often, the availability of AI-based data analysis techniques induces requests like the following:

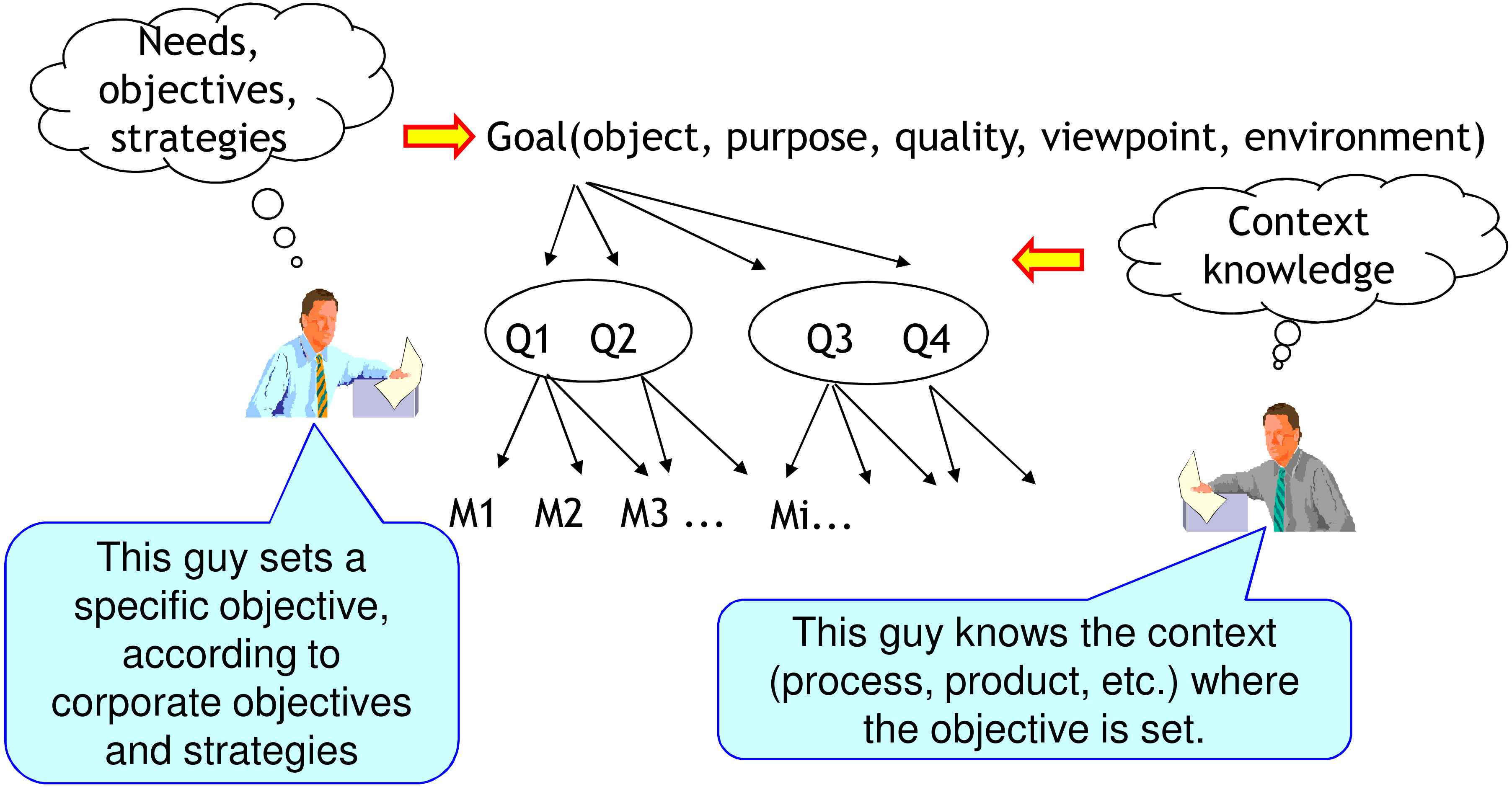
  "we have a huge bunch of data: can we get some benefit from it?"
- In fact, some AI techniques allow for this kind of bottom-up approach.
  - E.g., a large variety of production data may support the construction of models that let us identify defective products, thus allowing for savings on replacements and maintenance and improving customer satisfaction, etc.
- However, this hardly ever works with specific objectives.

Scope of GQM

- The GQM approach supports:
  - the operational definition of all kinds of measurement goals
  - their top-down refinement into metrics via questions
  - the explicit documentation of the refinement process
  - the participation of all expected beneficiaries in the goal definition and metrics identification process
  - the bottom-up interpretation of the collected data in the context of the goal

- Definition of goals and refinement can be guided by a set of templates and supported by tools.

GQM plan definition and execution



Needs, objectives, strategies

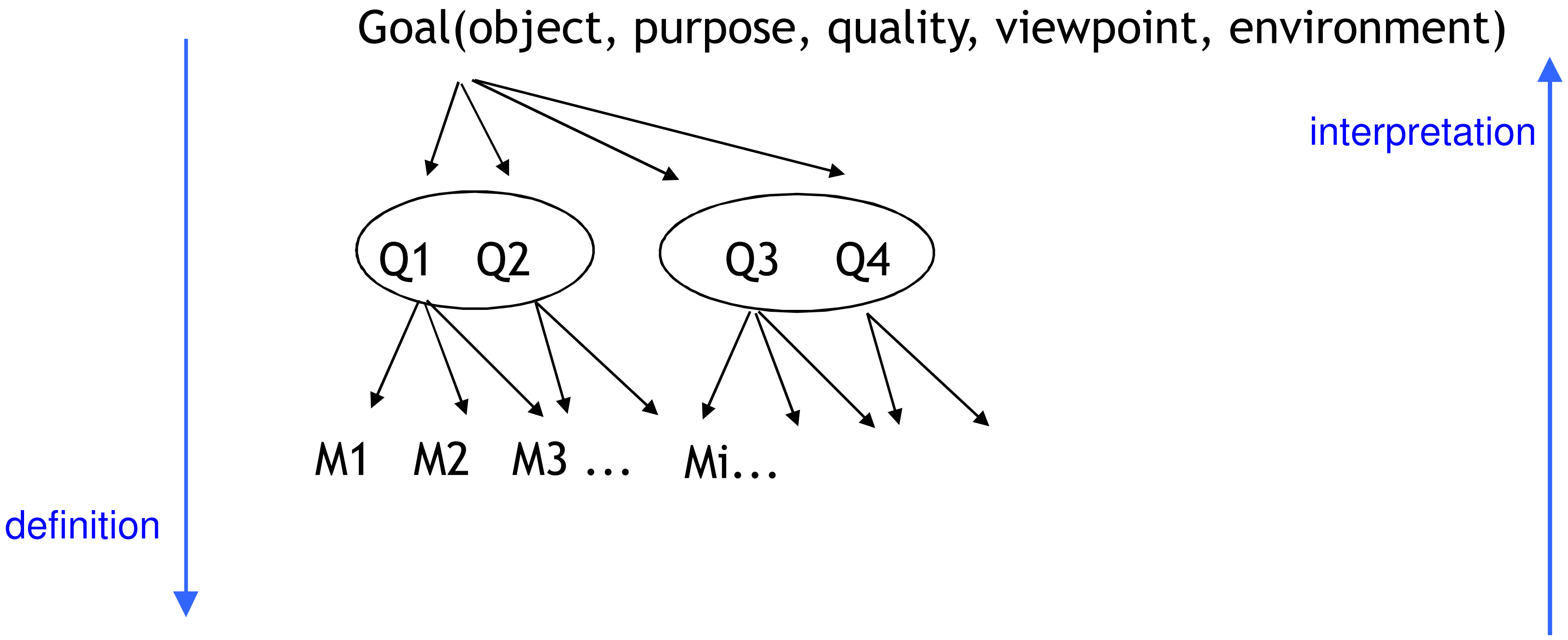
Goal(object, purpose, quality, viewpoint, environment)

Context knowledge

This guy sets a specific objective, according to corporate objectives and strategies

This guy knows the context (process, product, etc.) where the objective is set.

GQM plan definition and execution



GQM plan structure

- Goals
- A set of questions operationally defines the goal
- A set of metrics is associated with each question to provide it with a quantitative answer.

- Note:
  - A GQM plan can involve multiple goals
  - The relations between Goals and Questions are many-to-many
  - The same applies for relations between Questions and Metrics
  - The transition from Goals to questions is mediated via “abstraction sheets”

A GQM goal

- Each goal is defined
  - for an object,
  - for a variety of purposes,
  - with respect to various models of quality,
  - from various viewpoints,
  - relative to a particular environment.

Template for a GQM plan definition

- Analyze an object
  - (process, product, ...)
- for a purpose
  - (understand, evaluate, explain, optimize, control, certify, ...)
- with respect to a quality
  - (cost, correctness, safety, reliability, usability, effectiveness, speed, ...)
- from a viewpoint
  - (user, purchaser, manager, developer, company, ...)
- in a given environment
  - (persons, groups, departments, ...)

A GQM goal example

• Analyze the SW development process	• [object]
• for the purpose of improving	• [purpose]
• code maintainability	• [quality]
• from the developers’ viewpoint	• [viewpoint]
• in the context of ACME mobile app department	• [environment]

The abstraction sheet

- Deriving questions from goals directly may be difficult.
- Abstraction sheets help in this step
- An abstraction sheet identifies
  - the features to be measured,
  - the factors that may affect the object’s features,
  - the current perceived situation and the way variation factors are believed to affect the observed qualities.
- Questions operationally define each part of the abstraction sheet

Abstraction Sheet

- It is a high level view of the questions
- It is used to
  - acquire information from the project team
  - communicate information with people who do not participate in the experiment

GOAL: object, purpose, quality, point of view, environment	
quality foci	variation factors
baseline hypotheses	impact on baseline hypotheses

Example - abstraction sheet

GOAL	object: SW development process	purpose: improve code	quality: maintainability	viewpoint: developers	environment: ACME mobile app department

Quality focus • Effort to maintain code • Number of defects introduced by maintenance activities • Effectiveness of maintenance	Variation factors • size • programming language • security requirements • maintainers’ experience • …
Baseline hypotheses • *Current hypotheses concerning the quality foci and the variation factors* • *E.g., a medium difficulty change request involving only a low complexity method requires 2 hours, on average*	Impact on baseline hypotheses • *Current hypotheses concerning the impact of variation factors on baseline hypotheses* • *E.g., larger method require more effort; methods with stricter security requirements*

On the definition of GQM goals

- Original formulation:
  - Object: the SW development process
  - Purpose: improving
  - Quality: code maintainability
- Alternative 1:
  - Object: the SW maintenance process
  - Purpose: improving
  - Quality: code maintainability
- Alternative 2:
  - Object: the SW maintenance process
  - Purpose: improving code
  - Quality: maintenance

Are these goal formulations equivalent to each other?

On the definition of GQM goals

- Object:
  - the SW development process

  vs.
  - the SW maintenance process
- Are these objects equivalent (considering that the quality being evaluated is maintainability, in both cases)?
- In the former case we consider also activities concerning the development of new parts of code.
  - Which may be responsible for delivering hardly maintainable code
- The former formulation is more comprehensive

On the definition of GQM goals

- Purpose & quality:
  - improving code maintainability

  vs.
  - improving code maintenance
- Are these objects equivalent?
- In the latter case we address an activity, in the former we address the product.
- Improving the maintenance activity does not necessarily involve improving the product, but may involve additional/different improvements.
  - For instance, making maintenance faster and cheaper without changing the quality of the product satisfies the latter improvement, but not the former

Example – refining quality foci into questions

- Quality focus
  - Effort to maintain code
  - Number of defects introduced by maintenance activities
  - Effectiveness of maintenance
- Questions
  - How much effort is required to maintain code?
  - What is the maintenance effort required in relation to the size of code being maintained?
  - How many defects are introduced by maintenance activities?
  - How many change requests are successfully completed?

Example: taking variation factors into consideration

- To account the Variation factor: programming language
- *How much effort is required to maintain code?* becomes
  - *How much effort is required to maintain Java code?*
  - *How much effort is required to maintain Kotlin code?*
- *How many defects are introduced by maintenance activities?* Becomes
  - *How many defects are introduced in Java code by maintenance activities?*
  - *How many defects are introduced in Kotlin code by maintenance activities?*

Example: taking variation factors into consideration

- To account the Variation factor: programming language
- *How much effort is required to maintain code?*

  becomes
- *How much effort is required to maintain Java code?*
- *How much effort is required to maintain Kotlin code?*

Example: taking variation factors into consideration

- Independent variation factors tend to combine.
- When considering both the programming language and maintainers' experience, you get that different effort is needed by
  - Maintenance of Java code performed by inexperienced programmers
  - Maintenance of Java code performed by experienced programmers
  - Maintenance of Kotlin code performed by inexperienced programmers
  - Maintenance of Kotlin code performed by experienced programmers
- If we considered also high/low criticality, we would end up with 8 combinations.

Example – refining questions into metrics

- Question: *How much effort is required to maintain Java code?*
- Metrics:
  - Total effort [person*hours] dedicated to the Java code of product XYZ to release version n.m.
  - Total effort [person*hours] dedicated to the Java code of product XYZ to release version n.m, per class.
  - Total effort [person*hours] dedicated to the Java code of product XYZ to release version n.m, per method.
  - Relative effort [person*hours/LOC] spent Java code to release version n.m of product XYZ, per line of code.
  - Relative effort [person*hours/FP] spent Java code to release version n.m of product XYZ, per Function Point.

Refinements: what do you need?

- The refinement of goals into (abstraction sheets, and) questions and metrics has to be done by someone who knows the context
- For instance, in the example above the person in charge of the refinement needs to know:
  - That the considered code can be written in Java or in Kotlin
  - That developers have different levels of expertise
  - That some parts of the code may have security issues while others do not
  - Etc.
- Note: in general it is advisable to involve several persons who are familiar with the environment, to take into account all the relevant aspects of the process or product.

Refinements: what support is available?

- It is possible to build guidelines that indicate the attributes and qualities of software processes and products that are most frequently evaluated to pursue typical objectives in software processes.
- We shall sketch some guideline fragments
- Suggestion: build your own guidelines
  - By recording what attributes and qualities you address in each plan you define and execute.

Guidelines for product-related questions

- Definition of the product
  - Internal attributes
    - “physical” size, complexity, structure properties (cohesion, coupling, etc.), ...
  - External attributes of interest to users
    - “functional” size, ease of use, response time, resources required, security, criticality, license, compliance to laws (e.g., GDPR), …
  - External attributes of interest to developers
    - Maintainability, defectiveness, efficiency, dependence on libraries and components, …
  - External attributes of interest to managers
    - User profile, cost of operation, …

Guidelines for product-related questions

- Questions concerning the GQM plan definition and execution
  - What quality models have been used
  - What is the validity of the models used
  - What is the quality of collected data (wrt completeness, correctness, accuracy, etc.)
  - …
- Feedback: questions related to the improvement of the product
  - While we are retrieving data from developers and users, we can as well as them if they have any proposal for improving the product

These questions apply also to the process, environment and other objects of interest.

Guidelines for process-related questions

- Process characteristics
  - Cost and duration of the process activities
  - Characterization of activities in terms of required resources
  - Characterization of activities in terms of internal mechanisms
  - Characterization of activities in terms of input and output flows
- Process conformance
  - assessment of how well the process is performed with respect to “normal” or expected execution
- Process suitability
  - Assessment of the process characteristics in relation to the environment, the product, resources, etc.
  - E.g., was a waterfall lifecycle used when an agile process would have been better suited for the purpose?

Guidelines for environment-related questions

- Resource availability
- Requirements change
- Users (type, number, preferences, …)
- Technology evolution
- …

Dealing with cost

- Identifying questions and metrics is relatively easy.
- That is, creating a measurement plan is relatively cheap.
- However, actually collecting measures is often
  - Time consuming & expensive
  - A long process
    - Some measures concern activities that are carried out by humans at a relatively slow pace. For instance, you cannot measure the duration of an activity before it is completed: if the activities being observed is slow, data collection will be slow as well.
- So, what if the GQM plan is too slow and expensive?

Dealing with cost

- In general it is advisable to proceed in two steps:

1. The “ideal” GQM plan is created.
    - All the relevant questions and metrics are taken into account, without bothering about cost and collection time.
2. The plan is evaluated critically.
    - Metrics are evaluated with respect to feasibility, cost, collection time, relevance, …
3. The plan is streamlined, to account for constraints on measurement budget and the time available.
    - This may involve some trade-off between goals and practical measurement activities.

Types of metrics

- When defining and executing a GQM plan, several types of metrics can be needed.
  - Objective vs. subjective
  - Generic vs. specific
  - Complex vs. simple
  - Direct vs. indirect
  - Standard vs. ad-hoc
  - …

Types of metrics: objective vs. subjective

- Some properties can be defined precisely and objectively.
- Some can’t
- When a property is well defined it is easier to collect it automatically. Besides, its meaning is not subject to interpretation.
  - Examples: the size in Megabytes of an executable is a property that is defined precisely and objectively. There is hard any possibility of discussing what does it mean that you need 3.6 MB space to install an application.
- However, some properties cannot be defined precisely and objectively
  - Examples: source code complexity, code understandability, …

How to deal with subjective metrics

- Ask knowledgeable people
- Provide them with a suitable scale
  - E.g., high, medium-high, medium-low, low
  - Suggestion: avoid the “central” value, otherwise many respondents will choose noncommittally that value
- Provide them with a precise interpretation of the scale
  - E.g., what does it mean that a piece of code has high complexity?
  - Without such guideline, different respondents will grade differently the same entity
- In case of disagreement among experts, use available techniques for reaching the consensus.
  - E.g., the Delphi method.

Types of metrics: generic vs. specific

- Quite often, there is a generalization relationship between attributes
  - E.g., Java code is code (as is C code)
- The same measure can be applied to both generic and specific attributes
  - E.g., the number of lines of code can be counted, whatever the language
  - So we have generic LoC as well as Java LoC, C LoC, etc.
- As usual with generalization/specialization, it is useful to use specific measures <u>only if</u> there is the perception that this makes some difference.
  - E.g., the implementation language is a variation factor

Types of metrics: complex vs. simple

- A measure can take into account more or less detail

- E.g., we can count the number of defects (a simple count)

  or we can characterize each defect by type, cause, detection time, criticality, effort to detect, effort to eliminate, etc.

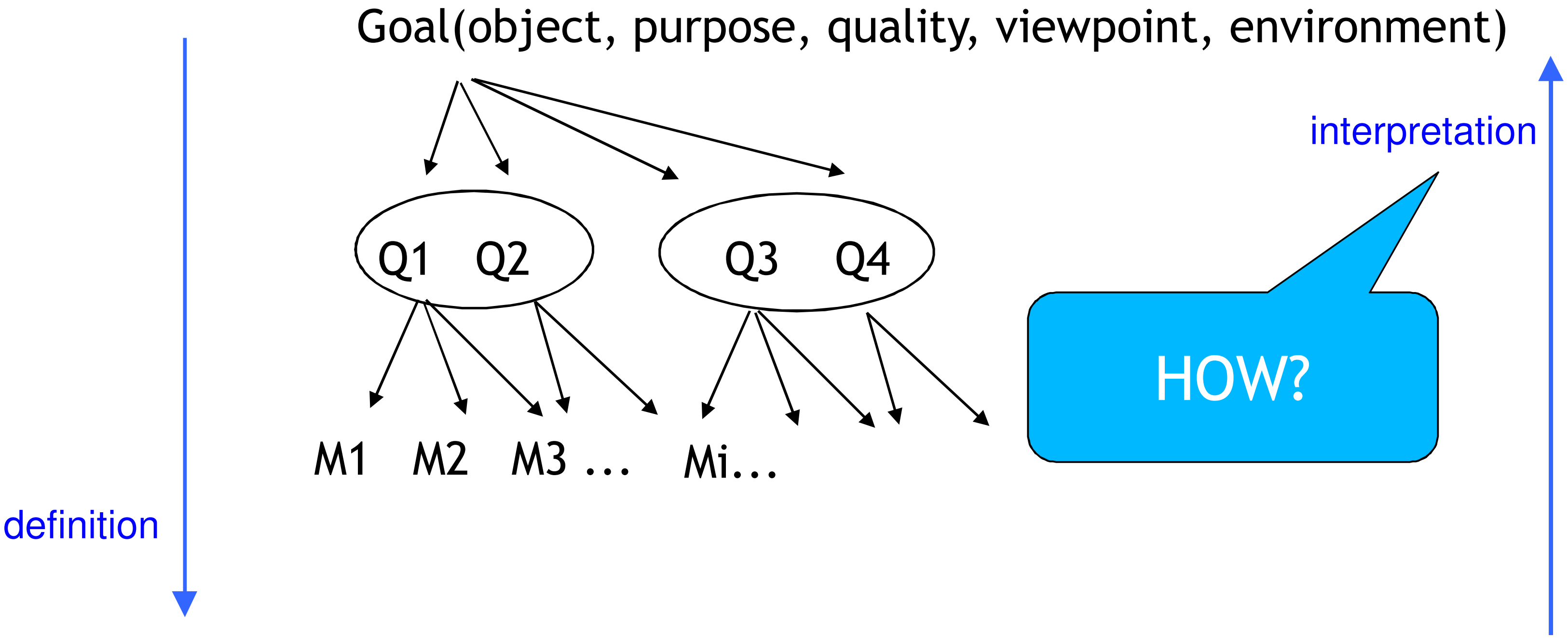
Types of metrics: direct vs. derived

- Direct measures represent properties of the given entities
  - E.g., the lines of code, McCabe complexity of a method, etc.
- Some measures are obtained by combining more elementary metrics.
  - E.g., one could define defect density as the number of defect per LOC, or development productivity as the number of Function Points obtained per Person*Hour, etc.

How to collect metrics

- Some measures are effectively collected by measurement tools
  - E.g., source code measures like LOC, McCabe complexity, coupling measures (CBO), cohesion measures (LCOM), etc.
- Some measures can be obtained by querying the repositories of development tools
  - E.g., we can query the repository of Jira to retrieve information about problems and issues
  - Note: in some cases the information is “mined”
- Some measures are obtained by asking people
- …

GQM plan definition and execution



Different types of “interpretation” activities

- When cause-effect relationships have to be derived, fairly sophisticated technique have to be used
  - From plain least-squares regression to neural networks
- When the aim is monitoring and control, typical Statistical process Control technique can be used
- When the aim is to retrieve information, or to make the situation explicit, dashboarding techniques are suitable.

Dipartimento di Scienze Teoriche e Applicate
Università degli Studi dell'Insubria

Example

Software Process Measurement in the Real World

Context

- A GQM measurement project was carried out in an organization dedicated to the maintenance of banking software with operating constraints:
  - limit the cost and duration of the measurement activities
  - exploit as much as possible the data that could be extracted automatically from development and measurement tools already in use.
- Despite these limiting constraints, by means of the GQM we achieved the initial goals to a good extent, in only three months, and spending a small amount of resources.

The environment

- An organization devoted to the maintenance of a dozen banking applications
  - Java, SQL, and HTML code.
  - Size ranging from about 30 KLOCs to over 500KLOCs.
- The maintenance process employed 41 full-time people (13 employees and 28 people hired from external organizations) organized in three groups, each coordinated by a maintenance team leader.

Previous measurement activities

- Previously, the management had started two measurement initiatives in order to support estimation activities and decision-making.
  - The CAST tool was used to measure the static properties of the managed software.
    - The whole set of applications was measured every three months.
      - LOCs, number of artifacts, backfired function points, number of files, number of classes, average Java coupling and complexity, number of SQL artifacts, average SQL coupling and complexity, number of web pages.
    - The variation of these measures between subsequent versions were computed.
    - CAST computed a set of high-level indicators.

Previous measurement activities

2) Change Requests (CRs) stored in ClearQuest were also measured.
   - the number of CRs per application and per state (according to a standard lifecycle) were measured.

The management questions

- Although these former activities provided the management with some useful data, they were not able to satisfy more complex evaluation needs, which the management expressed as a set of questions:
  - *Are we doing our job well?*
  - *Is the quality of the managed applications good?*
  - *How good are the people in charge of maintenance?*
  - *Are the customers satisfied?*
- These questions were originated by the need to control, verify, estimate and evaluate the process and products, and ultimately to support management decisions.

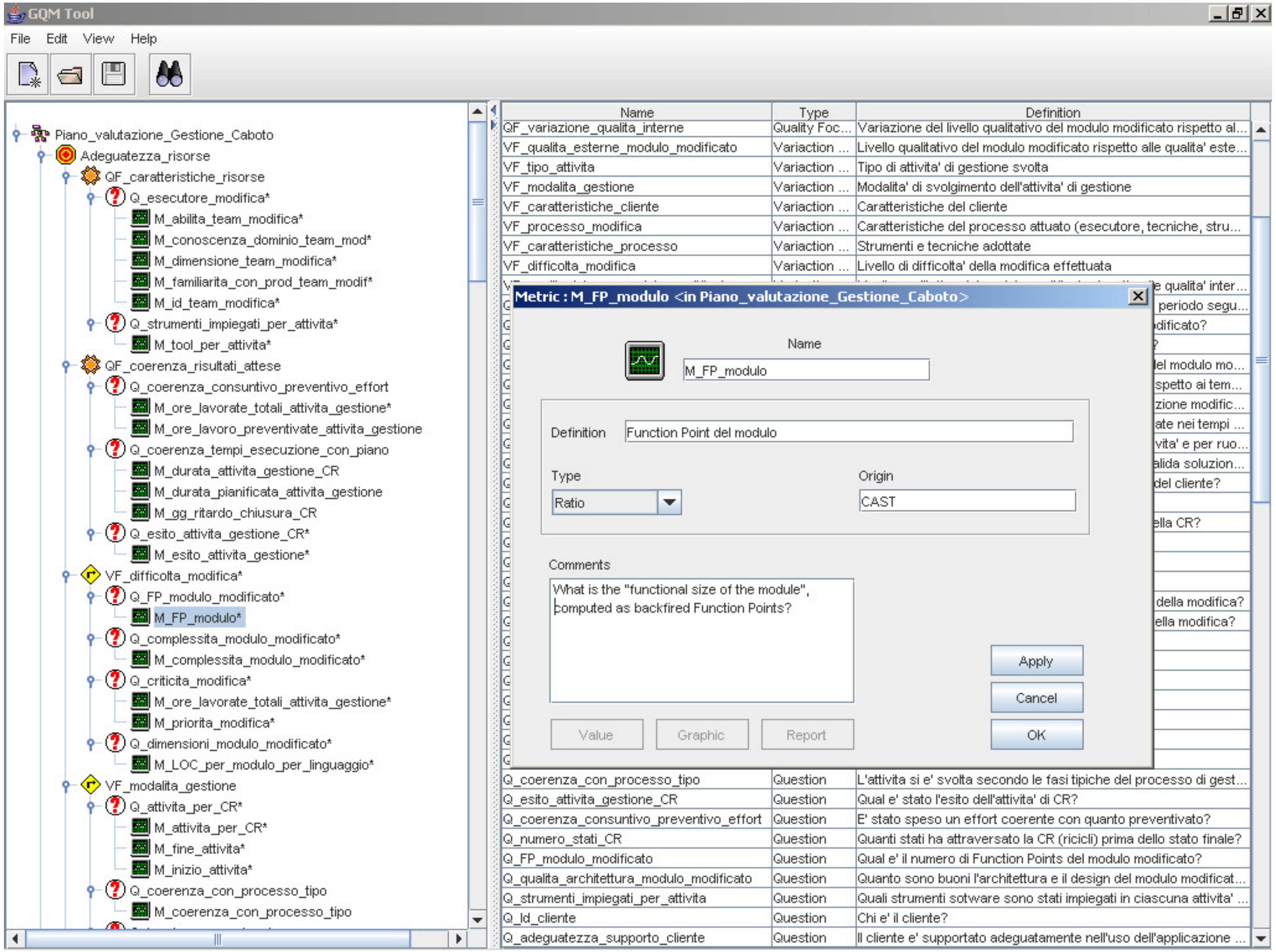
The planning phase

- The planning phase was carried out without taking into consideration any constraint.
  - it was not known in advance which metrics it was going to be possible to collect and which not.
  - the GQM team expected that the unconstrained GQM plan could provide a framework for assessing the relevance and quality of the available metrics, and for evaluating their meaning and reliability.

The planning phase

- The strategic goals given by the management were translated into the following GQM goals:
  - Analyze the maintenance process for the purpose of evaluating the quality of the product, form the point of view of the management of the organization.
  - Analyze the maintenance process for the purpose of evaluating the duration and cost of maintenance activities, form the point of view of the management of the organization.
  - Analyze the resources employed in the maintenance process for the purpose of evaluating their adequacy, form the point of view of the management of the organization.

The GQM tool



- Piano_valutazione_Gestione_Caboto
  - Adeguatezza_risorse
    - QF_caratteristiche_risorse
      - Q_esecutore_modifica*
        - M_abilita_team_modifica*
        - M_conoscenza_dominio_team_mod*
        - M_dimensione_team_modifica*
        - M_familiarita_con_prod_team_modif*
        - M_id_team_modifica*
      - Q_strumenti_impiegati_per_attivita*
        - M_tool_per_attivita*
    - QF_coerenza_risultati_attese
      - Q_coerenza_consuntivo_preventivo_effort
        - M_ore_lavorate_totali_attivita_gestione*
        - M_ore_lavoro_preventivate_attivita_gestione
      - Q_coerenza_tempi_esecuzione_con_piano
        - M_durata_attivita_gestione_CR
        - M_durata_pianificata_attivita_gestione
        - M_gg_ritardo_chiusura_CR
      - Q_esito_attivita_gestione_CR*
        - M_esito_attivita_gestione*
    - VF_difficolta_modifica*
      - Q_FP_modulo_modificato*
        - M_FP_modulo*
      - Q_complessita_modulo_modificato*
        - M_complessita_modulo_modificato*
      - Q_criticita_modifica*
        - M_ore_lavorate_totali_attivita_gestione*
        - M_priorita_modifica*
      - Q_dimensioni_modulo_modificato*
        - M_LOC_per_modulo_per_linguaggio*
    - VF_modalita_gestione
      - Q_attivita_per_CR*
        - M_attivita_per_CR*
        - M_fine_attivita*
        - M_inizio_attivita*
      - Q_coerenza_con_processo_tipo
        - M_coerenza_con_processo_tipo

Name	Type	Definition
QF_variazione_qualita_interne	Quality Foc...	Variazione del livello qualitativo del modulo modificato rispetto al...
VF_qualita_esterne_modulo_modificato	Variaction ...	Livello qualitativo del modulo modificato rispetto alle qualita' este...
VF_tipo_attivita	Variaction ...	Tipo di attivita' di gestione svolta
VF_modalita_gestione	Variaction ...	Modalita' di svolgimento dell'attivita' di gestione
VF_caratteristiche_cliente	Variaction ...	Caratteristiche del cliente
VF_processo_modifica	Variaction ...	Caratteristiche del processo attuato (esecutore, tecniche, stru...
VF_caratteristiche_processo	Variaction ...	Strumenti e tecniche adottate
VF_difficolta_modifica	Variaction ...	Livello di difficolta' della modifica effettuata
V		e qualita' inter...
Q		periodo segu...
Q		dificato?
Q		?
Q		el modulo mo...
Q		spetto ai tem...
Q		zione modific...
Q		ate nei tempi ...
Q		vita' e per ruo...
Q		alida soluzion...
Q		del cliente?
Q		
Q		ella CR?
Q		
Q		
Q		
Q		della modifica?
Q		ella modifica?
Q		
Q		
Q		
Q		
Q		
Q		
Q		
Q_coerenza_con_processo_tipo	Question	L'attivita si e' svolta secondo le fasi tipiche del processo di gest...
Q_esito_attivita_gestione_CR	Question	Qual e' stato l'esito dell'attivita' di CR?
Q_coerenza_consuntivo_preventivo_effort	Question	E' stato speso un effort coerente con quanto preventivato?
Q_numero_stati_CR	Question	Quanti stati ha attraversato la CR (ricicli) prima dello stato finale?
Q_FP_modulo_modificato	Question	Qual e' il numero di Function Points del modulo modificato?
Q_qualita_architettura_modulo_modificato	Question	Quanto sono buoni l'architettura e il design del modulo modificat...
Q_strumenti_impiegati_per_attivita	Question	Quali strumenti sotware sono stati impiegati in ciascuna attivita' ...
Q_Id_cliente	Question	Chi e' il cliente?
Q_adeguatezza_supporto_cliente	Question	Il cliente e' supportato adeguatamente nell'uso dell'applicazione ...

Metric : M_FP_modulo <in Piano_valutazione_Gestione_Caboto>

Name: M_FP_modulo

Definition: Function Point del modulo

Type: Ratio

Origin: CAST

Comments:
What is the "functional size of the module",
computed as backfired Function Points?

GQM support for data-centric projects

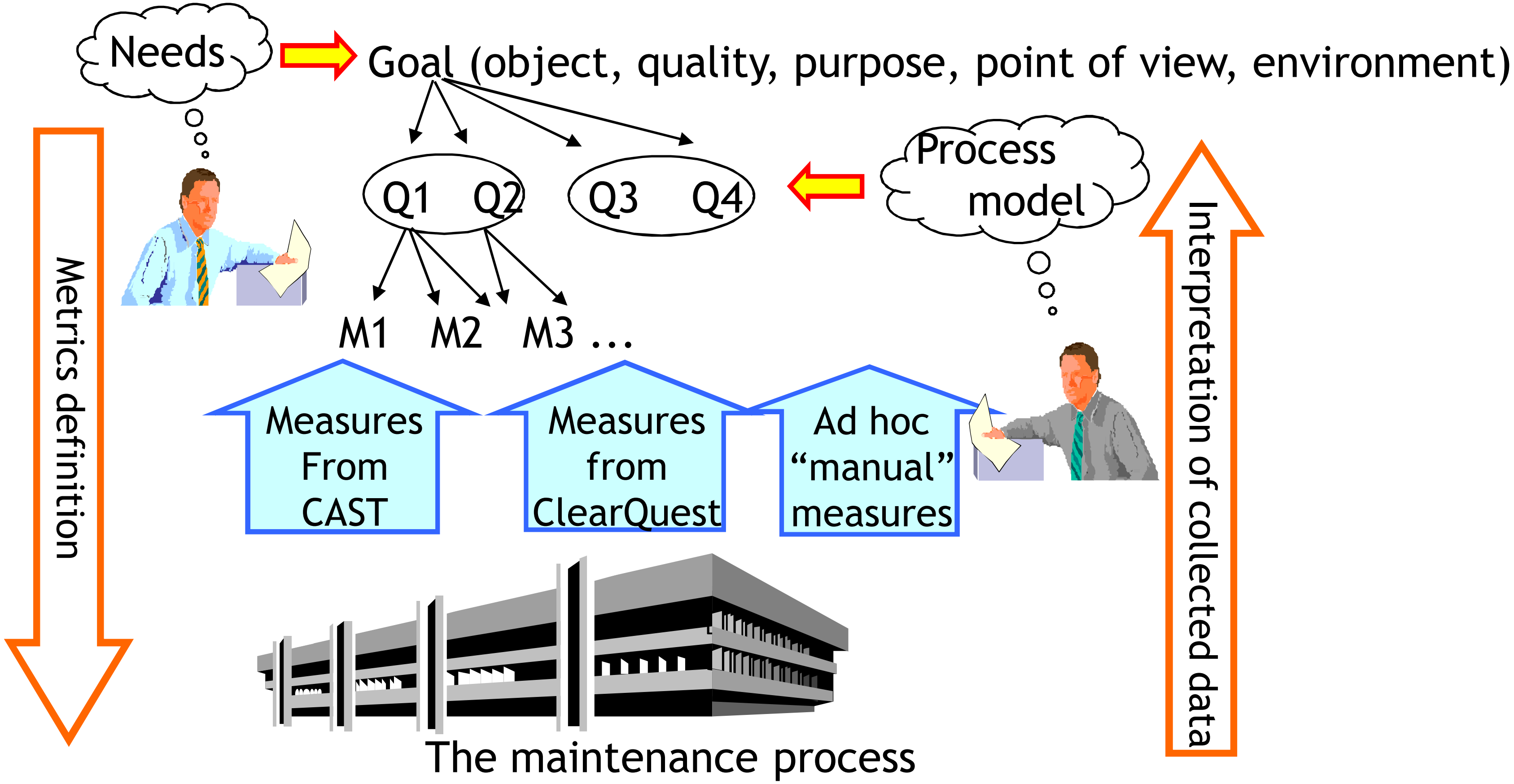
The GQM plan

- The complete GQM plan included 37 questions and 58 metrics.
- Most metrics concerned Change Requests.
- Every CR was characterized in terms of:
  - Time and effort spent.
  - Type. Maintenance activity included defect corrections as well as enhancements.
  - Lifecycle. Every CR was characterized by the sequence of states that were entered since the submission, until it reached a final state.
  - Application involved. Every CR concerned a specific application.
  - Amount and quality of the resources employed to perform the change.
  - Size, complexity, and quality of the application involved in the change.
  - Quantitative characteristics of the change: number of files involved, increase or decrease in size, quality and complexity, etc.
  - Criticality of the requested corrective change and urgency of the requested enhancement.

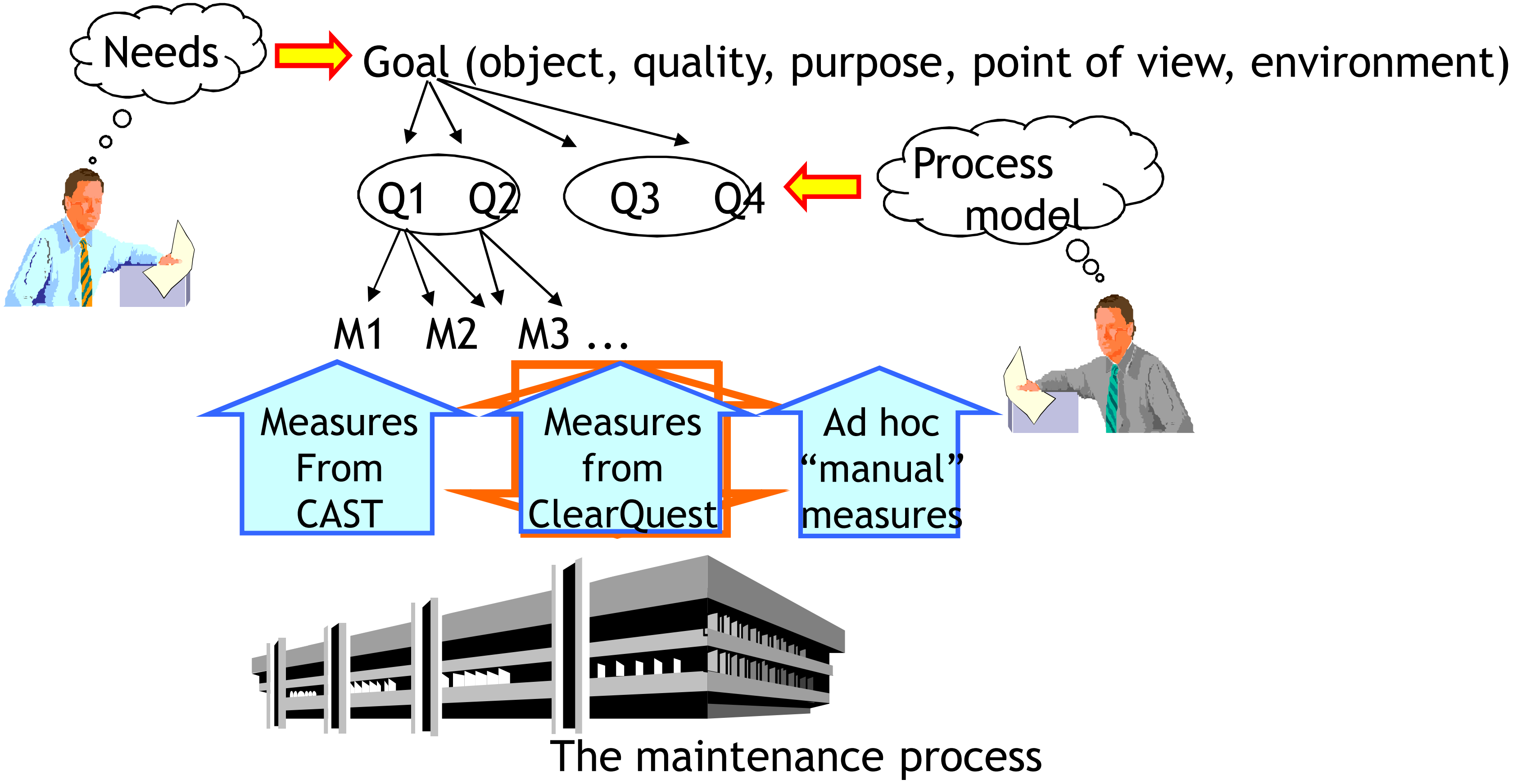
Problems with the available data

- The data ware not available at the required granularity level.
  - the metrics of the GQM plan were intended to capture the characteristics of each CR.
  - the available data concerned versions that were “separated” by tens, or –for some applications– hundreds of CRs.
- It was not possible to retrieve the correspondence between every CR and the code modified in the execution of the request. I.e., it was not possible to determine which source files had been affected by the CR.
- Some fields in ClearQuest records were not regularly or consistently compiled. In particular, the indications concerning the estimated and actual effort required to manage a CR were often lacking or imprecise.
- Some subjective metrics were not collected, because the person that had to support the GQM team was too busy in her regular work to be able to dedicate enough time to the measurement activities.

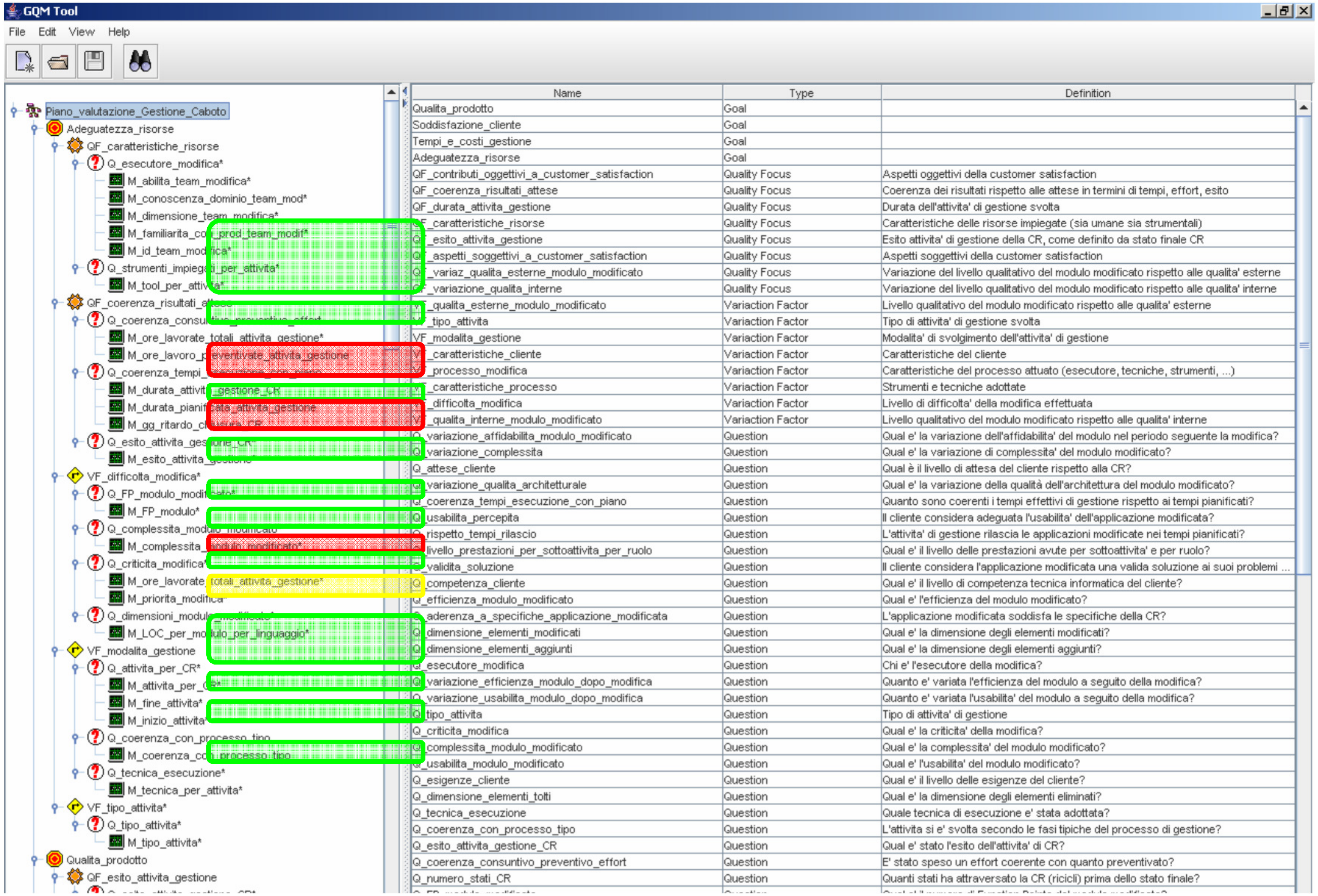
The ideal process



The actual process in presence of constraints



Feasibility of metrics



- Piano_valutazione_Gestione_Caboto
  - Adeguatezza_risorse
    - QF_caratteristiche_risorse
      - Q_esecutore_modifica*
        - M_abilita_team_modifica*
        - M_conoscenza_dominio_team_mod*
        - M_dimensione_team_modifica*
        - M_familiarita_con_prod_team_modif*
        - M_id_team_modifica*
      - Q_strumenti_impiegati_per_attivita*
        - M_tool_per_attivita*
    - QF_coerenza_risultati_attese
      - Q_coerenza_consuntivo_preventivo_effort
        - M_ore_lavorate_totali_attivita_gestione*
        - M_ore_lavoro_preventivate_attivita_gestione
      - Q_coerenza_tempi_esecuzione_con_piano
        - M_durata_attivita_gestione_CR
        - M_durata_pianificata_attivita_gestione
        - M_gg_ritardo_chiusura_CR
      - Q_esito_attivita_gestione_CR*
        - M_esito_attivita_gestione*
    - VF_difficolta_modifica*
      - Q_FP_modulo_modificato*
        - M_FP_modulo*
      - Q_complessita_modulo_modificato*
        - M_complessita_modulo_modificato*
      - Q_criticita_modifica*
        - M_ore_lavorate_totali_attivita_gestione*
        - M_priorita_modifica*
      - Q_dimensioni_modulo_modificato*
        - M_LOC_per_modulo_per_linguaggio*
    - VF_modalita_gestione
      - Q_attivita_per_CR*
        - M_attivita_per_CR*
        - M_fine_attivita*
        - M_inizio_attivita*
      - Q_coerenza_con_processo_tipo
        - M_coerenza_con_processo_tipo
      - Q_tecnica_esecuzione*
        - M_tecnica_per_attivita*
    - VF_tipo_attivita*
      - Q_tipo_attivita*
        - M_tipo_attivita*
  - Qualita_prodotto
    - QF_esito_attivita_gestione
      - Q_esito_attivita_gestione_CR*

Name	Type	Definition
Qualita_prodotto	Goal	
Soddisfazione_cliente	Goal	
Tempi_e_costi_gestione	Goal	
Adeguatezza_risorse	Goal	
QF_contributi_oggettivi_a_customer_satisfaction	Quality Focus	Aspetti oggettivi della customer satisfaction
QF_coerenza_risultati_attese	Quality Focus	Coerenza dei risultati rispetto alle attese in termini di tempi, effort, esito
QF_durata_attivita_gestione	Quality Focus	Durata dell'attivita' di gestione svolta
QF_caratteristiche_risorse	Quality Focus	Caratteristiche delle risorse impiegate (sia umane sia strumentali)
QF_esito_attivita_gestione	Quality Focus	Esito attivita' di gestione della CR, come definito da stato finale CR
QF_aspetti_soggettivi_a_customer_satisfaction	Quality Focus	Aspetti soggettivi della customer satisfaction
QF_variaz_qualita_esterne_modulo_modificato	Quality Focus	Variazione del livello qualitativo del modulo modificato rispetto alle qualita' esterne
QF_variazione_qualita_interne	Quality Focus	Variazione del livello qualitativo del modulo modificato rispetto alle qualita' interne
VF_qualita_esterne_modulo_modificato	Variaction Factor	Livello qualitativo del modulo modificato rispetto alle qualita' esterne
VF_tipo_attivita	Variaction Factor	Tipo di attivita' di gestione svolta
VF_modalita_gestione	Variaction Factor	Modalita' di svolgimento dell'attivita' di gestione
VF_caratteristiche_cliente	Variaction Factor	Caratteristiche del cliente
VF_processo_modifica	Variaction Factor	Caratteristiche del processo attuato (esecutore, tecniche, strumenti, ...)
VF_caratteristiche_processo	Variaction Factor	Strumenti e tecniche adottate
VF_difficolta_modifica	Variaction Factor	Livello di difficolta' della modifica effettuata
VF_qualita_interne_modulo_modificato	Variaction Factor	Livello qualitativo del modulo modificato rispetto alle qualita' interne
Q_variazione_affidabilita_modulo_modificato	Question	Qual e' la variazione dell'affidabilita' del modulo nel periodo seguente la modifica?
Q_variazione_complessita	Question	Qual e' la variazione di complessita' del modulo modificato?
Q_attese_cliente	Question	Qual è il livello di attesa del cliente rispetto alla CR?
Q_variazione_qualita_architetturale	Question	Qual e' la variazione della qualità dell'architettura del modulo modificato?
Q_coerenza_tempi_esecuzione_con_piano	Question	Quanto sono coerenti i tempi effettivi di gestione rispetto ai tempi pianificati?
Q_usabilita_percepita	Question	Il cliente considera adeguata l'usabilita' dell'applicazione modificata?
Q_rispetto_tempi_rilascio	Question	L'attivita' di gestione rilascia le applicazioni modificate nei tempi pianificati?
Q_livello_prestazioni_per_sottoattivita_per_ruolo	Question	Qual e' il livello delle prestazioni avute per sottoattivita' e per ruolo?
Q_validita_soluzione	Question	Il cliente considera l'applicazione modificata una valida soluzione ai suoi problemi ...
Q_competenza_cliente	Question	Qual e' il livello di competenza tecnica informatica del cliente?
Q_efficienza_modulo_modificato	Question	Qual e' l'efficienza del modulo modificato?
Q_aderenza_a_specifiche_applicazione_modificata	Question	L'applicazione modificata soddisfa le specifiche della CR?
Q_dimensione_elementi_modificati	Question	Qual e' la dimensione degli elementi modificati?
Q_dimensione_elementi_aggiunti	Question	Qual e' la dimensione degli elementi aggiunti?
Q_esecutore_modifica	Question	Chi e' l'esecutore della modifica?
Q_variazione_efficienza_modulo_dopo_modifica	Question	Quanto e' variata l'efficienza del modulo a seguito della modifica?
Q_variazione_usabilita_modulo_dopo_modifica	Question	Quanto e' variata l'usabilita' del modulo a seguito della modifica?
Q_tipo_attivita	Question	Tipo di attivita' di gestione
Q_criticita_modifica	Question	Qual e' la criticita' della modifica?
Q_complessita_modulo_modificato	Question	Qual e' la complessita' del modulo modificato?
Q_usabilita_modulo_modificato	Question	Qual e' l'usabilita' del modulo modificato?
Q_esigenze_cliente	Question	Qual e' il livello delle esigenze del cliente?
Q_dimensione_elementi_tolti	Question	Qual e' la dimensione degli elementi eliminati?
Q_tecnica_esecuzione	Question	Quale tecnica di esecuzione e' stata adottata?
Q_coerenza_con_processo_tipo	Question	L'attivita si e' svolta secondo le fasi tipiche del processo di gestione?
Q_esito_attivita_gestione_CR	Question	Qual e' stato l'esito dell'attivita' di CR?
Q_coerenza_consuntivo_preventivo_effort	Question	E' stato speso un effort coerente con quanto preventivato?
Q_numero_stati_CR	Question	Quanti stati ha attraversato la CR (ricicli) prima dello stato finale?
Q_FP_modulo_modificato	Question	Qual e' il numero di Function Points del modulo modificato?

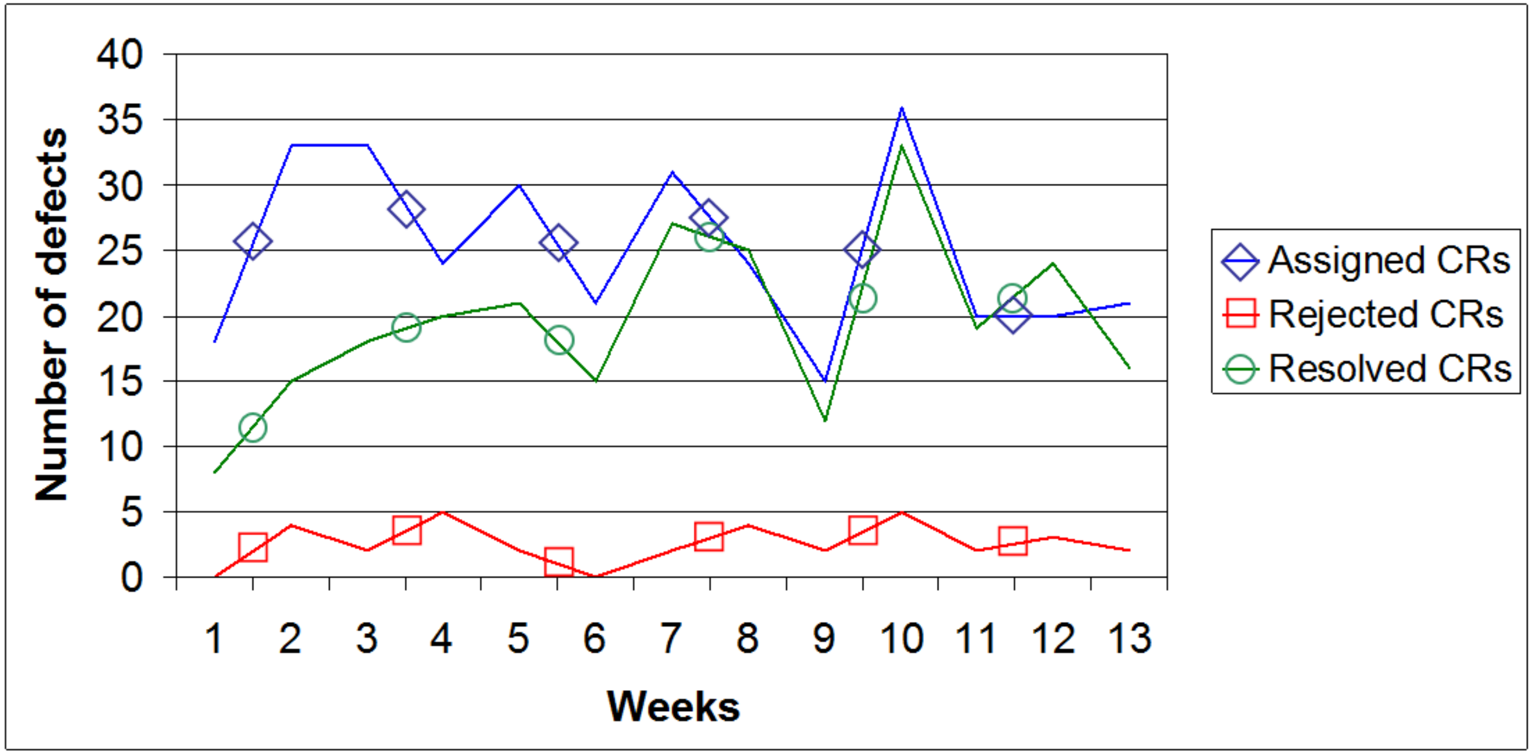
Results

- The goals were all partially fulfilled.
- The results of the measurement process were presented to the top management of the company. They appreciated the work done and were particularly satisfied with the method employed, also because the measurement and analysis process and toolset are reusable in the context of future, more extensive and accurate measurement campaigns

CR per application and type

Application	Number of defect CRs	Rejected defect corrections	Defect corrections rejected more than once	Number of enhancement CRs	Rejected enhancements	Enhancements rejected more than once
APP1	28	2	0	119	7	1
APP2	3	0	0	4	0	0
APP3	6	0	0	53	1	0
APP4	32	1	0	51	2	2
APP5	1	0	0	69	0	0
APP6	2	0	0	5	0	0
APP7	42	1	1	37	0	0
APP8	0	0	0	5	0	0
APP9	2	0	0	1	0	0
APP10	5	0	0	8	0	0
APP11	3	0	0	2	0	0
APP12	9	0	0	5	0	0
APP13	87	0	0	64	0	0
APP14	148	19	6	268	37	12

Number of defect per state in time (CRs)



Lessons learned

- Tools (including development tools not specifically conceived for supporting measurement) can provide useful metrics.
  - Data provided by tools –with some integration– can be sufficiently numerous and rich to support a whole measurement programme.
  - Interestingly, tools provided the needed data in a quite non intrusive way.

Lessons learned

- When selecting measurement tools, the possibility of exporting measures should be taken into due account.
  - It was easier to extract data from a problem tracking tool than from a measurement tool!
- The GQM tool was useful in organizing and documenting effectively the plan, and in supporting the identification of data unavailability and the evaluation of the consequences.
  - The visibility “at a glance” of the plan, combined with the rigorous description of the GQM elements, greatly eased the task of revising the plan.

Lessons learned

- The GQM can provide a measurement framework that is useful even in presence of constraints that prevent several metrics from being collected.
  - The revision of the plan according to the data restrictions proceeded in a bottom-up fashion, as the decisions at the conceptual (goal/question) level were performed taking into account the situation at the operating (metrics/data) level.
- The GQM performed well even in difficult and unprecedented operating conditions.

Conclusion

- The GQM is a method to
  - Understand what data you need
  - Reason on how you can retrieve the needed data
  - Reorganize plans in case the needed data cannot be collected in time and within the available budget (or are not observable altogether)
  - Connect clearly data to business objectives, thus providing basic guideline to the usage of data

Bibliography (1)

- V. Basili, H.D. Rombach "The TAME project: towards improvement-oriented software environments" IEEE TSE, June 1988
- M. Oivo, V. Basili "Representing software engineering models: the TAME goal oriented approach" IEEE TSE, October 1992
- H.D. Rombach "Practical benefits of goal-oriented measurement" in Software reliability and metrics, Fenton & Littlewood Eds., Elsevier 1991
- R. van Solingen, E. Berghout "The Goal/Question/Metric Method", Mc Graw-Hill, 1999
  - https://www.researchgate.net/publication/243765439_The_GoalQuestionMetric_Method_A_Practical_Guide_for_Quality_Improvement_of_Software_Development/link/562f4a9108ae04c2aeb6f7ab/download

Bibliography (2)

- A. Fuggetta, L. Lavazza, S. Morasca, S. Cinti, G. Oldano, E. Orazi, “Applying G/Q/M in an Industrial Software Factory”, ACM Transactions on Software Engineering and Methodology, vol. 7, n. 4, October 1998.
- L. Lavazza, “Providing automated support for the GQM measurement process”, IEEE Software, vol. 17, n. 3, May-June 2000.
- L. Lavazza e G. Barresi, “Automated Support for Process-aware Definition and Execution of Measurement Plans”, International Conference on Software Engineering ICSE 2005, St. Louis, Missouri, May 2005.
- C. Gresse, B. Hoisl, and J. Wuest. “A Process Model for GQM-based Measurement.” Technical Rep. STTI-05-94-E, Univ. Kaiserslautern, 1995.

Thanks for your attention!

QUESTIONS?