INF5180: Software Product- and Process Improvement in Systems Development

Part 07:

Goal-Oriented Measurement

Dr. Dietmar Pfahl

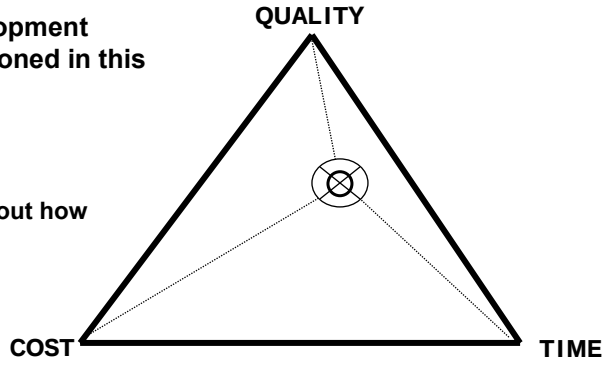
email: dietmarp@ifi.uio.no

Spring 2010

What should we measure?

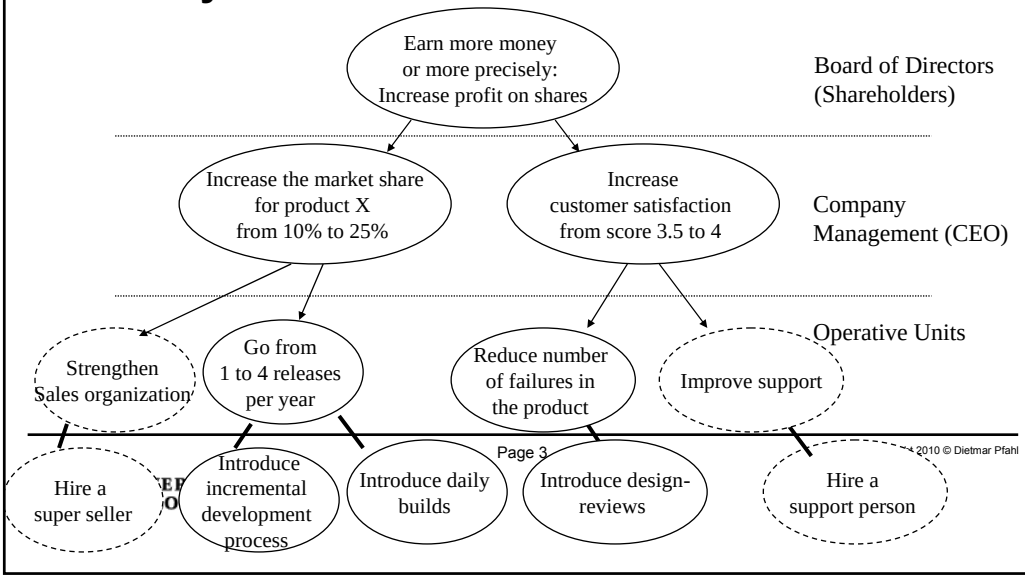
- **Where is my development organization positioned in this triangle?**

Example:
Quality > Time > Costs
This says something about how we prioritize or it may give hints for improvement potential/ objectives.



Recall: The three dimensions are partly conflicting but also partly reinforcing. (cf. Raytheon)

Hierarchy of Goals



Business Focus on Quality

Typical Quality-related Goals

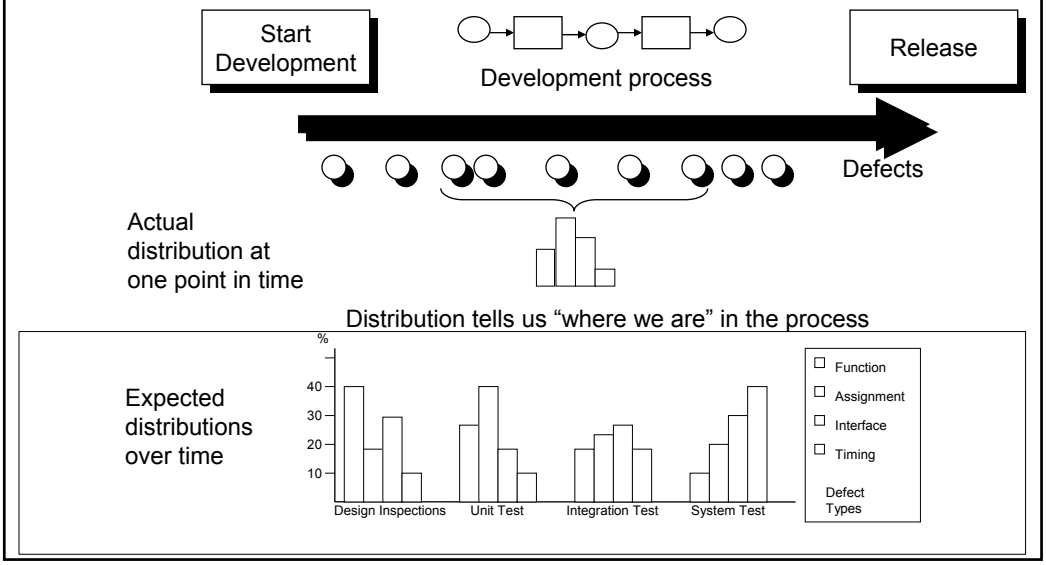
- Reduce number of failures in field (i.e., at customer's site)
  - by reducing number of faults in product
  - by abolishing error triggers
  - → has product, process, and people aspects
- Characterise quality
  - → this is often the starting point

  (see process-related example on next slide)

Typical changes in focus of interest:

- Introduce/alter verification techniques (e.g., inspections) or validation techniques (e.g., new test techniques)
  - to detect more defects (earlier)
- Establish/reorganize quality management
  - to improve defect data collection, storage, analysis, and maintenance
- Introduce better design techniques
  - to reduce possibilities of committing errors
  - to improve readability/testability of artefacts
- Intensify training
  - to reduce the probability of committing errors

Business Focus on Quality – Example



Business Focus on Cost … and Time

Typical Cost-related Goals

- Identify cost divers
- Decrease effort
  - by increasing productivity

Typical changes in focus of interest

- New methods (e.g., perspective based reading)
- Design for reuse
- Introduce component-based development (COTS)
- Outsourcing

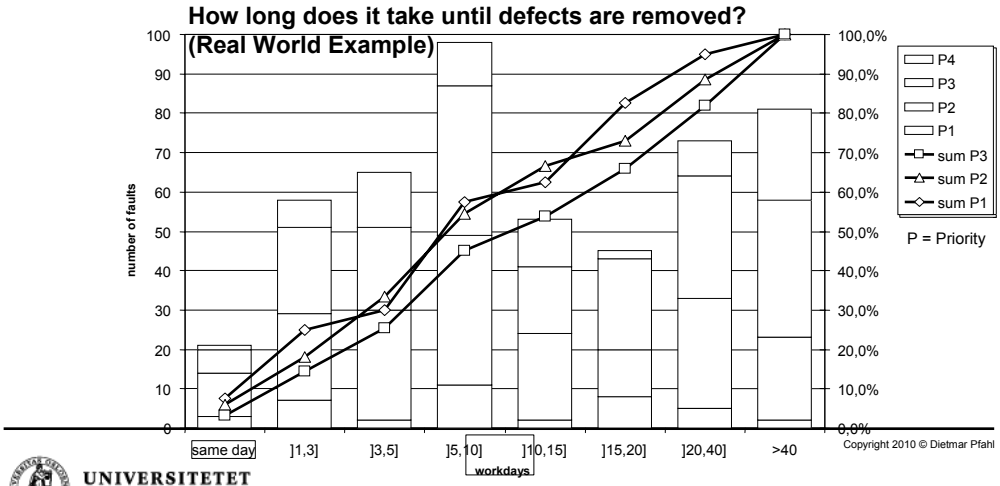
Typical Time-related Goals

- Reduce Time to Market
  - by increasing efficiency

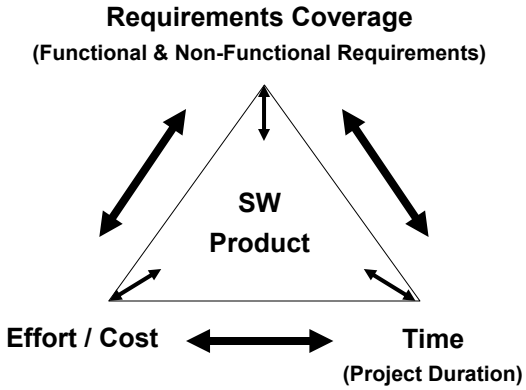
Typical changes in focus of interest

- Product-line development
- Parallel development (concurrent engineering)
- Evaluation of new methods, tools or techniques

Business Focus on Time – Example



Modelling Quality/Cost/Time Trade-Offs



- Measurement is a pre-requisite for developing models that capture trade-offs
  - Holistic models that capture the interaction between the three dimensions of the "Magic Triangle"

    Process Simulators capture Quality/Cost/Time Trade-Offs
  - Example of a model that captures Cost/Time trade-offs: COCOMO II

Goal-Oriented Measurement – Why?

- **Typical problems encountered when performing measurement programs:**
  - Unnecessary data is collected (→ data cemeteries)
  - Inadequate data is collected (→ useless data)
  - Collected data is not used properly (e.g., misused for evaluation of people)
  - People don't know/understand the goals and are not involved in the interpretation of analysis results

- **Experience shows that:**
  - Usefulness of measures cannot be judged out of context
  - There is no standard recommended set of measures for all contexts
  - Measures have to be chosen, customized, and used according to goals of interests and the context/environment

Goal-Oriented Measurement – Benefits

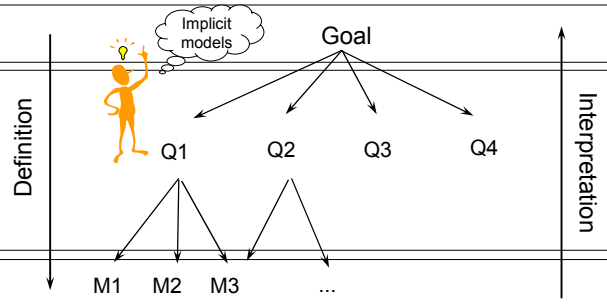
- **Goal-oriented measurement …**
  - is an approach for defining and using software measures to achieve predefined measurement goals (→ explicit, focused, and in context)
- **Goal-oriented measurement supports …**
  - structured (and rational) discussions about measurement
  - adequacy, consistency and completeness checking of data collection and of data
  - management of the complexity (and costs) of measurement programs
- The **Goal/Question/Metric (GQM)** method is a widely used approach to Goal-Oriented Measurement (at least in mature organizations)

GQM Principles

1. **Goal-Driven:** Define measurement goals (systematically).
2. **Documented:** Document measurement goals and their refinement explicitly.
3. **People-Oriented:** Actively involve all participants during the entire measurement program.
4. **Context-Sensitive:** Consider context/environment when defining measurement goals.
5. **Top-Down:** Refine goals top-down into measures via questions.
6. **Bottom-Up:** Analyze and interpret the collected data bottom-up in the context of the goal.
7. **Sustained:** Measurement is part of a systematic and continuous software quality improvement process.

GQM Core Elements

GQM has four elements:



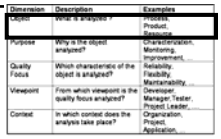
- Goals
- Questions
- Models (are associated with Questions)
- Measures

GQM Core Elements: Goals

- GQM goal (or: Measurement Goals) are derived from business or improvement goals
- A GQM goal defines which object is measured, for which purpose, with respect to which quality aspect, from which viewpoint, and in which environment (or context).
- GQM Goal Template

Dimension	Description	Examples
Object	What is analyzed ?	Process, Product, Resource
Purpose	Why is the object analyzed?	Characterization, Monitoring, Improvement, ...
Quality Focus	Which characteristic of the object is analyzed?	Reliability, Flexibility, Maintainability, ...
Viewpoint	From which viewpoint is the quality focus analyzed?	Developer, Manager,Tester, Project Leader, ...
Context	In which context does the analysis take place?	Organization, Project, Application, ...

GQM Core Elements: Goal – Object



- **Products**:
  - artifacts (documents) produced during system life cycle phases (e.g., specification, design, programs, test suites)
- **Processes**:
  - software related activities (e.g., specifying, designing, coding, testing, inspecting)
- **Resources**:
  - "items" used by processes in order to produce their outputs (e.g., people, hardware, software, office space)

Dimension	Description	Examples
Object	What is analyzed ?	Process, Product, Resource
Purpose	Why is the object analyzed?	Characterization, Monitoring, Improvement
Quality Focus	Which characteristic of the object is analyzed?	Reliability, Flexibility, Maintainability, ...
Viewpoint	From which viewpoint is the quality focus analyzed?	Developer, Manager, Tester, Project Leader, ...
Context	In which context does the analysis take place?	Organization, Project, Application, ...

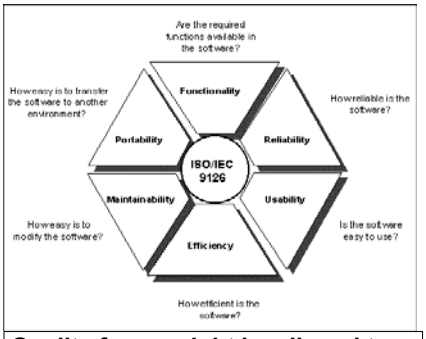
GQM Core Elements: Goal – Purpose

- ***Characterization:***
  - aims at forming a snapshot of the current state/performance of the software development processes and products
- ***Monitoring:***
  - aims at following the trends/evolution of the performance/state of processes and products
- ***Evaluation:***
  - aims at comparing and assessing the quality of products and the efficiency/effectiveness of processes
- ***Prediction:***
  - aims at identifying relationships between various process and product factors and using these relationships to predict relevant external attributes of products and processes
- ***Control* and *Change:***
  - aim at identifying causal relationships that influence the state/performance of processes and products
    - *Control* consists in influencing the course of a project in order to alleviate risks.
    - *Change* implies modifying the process from project to project in order to improve quality or productivity.
    - *Change* requires a finer grained understanding of the phenomena under study than *control*.

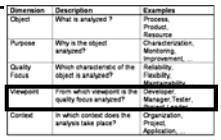
Dimension	Description	Examples
Object	What is analyzed ?	Process, Product, Resource
Purpose	Why is the object analyzed?	Characterization, Monitoring, Improvement
Quality Focus	Which characteristic of the object is analyzed?	Reliability, Flexibility, Maintainability, ...
Viewpoint	From which viewpoint is the quality focus analyzed?	Developer, Manager, Tester, Project Leader, ...
Context	In which context does the analysis take place?	Organization, Project, Application, ...

GQM Core Elements: Goal – Quality Focus

- **Cost**
- **Time-to-Market**
- **Efficiency** ⟷
- **Effectiveness** ⟷
- **Correctness** ⟷
- **Reliability** ⟷
- **Reusability** ⟷
- **Usability** ⟷
- **Maintainability** ⟷
- **…**



Quality focus might be aligned to to standards (e.g. ISO 9126)



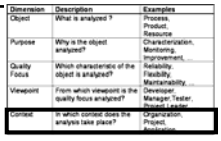
Dimension	Description	Examples
Object	What is analyzed ?	Process, Product, Resource
Purpose	Why is the object analyzed?	Characterization, Monitoring, Improvement, ...
Quality Focus	Which characteristic of the object is analyzed?	Reliability, Flexibility, Maintainability, ...
Viewpoint	From which viewpoint is the quality focus analyzed?	Developer, Manager, Tester, Project Leader, ...
Context	In which context does the analysis take place?	Organization, Project, Application, ...

GQM Core Elements: Viewpoint

Defines the stakeholder who is interested in the measurement results.

Examples:

- **Software Users**
  - interested in the quality and value of the software products
- **Senior Managers**
  - interested in overall understanding, control and improvement across projects in the business unit
- **Project Managers**
  - interested in understanding, control and improvement of the specific software projects they manage
- **Software Engineers**
  - interested in understanding, control and improvement of the specific software project activities and quality of work products in which they are involved
- **Software Process Engineers / Quality Assurance Team**
  - interested in a cross section of what the four previous audiences are interested in



Dimension	Description	Examples
Object	What is analyzed ?	Process, Product, Resource
Purpose	Why is the object analyzed?	Characterization, Monitoring, Improvement, ...
Quality Focus	Which characteristic of the object is analyzed?	Reliability, Flexibility, Maintainability, ...
Viewpoint	From which viewpoint is the quality focus analyzed?	Developer, Manager, Tester, Project Leader, ...
Context	In which context does the analysis take place?	Organization, Project, Application, ...

GQM Core Elements: Context

Defines the environment in which the measurement project takes place.

Is important for

- *assessing generalisability (external validity)*
- *future re-use of plans, measurements, and models*

Examples:

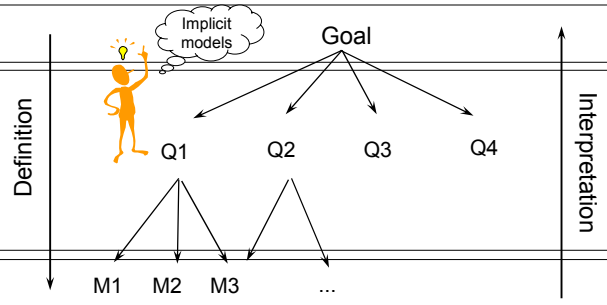
- **Organization**
  - **Company, Business Unit, Department, Project, etc.**
- **Type of Product**
  - **Business Application, MIS, Embedded System, etc.**
- **Product Domain**
  - **Telecommunication, Transportation Systems, Commerce (banks, insurance companies), medical health care systems, etc.**
- **Other**
  - **Development history**
  - **Organizational maturity**
  - **Platforms / Technologies used, etc. …**

GQM Core Elements: Goal – Example

Analyze	the test process
for the purpose of	characterization
with respect to (quality aspect)	effectiveness
from the viewpoint of the	test team
in the environment of	project X, organization Y.

GQM Core Elements

GQM has four elements:



- Goals
- Questions
- Models (are associated with Questions)
- Measures

GQM Core Elements: Questions & Models

- ***Questions:***
  - **Specify verbally the information required to achieve the goal**
- ***Models:***
  - **Specify formally (and make operational) the information required to achieve the goal**
  - **Type of model depends on goal purpose**
  - **Models are sometimes called Indicators**

Dimension	Description	Examples
Object	What is analyzed ?	Process, Product, Resource
Purpose	Why is the object analyzed?	Characterization, Monitoring, Improvement, ...
Quality Focus	Which characteristic of the object is analyzed?	Effectiveness, Flexibility, Maintainability, ...
Viewpoint	From which viewpoint is the quality focus analyzed?	Developer, Manager, Tester, Project Leader, ...
Context	In which context does the analysis take place?	Organization, Project, Application, ...


UNIVERSITETET I OSLO

GQM Core Elements: Question – Examples

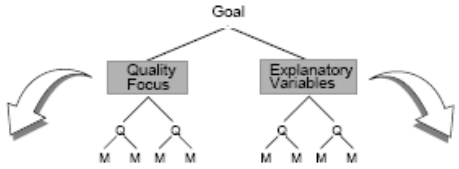
Dimension	Description	Examples
Object	What is analyzed ?	Process, **Test** Product, Resource
Purpose	Why is the object analyzed?	Characterization, Monitoring, Improvement, ...
Quality Focus	Which characteristic of the object is analyzed?	Effectiveness, Flexibility, Maintainability, ...
Viewpoint	From which viewpoint is the quality focus analyzed?	Developer, Manager, Tester, Project Leader, ...
Context	In which context does the analysis take place?	Organization, Project, Application, ...

- **<u>Goal:</u> Analyze the *test process* for the purpose of *characterization* with respect to (quality aspect) *effectiveness* from the viewpoint of the test team in the environment of project X, organization Y.**
- **Question 1: How many failures are detected during testing?**
- **Question 2: When are failures detected (time)?**
- **Question 3: What types of failures are detected?**
- **Question 4: How much testing effort is spent?**
- **Question 5: Which test techniques/tools are applied?**
- **Etc.**


UNIVERSITETET I OSLO

GQM Core Elements: Question Categories

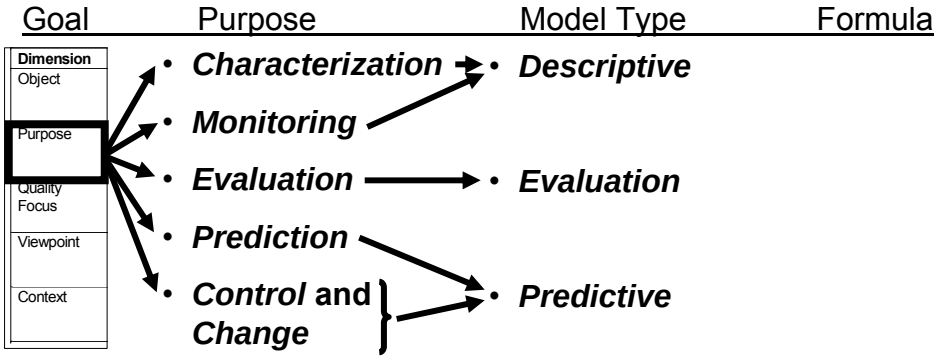
- **Goal:** **Analyze the *test process* for the purpose of *characterization* with respect to (quality aspect) *effectiveness* from the viewpoint of the test team in the environment of project X, organization Y.**
- **In order to help formulate appropriate questions, the goal is refined into two aspects:**
  - **Quality focus variables: Characterize quality focus defined by the GQM goal**
  - **Explanatory variables (or: variation factors): specify parameters that may have an impact on the quality focus: e.g., experience of testers, used test techniques/tools**
- **Questions may be generated for each of the two aspects**



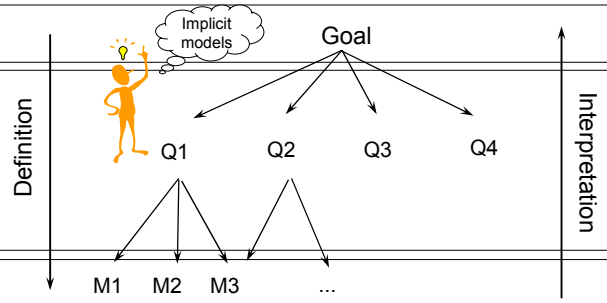
- **Question 1: How many failures are detected during testing?**
- **Question 2: When are failures detected (time)?**
- **Question 3: How are failures distributed wrt. criticality?**

- **Question 4: How much testing effort is spent?**
- **Question 5: Which test techniques/tools are applied?**

GQM Core Elements: Model Type ↔ Purpose



GQM Core Elements



GQM has four elements:

- Goals
- Questions
- Models (are associated with Questions)
- Measures

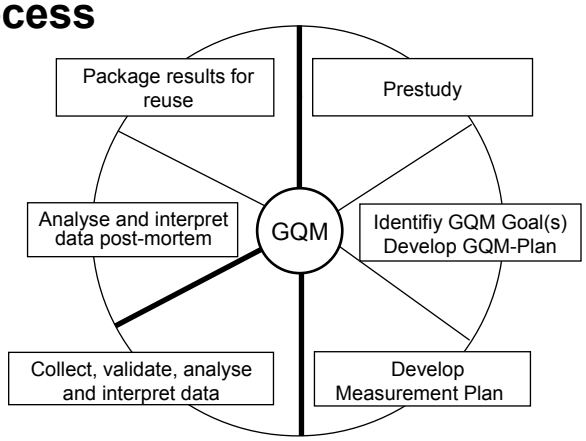
GQM Core Elements: Measure – Example

- Q3: What is the distribution of failures reported during test by criticality?
  - Model refines to …
  - M3.1: Criticality classification
    - scale: nominal
    - unit: criticality class
    - range: [critical, uncritical, other]
    - object: reported failure
    - attribute: criticality
- Q5: How experienced are the development team members?
  - Model refines to …
  - M5.1: Experience classification
    - scale: ordinal
    - unit: experience class
    - range: [inexperienced, low (< 5 modules developed), medium (5-10 modules developed), high experience (> 10 modules developed)]
    - objects: development team member
    - attribute: experience

The GQM Process

Phases

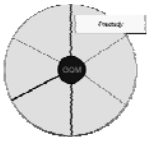
- Planning
- Execution
- Analysis



Planning Phase

Planning Phase – Pre-Study, GQM Planning, and Measurement Planning

- **Step 1 – Pre-Study:**
  - Characterization of and familiarization with organization
  - Selection of pilot projects
  - Motivation and training
- **Step 2 – GQM Planning:**
  - Definition of GQM Goals
  - Definition of GQM Plan (**what** is going to be measured?)
- **Step 3 – Measurement Planning:**
  - The Measurement Plan defines **by whom**, **how**, and **when** data collection is conducted.
  - Includes definition of Data Collection Forms, Analysis Methods, Presentation Diagrams

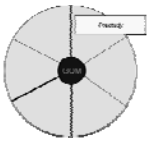


Step 1: Pre-Study – Characterization

- General organizational information

General organizational information (examples):
- Size of organization (number of employees)
- Percentage of software personnel
- Industrial sector(s)
- Product domain and/or types of services
- Certifications, assessments, etc.
- Improvement history
- Availability of process support group/quality assurance group
- Typical project duration (months)
- Typical project effort (staff months)
- Typical size of project group
- Typical product size (e.g., KLOC) ...

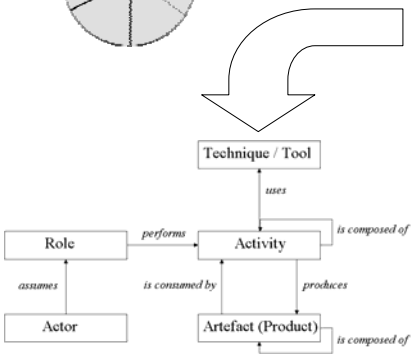
Familiarization with …
- Organizational process models
- Organizational product models
- Organizational quality models
- Organizational data collection practices

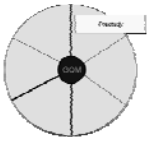


Step 1: Pre-Study – Characterization

Organizational process model:

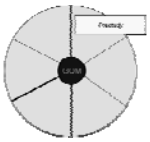
- Documented standard process
- Tools used for
  - Requirements modeling/maintenance
  - Design
  - Coding
  - Testing
  - Project management
  - Configuration management
  - Quality management
  - etc.





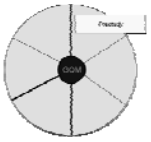
Step 1: Pre-Study – Characterization

- Organizational product model
  - Type of SW produced (e.g., embedded)
  - Use of SW (e.g., with systems, stand-alone)
  - Avg. number of installations at customer sites
  - Constraints (e.g., wrt. hardware used) ...
- Organizational quality model
  - Crucial quality aspects
  - Meaning of these quality aspects ...
- Organizational data collection practices
  - Measurement performed as basis for control and improvement
  - Availability of historical data ...



Step 1: Pre-Study – Pilot Project Selection

- As many as possible from the following selection factors should hold with the pilot project (s):
  - Project should be a mainstream project (i.e. “typical” project)
  - Duration of the software-project should be reasonably short
  - Staff size should be reasonably small
  - Process performance and productivity in the project should be relatively stable
  - The project team should be open-minded with regards to the measurement program
  - The project should not be too ‘risky’
  - The project (its people) should have credibility within the rest of the organization

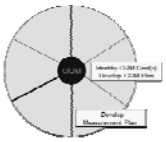


Step 1: Pre-Study – Motivation & Training

- Main goals:
  - Visible management commitment
  - Appropriate view of data confidentiality

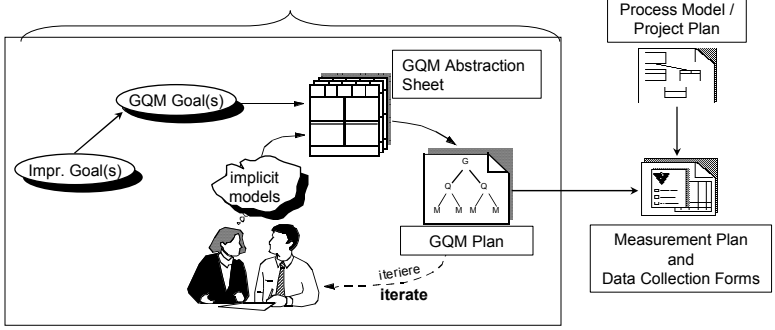
  → important: measurement program must not be used to evaluate people!

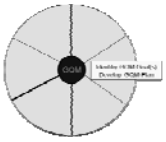
Planning Phase: Steps 2 & 3



Step 3: Measurement Planning

Step 2: GQM Planning

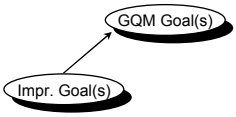




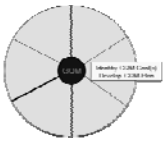
Step 2: GQM Planning

GQM Goal (or Measurement Goal):

- Determines which object is measured, for which purpose, with respect to which quality aspect, from which viewpoint, and in which environment (context)



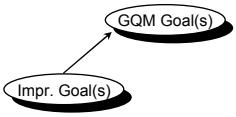
- It is derived from business (improvement) goals:
  - Main source of information: Management (from Senior Management down to Project Management)
  - Elicitation Format: Meeting / Brainstorming Session
- It is defined with the help of the GQM Goal Template

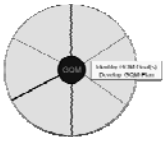


Step 2: GQM Planning

GQM Goal Template (with example):

Dimension	Description	Examples
Object	Analyze the	qual. assurance process
Purpose	for the purpose of	characterization
Quality Focus	with respect to	effectiveness
Viewpoint	from the viewpoint of the	software dev. team
Context	In the following environment	company X

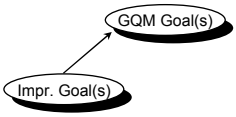


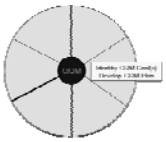


Step 2: GQM Planning

Goal Definition Considerations / Guidelines

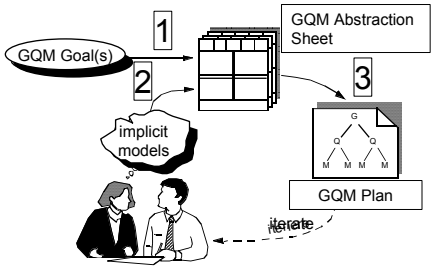
- Keep cost low
- Make sure everybody who participates benefits from the measurement program
- Start small / be focused
  - Not too many goals
  - Small number of people involved (stakeholders as well as engineers)
- Consider maturity of organization
  - Documentation & stability of processes
  - Measurement practices in place
- Start with characterization goals / prediction is more difficult / control is most difficult



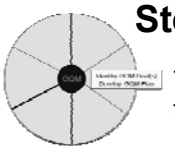


Step 2: GQM Planning

• Development of GQM Plan



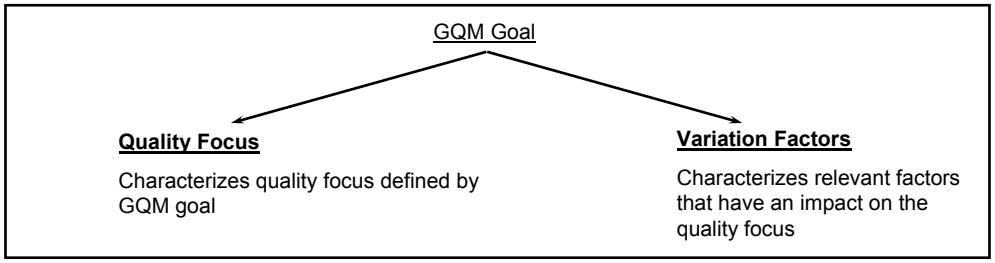
1) Refining GQM Goals
2) Involving Experts
3) Developing the GQM Hierarchy

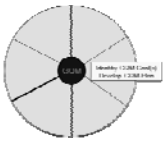


Step 2: GQM Planning – Refining GQM Goals

The GQM goal is refined into questions with respect to 1) the quality focus and 2) the factors influencing the quality focus.

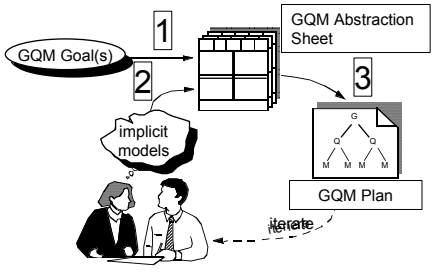
Important: To each question, a hypothesis about the actual value is provided by the experts.





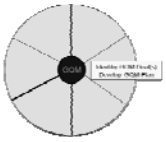
Step 2: GQM Planning

• Development of GQM Plan



1) Refining GQM Goals
2) Involving Experts
3) Developing the GQM Hierarchy

Step 2: GQM Planning – Involving GQM Experts



Goal ▶	object	purpose	quality focus	viewpoint	context
Quality Focus Which factors define the quality focus?			Explanatory Variables Which variables have an impact on the quality focus?		
Baseline Hypothesis What is the current expectation wrt. the quality focus?			Impact on Baseline Hypothesis How do the explanatory variables influence the quality focus?		

Purpose of expert involvement:

- Helps identifying questions that refine the goal
- Identify viewpoint's intuition (e.g., quality models) wrt. the measurement goal
- Reflect the understanding of the viewpoints
- Important tool: Abstraction Sheet

How to fill in Abstraction Sheets?

- Fill in *Quality Focus*
  - Formulate questions which concern the focus area
  - (Avoid environment factors)
- Fill in *Variation Factors*
  - Formulate questions which concern the environment and which are supposed to have influence on the quality focus

  For "characterisation" goals:
  - This will often be something we cannot control and which we – at this time – will not try to improve
  - This quadrant serves mainly to help interpret the results
- Fill in *Baseline Hypotheses*
  - Provide (expected) answers to all questions related to the Quality Focus
- Fill in *Impact Hypotheses*
  - Connects Variation Factors with Quality Focus
  - Try to cover all variation factors

Goal ▶	object	purpose	quality focus	viewpoint	context
Quality Focus Which factors define the quality focus?			Explanatory Variables Which variables have an impact on the quality focus?		
Baseline Hypothesis What is the current expectation wrt. the quality focus?			Impact on Baseline Hypothesis How do the explanatory variables influence the quality focus?		

Example of a GQM Abstraction Sheet

	Object	Purpose	Quality focus	Viewpoint	Context
Goal	QA process	characterization	effectiveness	sw development team	company y project x

Quality Focus

Failures (total number)
- by criticality

Faults (total number)
- by life cycle phase of detection
- distribution of fault types
- relation between faults and failures

Variation Factors

Experience of development team members

Conformance to inspection process

Degree of code reuse

Baseline Hypothesis

Failures (total number): 120
(before delivery)
- by criticality: 5% critical, 15%uncritical, 80% other

Faults (total number): 200
- per phase of detection: 10% requirements, ...
- ...

Impact on Baseline Hypothesis

Experience of development team members
- more experienced development team members -> smaller number of faults and failures

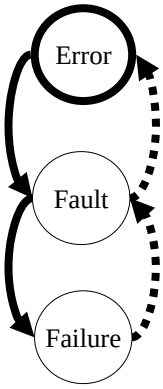
Conformance to inspection process
- good adherence -> smaller number of failures detected during test phase
- ...

NB:
The following is not part of the GQM-Method!

It is meant to illustrate through an example how Measurement relates to Software Process Improvement

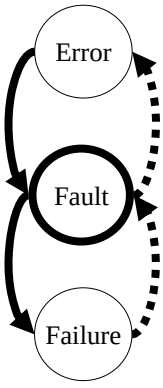
...

Definitions: Failure – Fault – Error /1



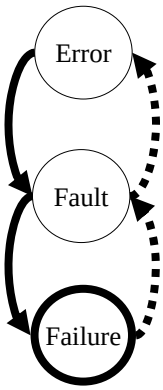
- **Error**
  - A human action that produces an incorrect result. [IEEE Std 610.12-1990]
  - In software development: Mistake made by the developer who injected a fault into the code due to cognitive bias or other reasons (e.g., misinterpretation of a design document, wrong usage of programming language)

Definitions: Failure – Fault – Error /2



- **Fault (or defect or "bug")**
  - An abnormal condition or defect at the component, equipment, or sub-system level.
  - A manifestation of an error in software.
  - A fault, if encountered may lead to a failure.

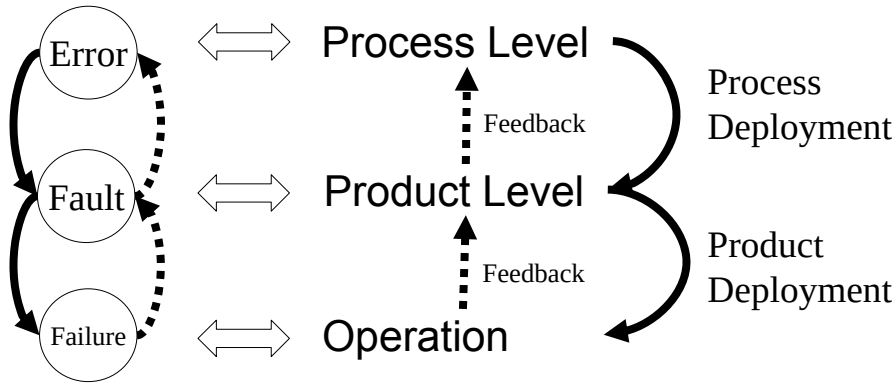
Definitions: Failure – Fault – Error /3



- **Failure**
  - Observed deviation of the software from its expected delivery or service.
  - Formal Definition [ISO/CD 10303-226]: The lack of ability of a component, equipment, sub system, or system to perform its intended function as designed. Failure may be the result of one or many faults.

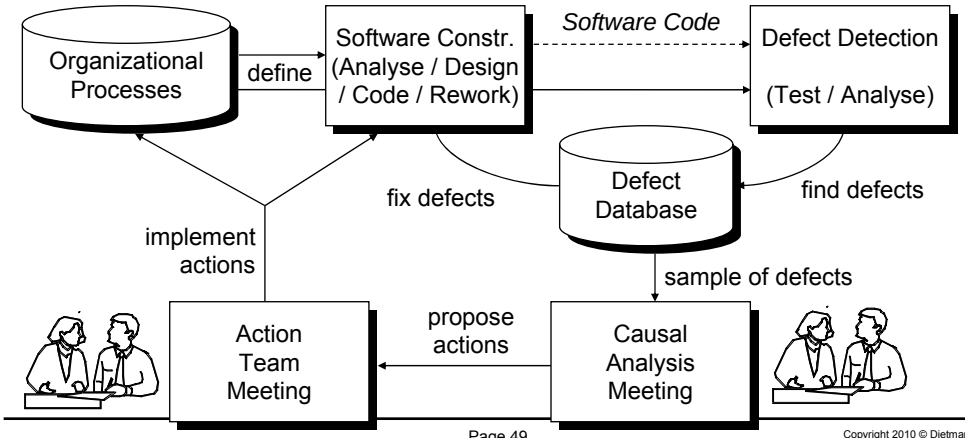
UNIVERSITETET I OSLO

Definitions: Failure – Fault – Error /4



UNIVERSITETET I OSLO

Defect Causal Analysis (DCA)



Causal Analysis Meeting

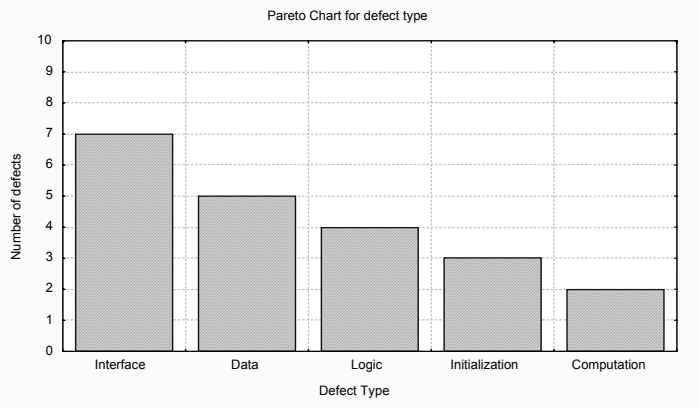
Causal Analysis Meeting

- Purpose: Developers analyze problems and recommend improvements at regular intervals

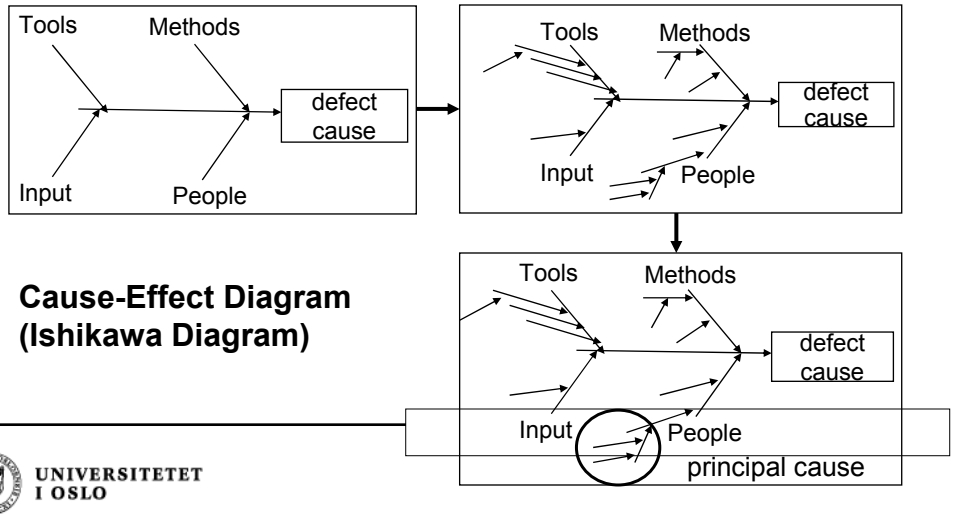
1. Select sample of defects
   → less than 20 representative defects
2. Classify selected defects
   → when inserted, when detected, how fixed
3. Identify systematic errors
   → an error that results in similar defects
4. Determine principal cause
   → most important factor contributing to systematic error
5. Develop action proposals
   → prevent or detect earlier the systematic defect
6. Document meeting results

Tool for identifying predominant defects

Pareto Chart



Tool for determining principal defect cause



Cause-Effect Diagram (Ishikawa Diagram)

Action Team Meeting

Action Team Meeting

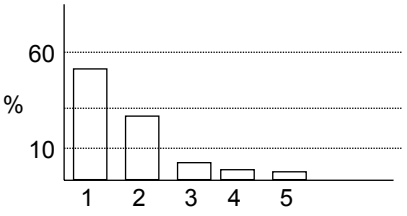
- Purpose: Software Engineering Process Group (SEPG) initiates actions with management support.

1. Prioritize action proposals
   - based on Pareto charts of causes, future development activities, relative ROI of actions
2. Resolve conflicts and combine related proposals
   - necessary for multiple causal analysis teams
3. Establish implementation plan for high-priority items
4. Allocating resources and assigning responsibility for implementation plan
5. Monitor progress of implementation and effectiveness of actions
6. Ensure that success stories are recognized (and successful individuals identified → might be culture dependent)

Mantra:

One implemented action has more value than 10 proposed actions!

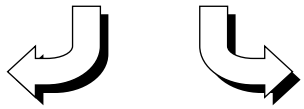
DCA Example /1



Major Defect Type Categories

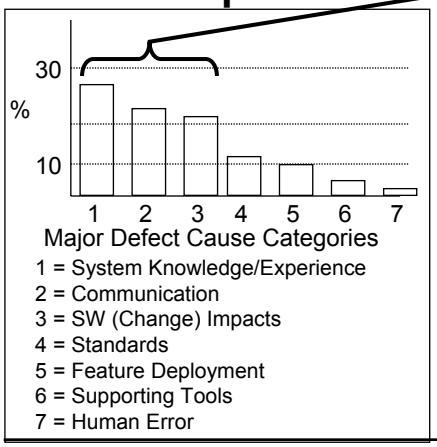
1 = Design Fault
2 = Incompatible Interface
3 = Incorrect Synchronization
4 = Incorrect Object Patch Carryover
5 = System Resource Exhaustion

- Problem: A company wanted to learn about faults introduced when maintaining their product (total size ~2.9 MLOC)
- Action: Apply Defect Causal Analysis for a new version of a large telephone system (new features & bug fixes / changes ~1 MLOC)
- Pareto Chart: During integration test, 200 faults were identified and analyzed



Defect Causes: next slide

DCA Example /2



- Pareto Chart: 80% of faults due to insufficient knowledge/experience, communication, failure to consider all software impacts
- Action
  - Increase experience of maintenance team
    - motivate designers to continue working in maintenance project
    - improve maintenance training and mentor programs
  - Improve communication
    - disciplined maintenance methodologies
    - more thorough documentation
    - communicate info on potential impact of modification
- Impact: Subsequent version had substantially fewer faults


UNIVERSITETET I OSLO

Implementing DCA

Step 1
- Define the DCA process
  - When will causal analyses be conducted?
  - How will causal-analysis teams be organized if more than one is needed?
  - Who are the members of the action team?

Step 2
- Provide training to participants
  - Moderator training
  - Causal-analysis team training
  - Management briefing

Step 3
- Evaluate the DCA process
  - Participant feedback on the DCA process itself
  - Quantitative data on the effects of DCA-originated actions


UNIVERSITETET I OSLO

DCA - Pro's and Con's

Pro's

- Helps improve both quality and productivity of organization
- Provides feedback at any stage of development process
- Helps show developers the value of conforming to process

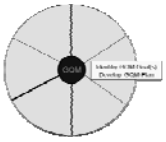
and

Con's

- Requires significant resources (ca. 1.5% of project budget)
- Only a sample (= sub-set) of all defects can be investigated
- Focusing on individual defects may result in less attention to finding solutions addressing a larger scope of problem

... back to GQM ...

Step 2: GQM Planning – Involving GQM Experts



Goal ▶	object	purpose	quality focus	viewpoint	context
Quality Focus Which factors define the quality focus?			Explanatory Variables Which variables have an impact on the quality focus?		
Baseline Hypothesis What is the current expectation wrt. the quality focus?			Impact on Baseline Hypothesis How do the explanatory variables influence the quality focus?		

Purpose of expert involvement:

- Helps identifying questions that refine the goal
- Identify viewpoint's intuition (e.g., quality models) wrt. the measurement goal
- Reflect the understanding of the viewpoints
- Important tool: Abstraction Sheet

Exercise

Situation

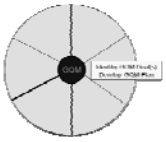
A small company develops a very innovative web-enabled wireless device. It is expected to outclass all existing designs by providing revolutionary special features. The competitors are working on such a product as well. However, the company has a lead of at least one calendar year. The product is developed by a team of 20 highly creative people of which 15 have less than 2 years of experience in software development. Your are the assigned SPI co-ordinator.

Consider the following measurement goal:

- Analyse: **the software development process**
- For the purpose of: **understanding (=characterization)**
- With respect to: **efficiency**
- From the viewpoint of: **software developers**
- In the context of: **your company**

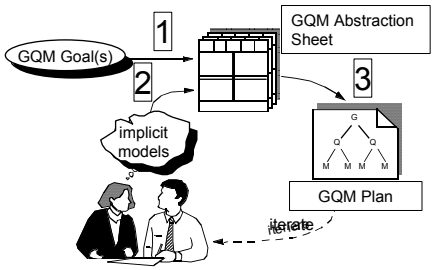
5 minutes

→ ***Your Task: Specify 5 GQM questions to operationalise this goal***



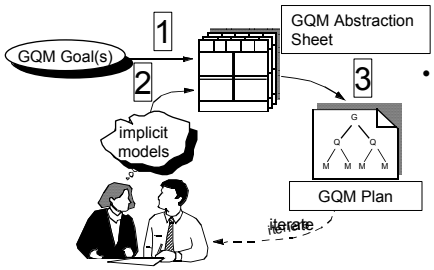
Step 2: GQM Planning

- **Development of GQM Plan**



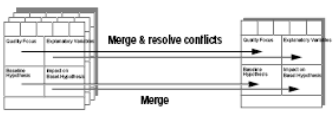
1) Refining GQM Goals
2) Involving Experts
3) Developing the GQM Hierarchy

Step 2: GQM Planning – Developing the GQM Hierarchy

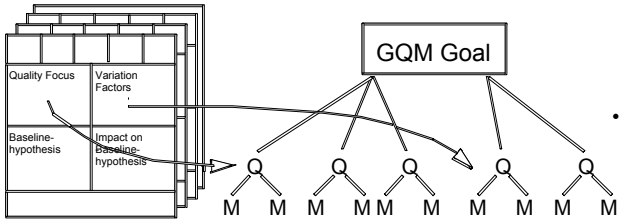


Merging GQM Abstraction Sheets:

- Results from individual interviews are merged into one abstraction sheet
- Conflicts have to be resolved



Step 2: GQM Planning – Developing the GQM Hierarchy



- Map contents of the merged abstraction sheets to GQM plan and define measures:
- Formulate the quality foci and their variation factors in the abstraction sheet as questions and document them in the GQM plan in a structured way
- For every item in the upper quadrants of the abstraction sheet, at least one question should be derived
- Define models and measures based on the questions

GQM Plan

- The models and measures are identified by answering ”What kind of information do we need in order to answer the questions?"
- The GQM-tree is documented in tabular form
- Each measure is defined by:
  - Name, ID
  - Definition (scale, range)
  - Hypotheses

Model

Goal	Question (Model)	Measure
G1	Q1	M1 M2 M3
	Q2	M1 M4 M5

Step 2: GQM Planning – Developing the GQM Hierarchy

Example Questions – Quality Focus:

- Q(uestion)1.1: What is the total number of failures?
  - Model: Sum of failures
  - *Hypothesis: 120*
- Q1.2: What is the distribution of failures by criticality?
  - Model: Sum of failures per criticality class / Total sum of failures
  - *Hypothesis: 5% critical, 15% uncritical, 80% other*
- Q1.3: What is the distribution of failures by detection phase?
  - Model: Sum of failures per detection activity / Total sum of failures
  - *Hypothesis: 5% maintenance, 95% test*

Step 2: GQM Planning – Developing the GQM Hierarchy

Example Questions – Quality Focus (cont'd):

- Q2.1: What is the total number of faults?
  - Model: Sum of faults detected (without duplicates)
  - *Hypothesis: 200*
- Q2.2: What is the distribution of faults by detection phase?
  - Model: Sum of faults per detection activity / Total sum of faults
  - *Hypothesis: 8% REQ, 20% HLD, 20% LLD/IMP, 50% TEST, 2% Maintenance*
- Q2.3: What is the effort distribution for fixing faults, per phase?
  - Model: Sum of fault fix effort per phase / Total fault fix effort
  - *Hypothesis: 5% REQ, 10% HLD, 25% LLD/IMP, 30% TEST, 30% Maintenance*
- ....

Step 2: GQM Planning – Developing the GQM Hierarchy

Example Questions – Variation Factors:

- Q4: What is the experience of the development team members?
  - Model: Average experience of all team members
  - *Hypothesis: The higher the experience of the development team members the smaller the number of faults and failures*
- Q5: How close was adherence to inspection process?
  - Model: Average adherence to inspection process
  - *Hypothesis: The closer adherence to inspection process, the lower the number of failures detected during test*
- ....

Step 2: GQM Planning – Developing the GQM Hierarchy

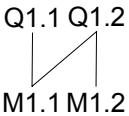
Example Measures – Quality Focus:

- Q(uestion)1.1: What is the total number of failures?
  - M(easure)1.1: Failure (failure reports) count (Scale: absolute; Unit: integer; Range: positive integer; Object: product version 1.0)
- Q(uestion)1.2: What is the distribution of failures by criticality?
  - M1.1, M1.2: Failure Criticality (Scale: nominal, Unit: n/a, Range: {critical, uncritical, other}, Object: failure report)

    where:

    Critical = complete breakdown of the system

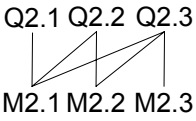
    Uncritical = unable to perform one or more of the functions F1 to F6, but system still running
- Q(uestion)1.3: …



Step 2: GQM Planning – Developing the GQM Hierarchy

Example Measures – Quality Focus (cont'd):

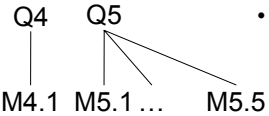
- Q2.1: What is the total number of faults?
  - M(easure)2.1: Fault (fault report) count before delivery (Scale: absolute; Unit: integer; Range: positive integer; Object: product version 1.0)
- Q2.2: What is the distribution of faults by detection phase?
  - M2.1 (as above)
  - M2.2: Life cycle phase (Scale: nominal; Unit: n/a; Range: REQ, HLD, LLD/IMP, TEST, Customer; Object: fault)
- Q2.3: What is the effort distribution for fixing faults, per phase?
  - M2.1, M2.2 (as above)
  - M2.3: Effort for fault fixing – per fault (Scale: ratio; Unit: person-hour; Range: [0, ∞); Object: fault)
- ....



Step 2: GQM Planning – Developing the GQM Hierarchy

Example Measures – Variation Factors:

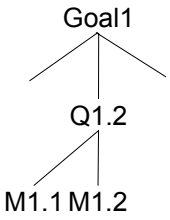
- Q4: What is the experience of the development team members?
  - M4.1: Experience of team member (Scale: ordinal; Unit: n/a; Range: {high = developed more than 10 modules, medium = developed 2 to 9 modules, low = developed less than 2 modules}; Object: Team member)
- Q5: How close was adherence to inspection process?
  - M5.1 / M5.2: Document Count / Type (…)
  - M5.3 / M5.4: Inspection Count /Type (…)
  - M5.5: Adherence to Inspection Process (Scale: ordinal; Unit: n/a; Range: {high = …, medium = …, low = …); Object: Document/Inspection-pair)
- ....



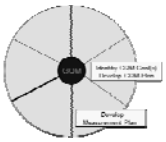
Step 2: GQM Planning – Developing the GQM Hierarchy

Example GQM Hierarchy (incomplete):

- Question 1.2: What is the distribution of failures by criticality?
- Model: D = F(x, y) = x[y]/x[all], x = Measure 1.1, y = Measure 1.2, where D: distribution of # failures per criticality class



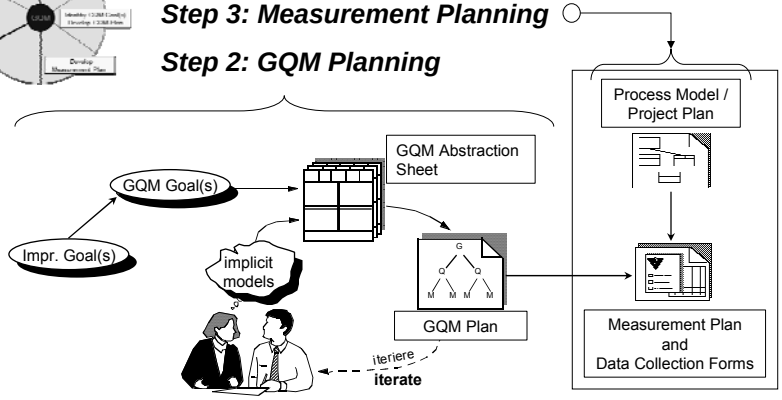
- Measure 1.1: Failure count (S: absolute; U: integer; R: positive integer; O: product version 1.0)
  - *Hypothesis: 120 failures*
- Measure 1.2: Failure criticality (S: nominal; U: n/a; R: {critical = complete breakdown of system, uncritical = unable to perform one or more of the functions F1, ..., F6, other}, O: failure report)
  - *Hypothesis: 5% critical failures, 15% major failures, 80% minor failures*



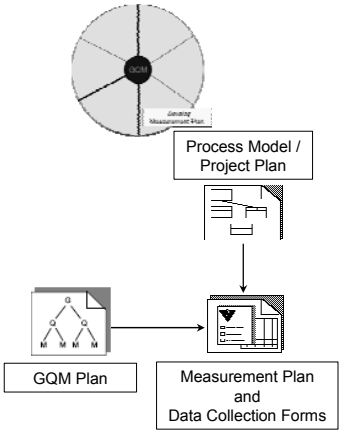
Planning Phase: Steps 2 & 3

Step 3: Measurement Planning

Step 2: GQM Planning



Step 3: Measurement Planning



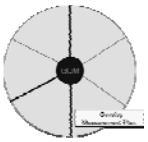
Measurement Plan – Objectives

- The measurement plan defines by whom, how, and when data collection for each measure should be performed.

Specification of:

- What data is collected? (← GQM plan)
- When is the data collected? (← process model)
- By whom is the data collected? (← process model)
- How is the data collected? (automatic by tool; via data collection forms, interviews, etc.)
- Who is responsible for quality assurance of the data?
- Who is responsible for the data handling/storage?

Step 3: Measurement Planning



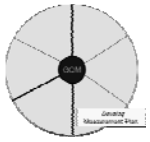
Measurement Plan – Example

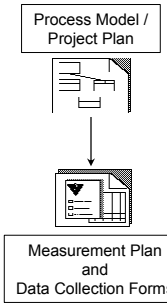
- Table for tracing Measurement Plan entries to GQM Plan, Project Plan and Data Collection Forms

Goal-ID	Metric-ID	Metric-Name	Data Creation Event	Data Col. Time	Data Col. Resource	Data Provider	Data Collector	Form-Id
…	…	…	…	…	…	…	…	…
Goal 1	M1.1	Failure count	Failure Report Summary	Test COMPLETE	TOOL: Failure Management System	Tester	QA Manager	Form X
Goal 1	M1.2	Failure criticality	Failure Report	Test report COMPLETE	TOOL: Failure Management System	Tester	QA Manager	Form X
…	…	…	…	…	…	…	…	…
Goal 1	M4.1	Dev. team experience	Project team assignment	Project START	HUMAN: Interview or Questionnaire	Team member	Project Manager	Form Y
Goal 1	M5.1	Document count	CM system report	Test COMPLETE	TOOL: CM system	Developer / Tester	Project Manager	Form Z
Goal 1	M5.2	Document type	Document complete	Test COMPLETE	TOOL: CM system	Developer / Tester	Project Manager	Form Z
…	…	…	…	…	…		…	…

GQM Plan Project Plan Data Collection Forms

Step 3: Measurement Planning

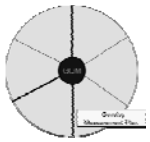


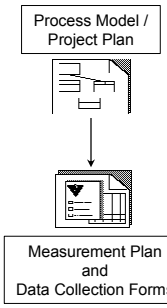


Data Collection – When?

- PERIODICALLY
  - Example: % of modules tested, cumulative effort over time
  - Required knowledge: typical frequency of information updates (daily, weekly, monthly, etc.)
- START/END of activity (phase)
  - Example: fault detection rate, effort of testing activities
  - Required knowledge: process model
- STATE of artifact (product), i.e., COMPLETE, CREATED, etc.
  - Example: observed quality of artifact
  - Required knowledge: state-transition diagrams of products

Step 3: Measurement Planning

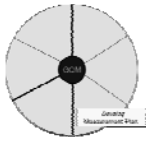


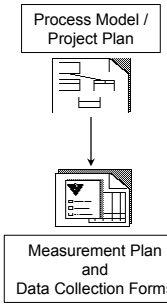


Data Collection – By Whom?

- Data providers and collectors should be selected considering their:
  - Expertise: Who has the technical/managerial expertise to provide/collect the data accurately?
  - Bias: Is there any reason for the data provider/collector to show any bias in the information s/he provides?
  - Access: Who has access to the object being measured?
  - Cost: Is the time needed for data provision/collection within budget?
  - Availability: Is the person available to spend time on data provision/collection?
  - Motivation: How committed is the person to the measurement program?

Step 3: Measurement Planning

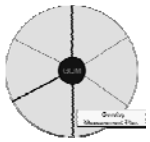


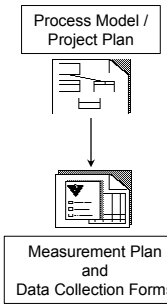


Data Collection – How?

- Data collection "tools" are:
  - Interviews
  - Questionnaires (paper form or on-line)
    - e.g. Failure Report Form
  - Automated tools
    - e.g., Static Code Analyzer
    - Automated data collection tools can be used for objective product measurement, whereas subjective measures are usually collected by forms.
    - Measurement tools can be triggered automatically by development tools, e.g. configuration management tools, defect management tools, test tools, compilers, etc.

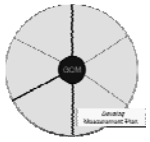
Step 3: Measurement Planning





Data Collection – Properties of Questionnaire

- Content/Format:
  - Identifying information: project name, date, etc.
  - Contains the measures listed in the Measurement Plan to be provided at the specified point of time and by the specified role.
- Structure/Style:
  - Requesting only short answers
  - Checklists
  - Logical, "natural" order of questions
- To make providing data easy:
  - Offer concise & clear definitions of the data to be provided
  - Use organizational standard terms and concepts
  - Explain used categories, if necessary (e.g., what is "critical"?)



Process Model / Project Plan

Measurement Plan and Data Collection Forms

Data Collection – Example Questionna

Defect Report Form

Project: _____

Date: _____

Name: _____

Please fill in one report form for each defect you detect.

Defect number: _____

How much time did you spend to isolate the defect?

_____ h _____ min

How much time did you spend to correct the defect?

_____ h _____ min

What is the defect type?

- ☐ Calculation
- ☐ Interface
- ☐ Control Flow
- ☐ Other

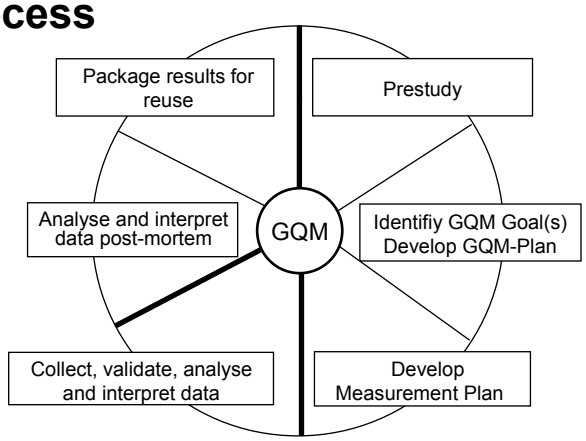
When was the defect injected?

- ☐ Requirements specification
- ☐ Design
- ☐ Coding
- ☐ Unknown

The GQM Process

Phases

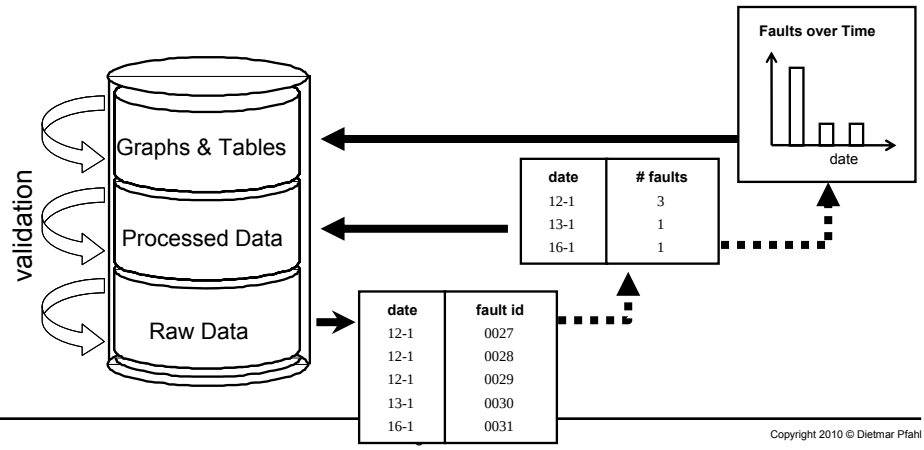
- Planning
- Execution
- Analysis



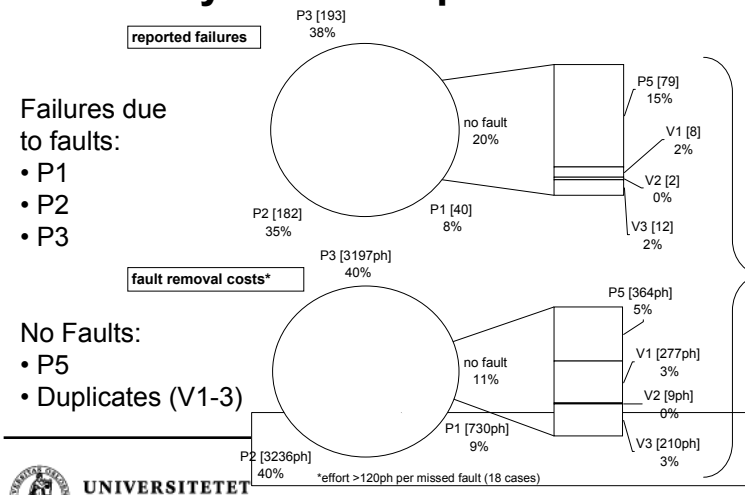
Data Collection and Analysis Principles

- Make sure that the data is collected according to the measurement plan
- During the process:
  - Validate the data
  - Analyze the data
- Format the data in understandable graphs/diagrams. Show trends.
- **Give feedback!** This is very central in GQM - present preliminary results to project members.
  - **Exercise**: Give three reasons in support of feedback during the process.
- After the end of the project, present the analysis with focus on learning from experience. Compare with the hypotheses. Discuss and involve everybody!
  - Project Post-Mortem

Data Collection & Storage: Example



Data Analysis and Report Generation: Example



Failures due to faults:
- P1
- P2
- P3

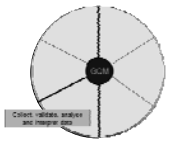
No Faults:
- P5
- Duplicates (V1-3)

20% of the failures reported during system test are no faults/no new faults.

They produce 11% of the total fault removal costs (= costs for analysis and correction).

UNIVERSITETET I OSLO

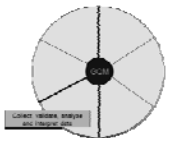
How many of the reported failures are not faults?



Measurement Implementation – Data Collection

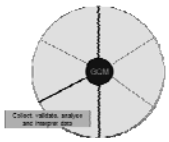
- Most of the data is usually provided by members of the project team
- People ***must not*** be controlled by measurement!
  - Anonymous data sources (i.e., no names shown) in feedback sessions
    - e.g., by accumulation of collected data
  - Use of data only for intended purposes
- Communication with data providers:
  - Data collectors must know which questionnaires they have to complete at which point of time
  - In case of unclear questions or misunderstandings, questionnaires have to be revised.
- The process for submitting completed data collection forms must be simple

UNIVERSITETET I OSLO



Measurement Implementation – Tool Support

- Data collection can be:
  - (semi-)automatic: via tools or on-line questionnaires
  - Manual: via paper-based questionnaires
- If available, use available tools (databases) with appropriate support for:
  - data storage, retrieval
  - data analysis
  - data representation and reporting (e.g. business graphics)



Measurement Implementation – Data Validation

Validation of raw data: What to check?

- Data collection forms have been submitted and are complete
- Values are of the specified type
- Values are of the specified range
- Look for outliers
- Dependencies between the data collection forms are explicit

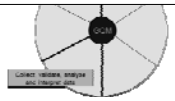
Validation of data in the database: What to check?

- Entries in the database match source values

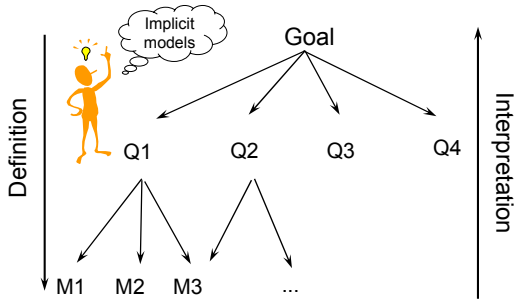
Validity of GQM documents: What to check?

- Are assumptions still correct? (If necessary: adaptation)

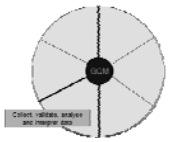
Measurement Implementation – Data Analysis & Interpretation



- Data Analysis and Interpretation follows GQM plan bottom-up



- Data Analysis
  - Statistical analysis to identify correlations, etc.
  - Analysis as preparation for ***feedback sessions***



Measurement Implementation – Data Analysis

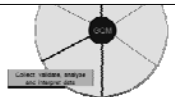
Data concerning the quality focus:

- Collected data are compared to actual or hypothesized baseline. This allows for:
  - Explaining the differences and determine if they are symptomatic of a problem
  - Trigger discussions with developers, project leaders, and management
  - Show the usefulness of measurement by identifying deviations from expectations or common knowledge

Data concerning the explanatory variables:

- Depending on the purpose, the following strategies are applied:
  - Did the explanatory variables have the expected impact on the quality focus? Is there evidence supporting the hypothesized relationship?
  - Post-hoc analysis of relationships

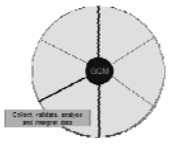
Measurement Implementation – Feedback Sessions /1



- ***Feedback sessions should be held periodically and include data providers/collectors, viewpoints (stakeholders) and GQM experts.***

Objectives:

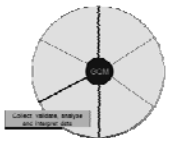
- Interpretation of correlations/trends/etc. identified by the data analysis
- Identifications of improvement opportunities
- Trigger of corrective actions concerning the development project, its underlying processes, or the measurement program
- Assessment/refinement the GQM plan and related documents.



Measurement Implementation – Feedback Sessions /2

Guidelines for Feedback Sessions:

- Focus on issues that need to be discussed
- Prepare presentation material
- Provide presentation material to participants in advance
- Perform feedback session with data providers and people in the viewpoint
- Report interpretations and conclusions based on the measurement data
- **Prepare/Plan the implementation of changes**



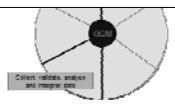
Measurement Implementation – Feedback Sessions /3

Presentation Material for Feedback Sessions:

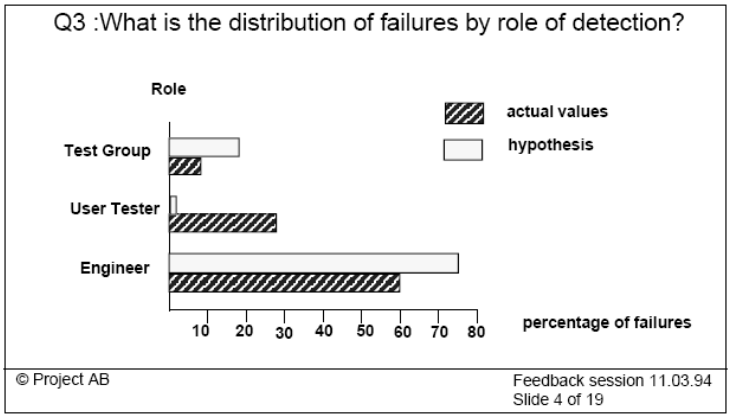
- For analyzed data show:
  - the questions of the GQM plan they intend to answer
  - corresponding hypotheses
  - descriptive statistics (variance, mean, etc.), histograms, box-plots, trend lines, models, etc.
  - number of underlying data points
- General rule: data should be displayed in an easy-to-understand way

UNIVERSITETET I OSLO

Measurement Implementation – Feedback Sessions /4

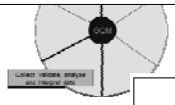


Data Presentation – Example 1:

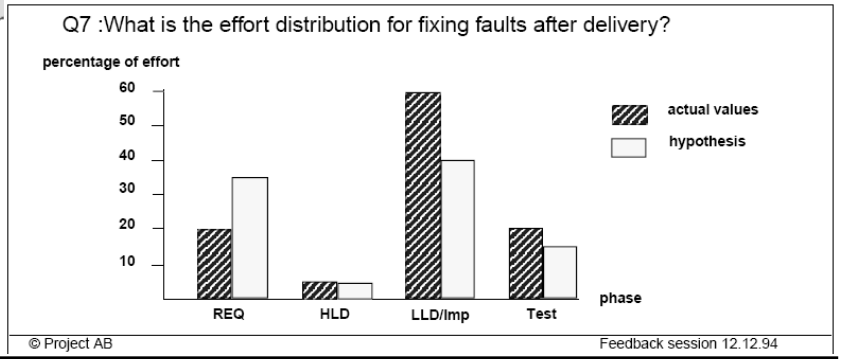


UNIVERSITETET I OSLO

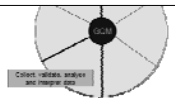
Measurement Implementation – Feedback Sessions /4



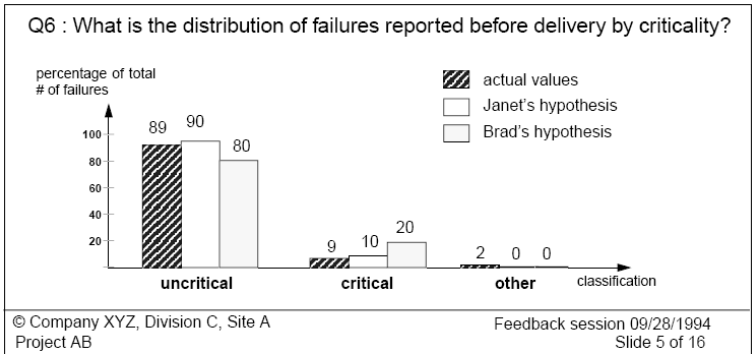
Data Presentation – Example 2:

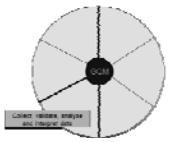


Measurement Implementation – Feedback Sessions /4



Data Presentation – Example 3:





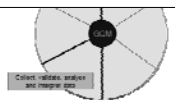
Measurement Implementation – Feedback Sessions /5

Main Steps:

1. **Discussion of deviations of the measurement data from hypotheses (hypotheses are the starting point for data interpretation)**
2. **Identification of causes abnormal values; often several explanations (causes) might be identified**
3. **In order to determine which interpretation is (most) appropriate, usually additional investigations have to be done**

UNIVERSITETET I OSLO

Measurement Implementation – Feedback Sessions /6

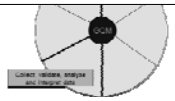


Interpretation Example [Nat94b]

- Low fault rates associated with a (intermediate) product have been reported; possible reasons can be:
  - Good quality artifact(s)
  - Unexpectedly simple artifact(s)
  - Incomplete artifact
  - Poor verification/validation
  - Large amount of code reuse or automatic code generation
  - Not all faults reported

UNIVERSITETET I OSLO

Measurement Implementation – Feedback Sessions /7



Interpretation Example (cont'd) [Nat94b]

Analysis result:

- Less faults than expected have been detected during inspection

Interpretation:

- Lack of process conformance due to insufficient training

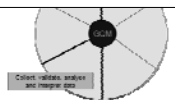
Short-term changes:

- Re-do the current activities with more experienced staff

Long-term changes:

- Review and improve staff training procedures on inspection

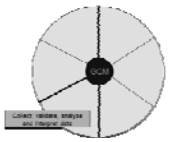
Measurement Implementation – Feedback Sessions /8



Follow-up to Feedback Session:

- **Plan process changes to achieve improvements**
- **Specify a "Process Improvement Plan" for each concluded change:**
  - ***Which*** **modification was agreed upon, e.g. modified standards, process models, new technologies, etc. ?**
  - ***Who*** **is responsible for the implementation of modifications?**
  - ***When*** **shall the modification be implemented?**
- **Implement changes !!!**

  If the modifications are not implemented, the measurement program only causes additional effort (and is useless)!



Measurement Implementation – Feedback Sessions /9

Follow-up to Feedback Session:

- **Monitor/Assess (implemented) changes through measurement as part of …**
  - **Controlled experiments**
  - **Case Studies**
  - **Surveys**

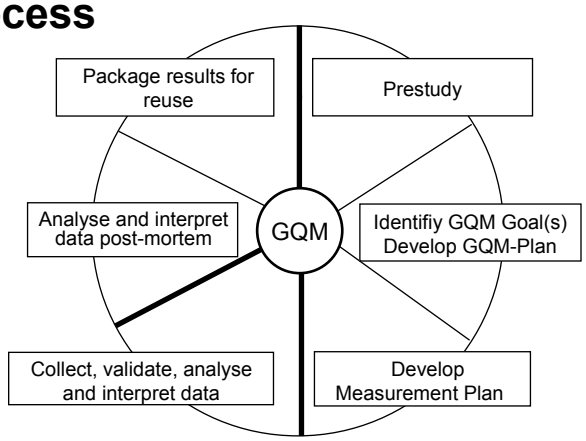
The GQM Process

Phases

- Planning
- Execution
- Analysis

⇩

Learning from Experience (Part 09)



GQM Success Factors

- Motivate the measurement program by clearly showing the relationship to improvement goals
- Assure management support
- Start small
- Expand the program slowly
- Involve all project members (information, feedback)
- Disseminate successful results

Examples of Measurement Programs

Siemens

Motorola

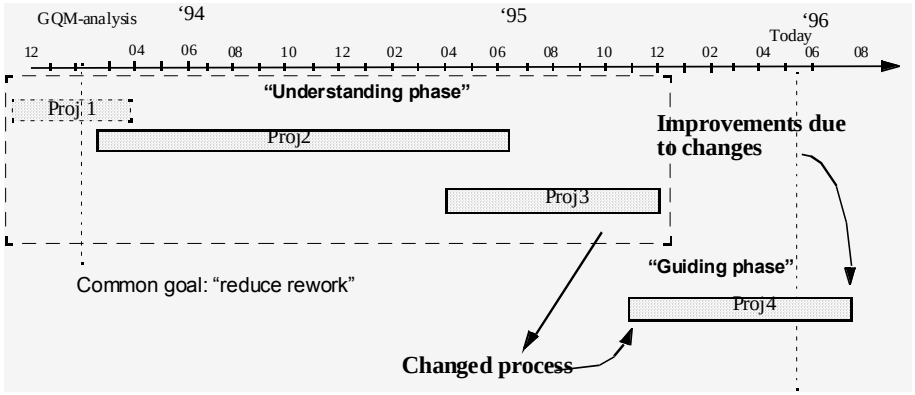
Measurement Program at Siemens

- The measurement program objective: *Reduce Rework*
- *Rework* = all work which should be done again, either because of failures, unclear or changed requirements.
- Characteristic of Siemens Defense
  - Well-defined waterfall model
  - Clearly-defined phase transitions
  - Culture of reviews
  - Cleanroom Software Engineering partly introduced
  - Well-established, advanced tool chain from design to test - automatic code generating

- Phase model:
  - Establishment - E
  - Analysis - A
  - Structural Design - SD
  - Detailed Design - DD
  - Implementation - Im
  - Integration - N
  - System Test - ST

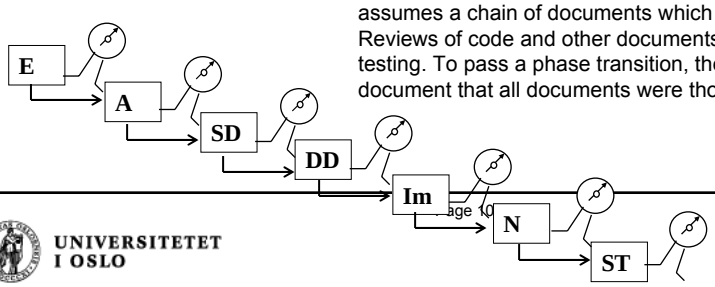
Improvement Plan Siemens

High-level plan – update June 1996



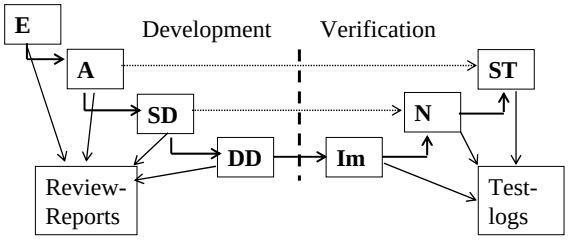
Measurement Program at Siemens

- Goal: to ***understand*** the origin of rework
- Underlying hypothesis: reduced rework yields **better reliability**, **shorter time-to-market** and **higher productivity**.
- Assumption: all development activities in the different phases can be improved. To understand the origin of changes, data must be collected about changes relative to phase.
- Cleanroom follows the *Stepwise Refinement* principle, i.e. it assumes a chain of documents which are gradually refined. Reviews of code and other documents are as fundamental as testing. To pass a phase transition, the responsible team must document that all documents were thoroughly checked.



Measurement Program Siemens

V-model:



Exercise:
To understand defect causes:
What type of information do we need to gather from review reports and test logs?

Measurement Program Siemens

- For each defect found, the following information is provided:
  - Phase in which defect was found
  - Classification (Minor, Major)
  - Origin (phase) when defect was injected
  - How much time used for detection, analysis, correction
  - Comment (defect cause, how could it be avoided etc...)
- Derived measures:
  - Total defects found for every phase (per defect class)
  - Total defects originating in phase (per defect class)

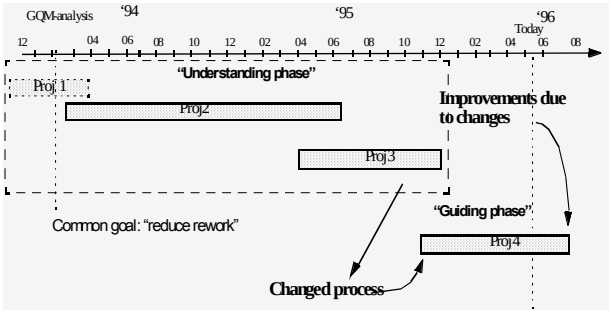
Results Siemens

***Defect Slippage Model* for project X (sum of minor and major):**

Rework effort					Origin of defect				
spent	Sum	%	E	A	SD	DD	lm	N	ST
E	0.00		0.00						
A	7.00	1.41	0.00	7.00					
SD	5.30	1.06	4.00	1.30	0.00				
DD	115.20	23.15	33.70	0.50	74.70	6.30			
lm	6.50	1.31	0.00	0.00	1.50	4.50	0.50		
N	90.40	18.16	0.00	0.00	38.80	40.10	9.00	2.50	
ST	273.30	54.91	1.50	6.50	5.30	170.20	85.30	0.00	24.00
Sum	497.70	100.00	39.20	15.30	120.80	221.10	74.80	2.50	24.00
%		100.00	7.88	3.07	24.27	44.42	15.03	0.50	4.82

Results Siemens

- The overall goal was to *Reduce Rework*: Did we reach this goal?



Software Metrics Initiative at Motorola [Das92]

Why?

- Engineers and managers wanted to better understand the software development process and be able to determine necessary changes to improve productivity, quality, and cycle time.

How?

- Definition of software processes
- Focusing on continuous process and product improvement
- Setting quantitative goals
- Controlling the achievement of goals

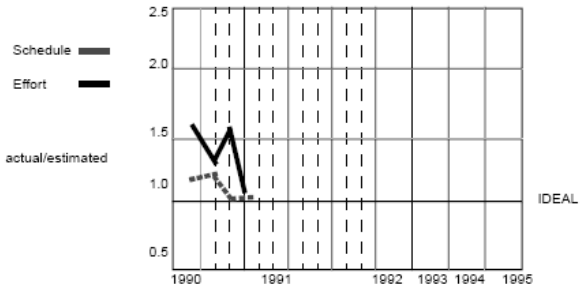
→ Measurement became an integral part of the software development process

Improvement Goals

- Goal 1: Improve project planning
- Goal 2: Increase defect containment
  → ability to detect and correct defects as soon as they are injected
- Goal 3: Increase software reliability
- Goal 4: Decrease software defect density
- Goal 5: Improve customer service
- Goal 6: Reduce cost of non-conformance
- Goal 7: Increase software productivity

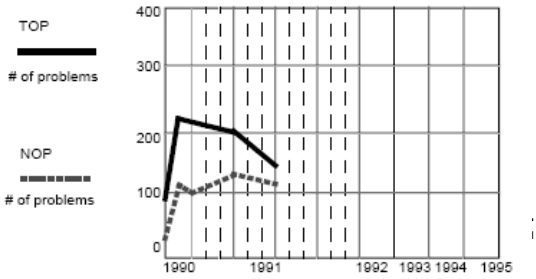
Goal 1: Improve Project Planning

- Question 1.1: How accurate are the estimates of the actual project schedule (duration)?
  - Metric 1.1: Schedule Estimation Accuracy (actual project duration/estimated project duration)
- Question 1.2: How accurate are the estimates of the actual project effort?
  - Metric 1.2: Effort Estimation Accuracy (actual project effort/estimated project effort)



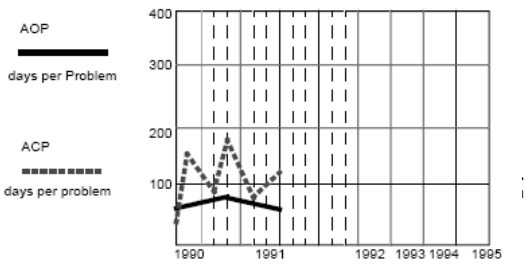
Goal 5: Improve Customer Service

- Question 5.1: What is the number of new problems that were opened during the month?
  - Metric 5.1: New Open Problems (NOP = number of new post-release problems that remain open at the end of the month)
- Question 5.2: What is the total number of open problems at the end of the month?
  - Metric 5.2: Total Open Problems (TOP = total number of post-release problems that remain open at the end of the month)



Goal 5: Improve Customer Service (cont'd)

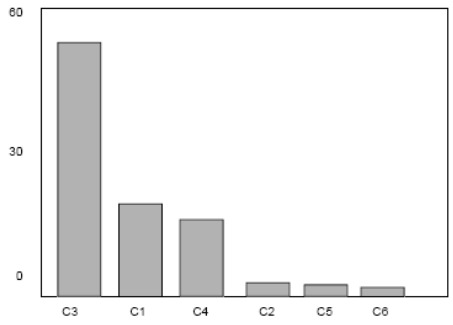
- Question 5.3: What is the mean age of open problems at the end of the month?
- Metric 5.3: (Mean) Age of Open Problems (AOP = total time post-release problems remaining open at end of month have been open / TOP)
- Question 5.4: What is the mean age of problems that were closed during the month?
- Metric 5.4: (Mean) Age of Closed Problems (ACP = total time post-release problems closed within the month were open / number of post-release problems closed within the month)



Use of Metrics for In-Process Project Control

- The charts shown on the previous slides are examples of the so-called “10-up software metrics charts”. These can be used for in-process control.
- More detailed data for in-process control includes:
  - Tracking of Life-Cycle Phase / Schedule Progress
  - Cost/Earned Value Tracking
  - Tracking of Impact of Requirements Changes on he project
  - Tracking of Design Progress
  - Fault-Type Tracking
  - Remaining Defects Estimates (e.g., using an assumed Rayleigh curve distribution for fault detection rate)
  - Effectiveness of Reviews (Design, Code)
  - Tracking the fixing of defects per priority/severity class, …

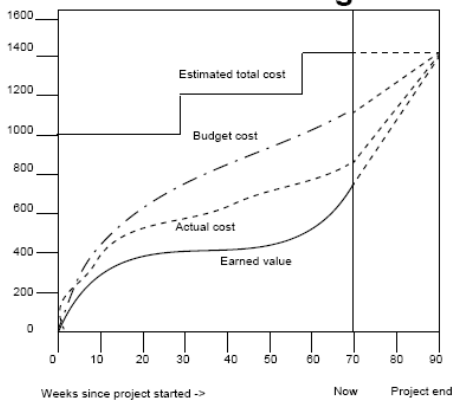
Fault Type Tracking



Purpose:

- Understanding (and communicating) the nature of code faults (and possibly their root causes) in order to prevent programmers from injecting similar faults in the future

Cost/Earned Value Tracking of the Project



Purpose: to allow manager to track in-process the following cost-related quantities (and update the project plan if necessary):

- Estimated total cost of the project
- Budgeted cumulative cost of the project
- Actual cumulative cost of the project
- Earned value of the project (the sum of the budgeted cost for the activities already completed by the project)

↪ summary of the actual progress of the project and how this relates to the project budget/cost.

Slide 273 (303)

Lessons Learnt

- Necessary prerequisites: infrastructure (cost accounting, configuration management, problem reporting), documented process
- Start with a small set of metrics addressing important improvement areas; then evolve over time
- Initial charts were used for in-process control and feedback (→ immediate impact of measurement)
- Data analysis should be done by engineers and managers, not by external experts (= facilitators of the measurement program)
- The code review package deployed by the Metrics Working Group was heavily used (67% of software engineers and managers)
- Metrics can only show problems and trigger corrective action; only if action is implemented benefits can be achieved

➔ Measurement is not the goal. The goal is improvement through measurement, analysis and feedback.

Cost of Measurement at Motorola

- Cost for meetings:
  - Metrics Working Group meetings ~ 8 participants (twice a quarter)
  - Metric User Group meetings (→ feedback sessions) ~ 15 participants (quarterly)
- Additional cost for data collection (incl. providing necessary tools), analysis and meeting preparation (~1% of total project resources)

Benefits of Measurement at Motorola

- Direct benefits wrt. quality, productivity and cycle-time improvement
  - Example: 50x reduction in released SW defect density within 3.5 years
- Indirect benefits: improvement wrt. Ship-acceptance criteria, estimation accuracy, continuous learning (engineers avoid mistakes made in earlier projects), better customer satisfaction (due to better quality of shipped products)
- Long-term benefit expected: cost reduction due to improved quality (reduced re-work) and reduced cycle-time
- Observations:
  - As engineers and managers start using metrics, they realize the potential benefits of such use, and they start investigating additional ways to obtain even more benefits
  - People start to think about the SW process and quality and are motivated to improve both