ESPA Seminar – Understanding Software Quality
May 7th, 2009, Spektri Duo, Espoo

Quality goal setting method
Experiences from four software product companies

Juha Itkonen, SoberIT, TKK

juha.itkonen@tkk.fi

http://www.soberit.hut.fi/espa

http://www.soberit.hut.fi/jitkonen

Quality goals – why?

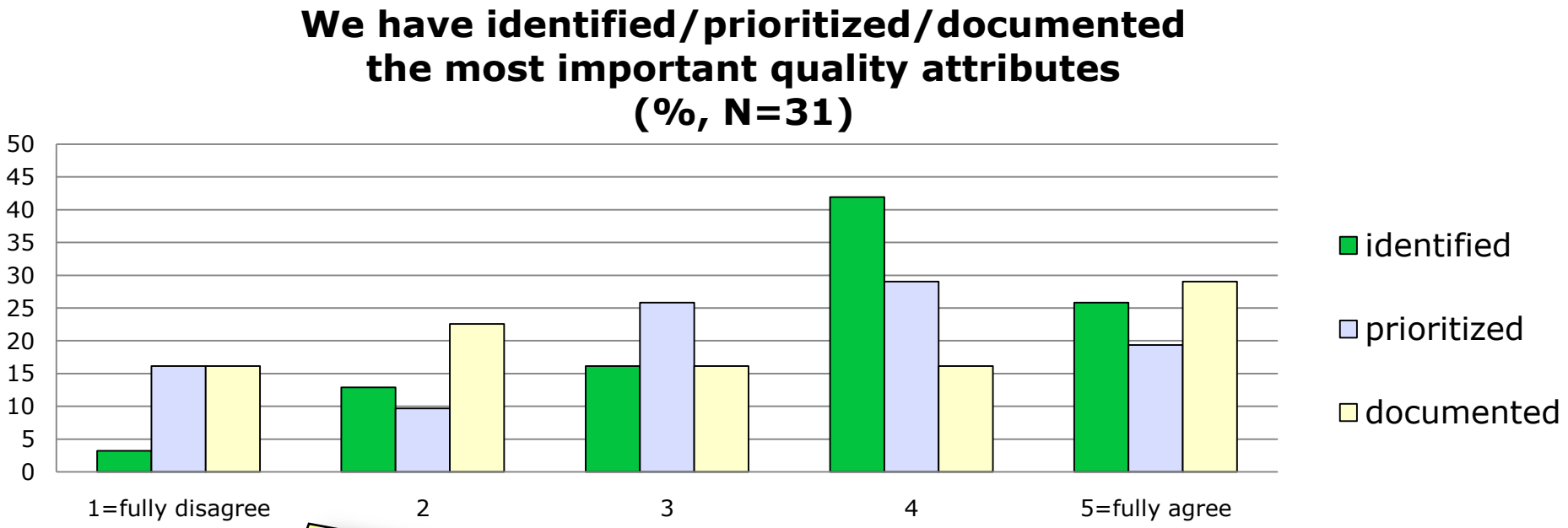
- Quality is in the eye of the beholder
  - Vague concept, different meanings, varying viewpoints
- Without goals, no matter which road do you take
- Quality goals help understanding and concretizing the desired quality characteristics and their level
  - Selecting and improving quality practices based on the quality goals
  - Improving the communication of the quality goals
  - Following the achievement of the quality goals

Software quality goal

A goal related to
an *external or internal* quality attribute
of final or intermediate *software product*.

- Characterizes the product from some quality viewpoint
- There are two types of goals
  - success factors, i.e. goals that are already reasonably well achieved
  - challenges, i.e. goals for which clear improvements are needed

How are quality goals handled in practice?
A sample from ESPA survey



Some companies have not identified their quality attributes, i.e., have no quality goals to strive for

Many companies have not prioritized or documented their quality goals

Quality Goal Setting Method

- Objective is to *identify*, *prioritize* and *elaborate* quality goals in a *certain context*
- Collaborative approach
  - Workshop
  - Many roles and viewpoints represented
  - Genuine discussion on the goals
- Context specific
  - Specific goals for a selected product
  - Concrete goals in a project context
- Subjective goal identification
  - Experience based
  - Supported by a quality model (e.g. ISO 9126)
- Focusing to the most important qualities
- Sustainable effort

The context where the method was created and applied

- Software product development organizations
  - Different sizes, mainly SME (all < 500 persons)
  - Existing, mature products
    - Incremental product development
  - Products with long development life-cycle
    - Several past and forthcoming releases
- Method was applied
  - to a specific software product
    - in the context of a certain project type
    - in the context of a specific release project

QGS method: Overview of the phases

- ✓ Step 1: Preparation and pre-assignment

Individual assignment
- 3-8 participants from different roles
- ½ hours / participant

- ✓ Step 2: Brainstorming
- ✓ Step 3: Prioritization by voting
- ✓ Step 4: Goal elaboration

Workshop
- 3-8 participants from different roles
- 4 hours / participant

- ✓ Step 5: Post-workshop finalizing activities

Individual/pair assignment
- Few hours

QGS method: Step 1 (individual assignment)

Preparation

- Selecting the context and participants
  - Roles of the participants
    - Management
    - Business / Sales
    - Marketing
    - Development
    - Quality assurance
    - Representative of customer and end-user
    - …

- Scheduling a workshop meeting
- Pre-assignment to participants
  - "prepare by using 10-30 minutes to list the most important quality goals for <*the product/project*> from your own viewpoint"

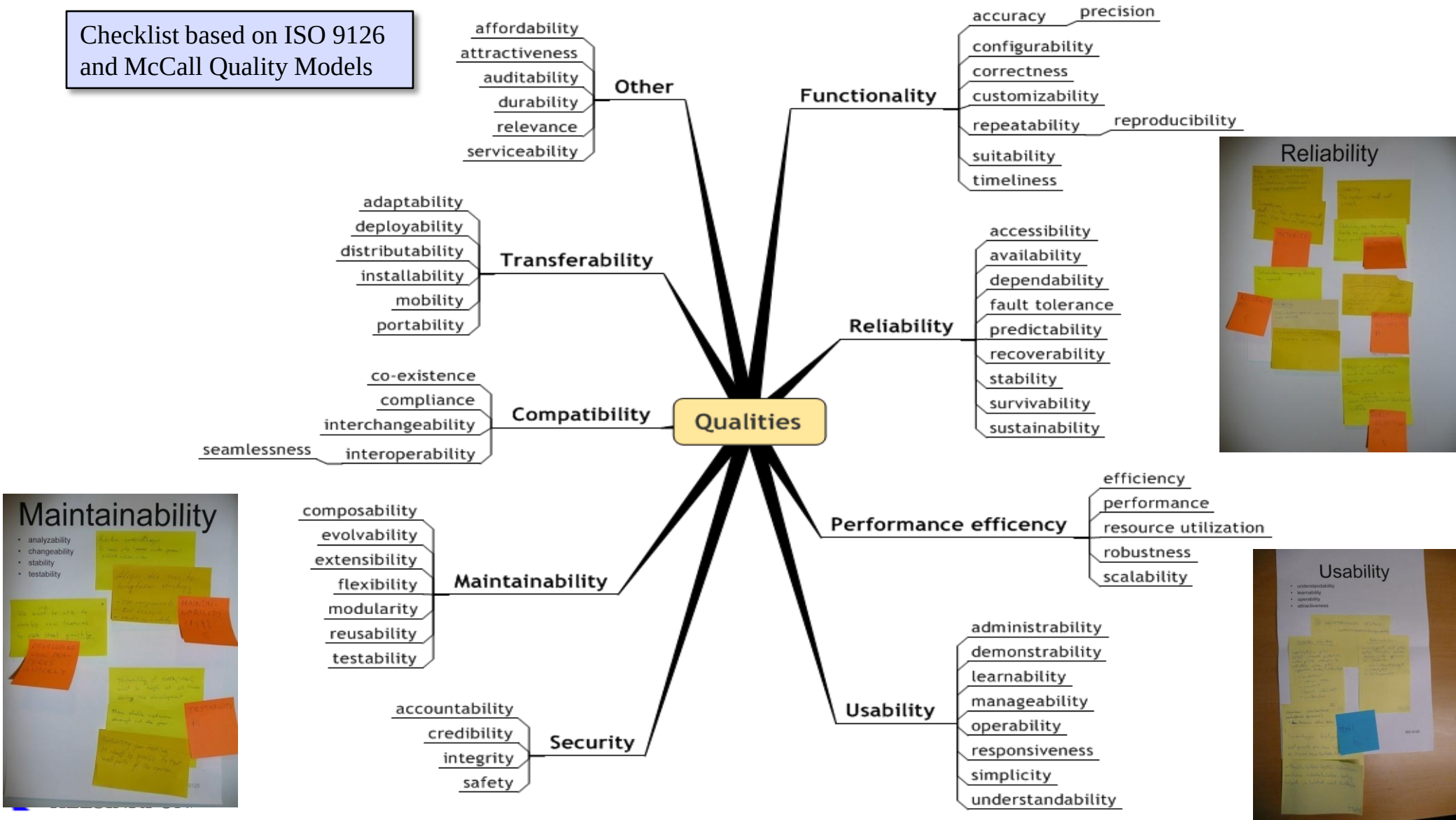
QGS method: Step 2 (workshop)

Brainstorming

- Brainstorming important quality goals for the product
  - Post-it notes on the wall
  - Ideas form the pre-assignment and new ideas
- Both success factors and challenges
  - Both types explicitly collected
- Briefly described
  - Can only state the quality attribute, not the actual goal
  - Elaborated later
- Similar and related ideas grouped to form goals

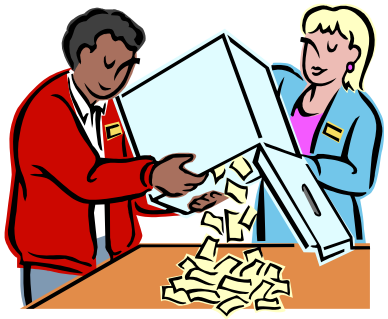
Additional brainstorming using a checklist

Checklist based on ISO 9126 and McCall Quality Models

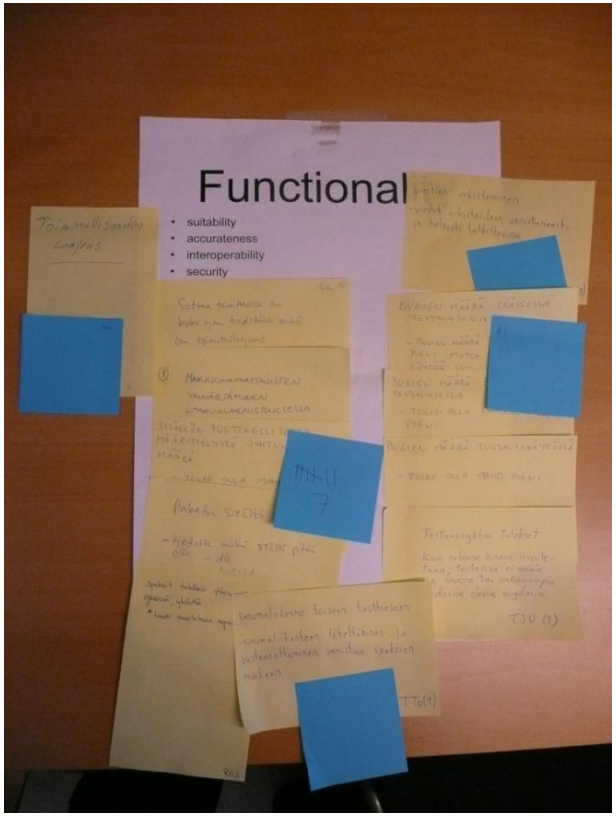


QGS method: Step 3 (workshop)

Prioritizing



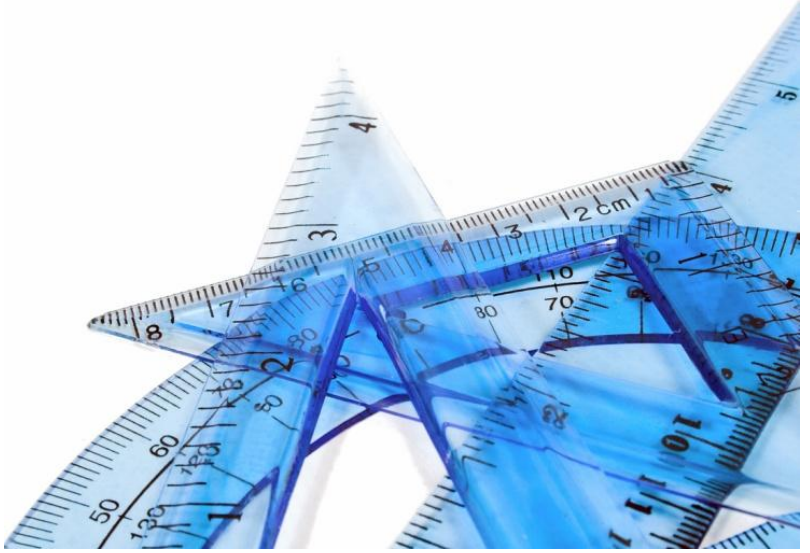
- Everyone has as many votes as there are goals
- Everyone can give any number of their votes to any goal
- Success factors and challenges should be treated equally
  - voting based on the importance of the goal
    - not on the fact that some goals have already been achieved better than others



QGS method: Step 4 (workshop)

Elaborating

- Top-5 goals are chosen for elaboration
- Goals described in more detail by following a given template
  - In groups of 2-3 people
  - 20-30 minutes per goal
- Discussion and refining the goals if needed



Goal template

- Name
  - short name for the goal
- Description
  - Description that documents the goal on higher level than a single measure.
- Rationale
  - Motivates why the goal is important. Gains and risks; E.g., costs, business benefits, customer viewpoint.
- Related factors
  - Tentative ideas of what contributes, or prevents achieving this goal
- Votes
  - Number of votes in prioritization

One or more quality indicators

- Measures that can be used indicate if this goal is achieved

- For each quality indicator:
  - Description
    - To understand exactly what is the indicator and a measure for it
  - Current level
    - Current value of the measure
  - Target level
    - Target for this indicator to achieve the goal
  - Breakpoints*
    - Utility, differentiation, saturation
  - Cost barriers*
    - If can be identified

*concepts from QUPER

Example of an elaborated goal

- Name: Easy updateability
- Description:
  - Updating the software should be quick and easy, ideally possible without deep technical or product knowledge
- Rationale:
  - Direct cost savings related to updates. Reduced risk of errors during updates.
- Related factors:
  - Robustness of the software, configurability, quality of installer software
- Votes: 6

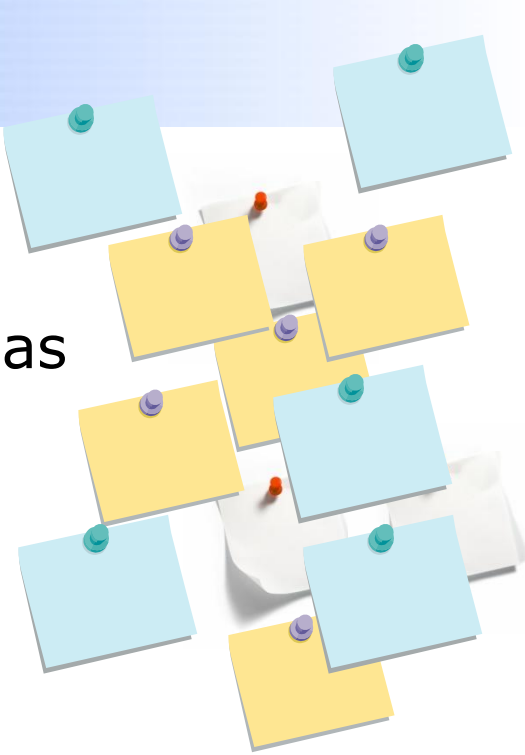
- QI 1: Updating effort
- Amount of average working hours consumed by making an update for a single customer installation.
  - Current level: 3 h
  - Target level: 15 min
  - Breakpoints:
    - Utility: 4h
    - Differentiation: 1h
    - Saturation: 10 min
  - Cost barriers
    - Automated installers have to be developed to reach better values than 1 h.

QGS method: Step 5 (individual/pair assignment)

Finalising documentation

- Finalize the description with the goal owner if needed
- Dig out the existing data on current level
  - for the quality indicators
- Communicate the goals
- List current practices that contribute achieving the goals

Results of applying the method



- Brainstorming -> around 40-50 individual ideas
- Combined -> 8-14 quality goals
- Detailed elaboration -> 2-5 goals

- Common goals
  - Usability
  - Installability & updateability
  - Functional correctness
  - Functional suitability
  - Performance efficiency

Utilizing the quality goals

- Selecting and improving practices
- Guiding work and tracking progress
- Communicating the goals

… to achieve the goals

Utilizing quality goals on project level
QGP for Features

- Bringing goals and practices down to project plan level
  - Use product level goals and practices to create a project level quality assurance plan
- Project specific goals are specified
  - In the context of a certain feature in a certain project
- Quality practices are designed to match the project specific goals
- Quality goals and practices are communicated for designers, developers, testers, …
  - What goals do we aim at
  - What exactly must be done to achieve the goals

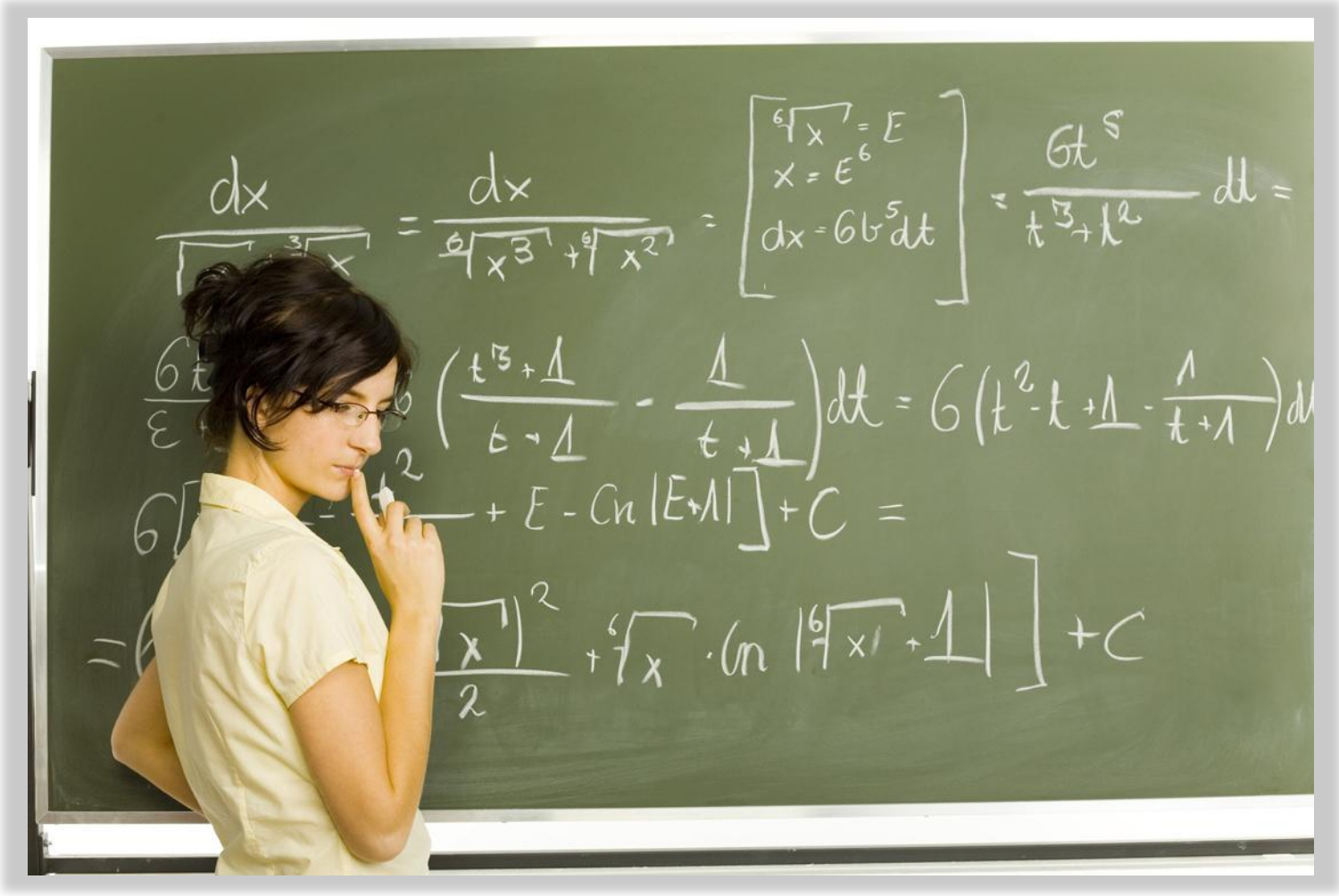
QGP for Features

1. Identifying goals for selected features
   - For each selected feature define ~1-3 goals
   - Elaborate each goal
     - What does e.g. "efficient to use" mean in the context of this feature in this project
2. Designing practices for achieving goals
   - Select at least one good practice for each goal
     - What practices has to be performed?
     - How the practices help to achieve the goal?
3. Planning tasks based on the selected practices
   - Results are included into the project plan
   - Plan tasks for each selected practice and feature
     - What needs to be done
     - How much effort
     - When
     - Who is responsible

Example:
How to bring product quality goal on feature level

- **Product Quality goal:** Easy to use – The product is easy to use for normal users. A deep "guru”-level knowledge is not required in normal use.
- **Feature:** A new sketching tool for an engineering application
- **Detailed quality goal for the feature:**
  - Tool use can be learned in a few minute demo
  - An existing user who knows the principles of modeling can use the tool correctly without referring to manuals
- **Practices:**
  - Manual testing by independent tester
  - Creating a 1-2 min how-to video
  - Beta testing with real users
  - Using beta version at training sessions

Lessons Learned



Lessons Learned:
Workshop based method seems to work

- Defining useful quality goals is possible
  - By little guidance and mentoring
  - Even easier than expected
  - Resulting quality goals were perceived good and useful
- Measures clarify the sometimes vague quality goals
  - Example: **Installability -** The software must be easy to install, update and configure.
    - Throughput time per update (including fixes) < 5h
    - Average number of reported defects from updates < 2
- Quality model was partly useful
  - provided a good structure to organize the goals
  - only few additional goal propositions raised, if any

Lessons Learned:
Effects of the Context

- A context of a specific project tends to be challenging
  - Focusing on specific details of a project context first
  - A lot of generic goals and practices emerge

- Method worked smoothly in the context of one product
  - Easier to maintain focus
    - Wide and generic goals avoided
      - generic or company-policy level
  - Drilling down to the project level was easy based on product level goals and practice list

Lessons Learned:
Perceived challenges

- Separating different types of goals and practices essential
  - Success factors and challenges
  - Current practices and improvement ideas
- People have a tendency to think solutions first
  - Before analysing or even identifying properly the goals
- Multiple viewpoints in prioritization
  - Focusing to goals that need improvement
  - Important success factors were easily neglected
  - People in different roles see different priorities
  - Effect of the dependencies between the goals
- Workshops with a large group take a lot of effort
  - Trade-off between spent effort and wide involvement
  - Easy to slip into inefficient chit-chat around the quality topics

Lessons Learned:
Practical tips

- Ensure that participants have enough authority to represent their group
- Four-hour workshop is a heavy exercise – split it into 2 two-hour workshops
  - Gives time to think and recharge
- Cut down the effort by focusing
  - Do the initial brainstorming by smaller group
    - Pre-assignment can be collected from larger group
  - Do elaboration individually or in pairs
  - Use large workshop for communicating and prioritization
- List first the success factors and after that the goals that need improvement
- Chair should ensure goal names reflect goals, not solutions
  - E.g., "testability", not "efficient test automation"

Questions and more discussion

Contact information

Juha Itkonen

juha.itkonen@tkk.fi

+358 9 451 6041

http://www.soberit.hut.fi/espa

http://www.soberit.hut.fi/jitkonen