

Selecting and Using Software Measures

JPL SIM Workshop

January 29, 2004

Rose Pajerski



Topics

- **Metrics Selection**
- **Metrics Analysis/Collection**
- **Example**



Considerations in Selecting Metrics - 1

- **Decide measurement goals up front**
 - **WHO are the stakeholders?**
 - **Internal and external to project**
 - **Different perspectives may yield different measures**
 - **Quality measures**
 - **To system engineer: hw/sw interface requirements problems found/fixed**
 - **To software engineer: design and coding errors found/fixed**
 - **To system tester: requirements specification detailed and testable**
 - **Progress measures**
 - **Schedule performance is primary driver**
 - **Effort/budget performance is primary driver**
 - **Impact of changes to requirements is seen as high risk**



Considerations in Selecting Metrics - 2

- **Decide measurement goals up front**
 - **WHAT do they want to know?**
 - **Pose questions from different stakeholders' perspectives**
 - **Pose questions based on knowledge needs, e.g.,**
 - **Baseline/characterize**
 - **Model**
 - **Predict**
 - **WHEN do they need to know?**
 - **Weekly, monthly progress reporting**
 - **Link to milestone events**



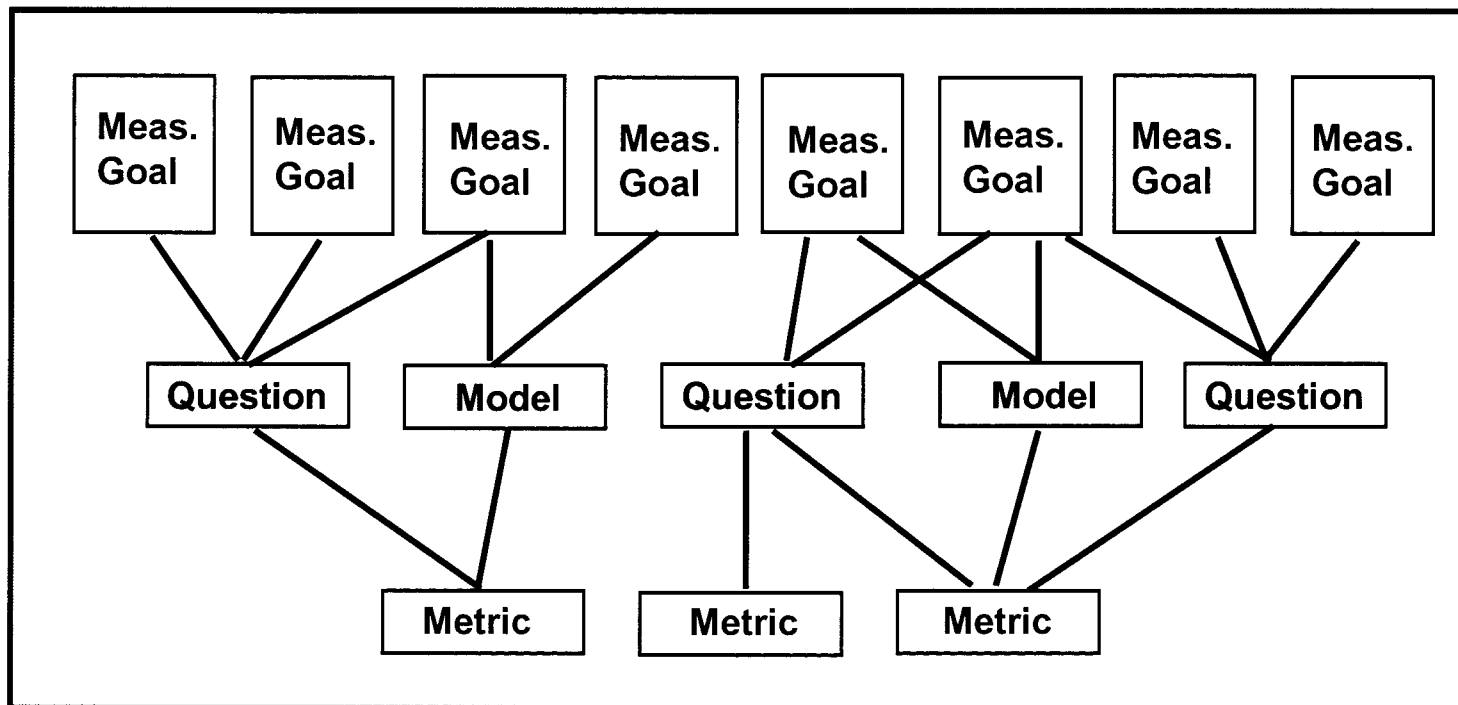
Considerations in Selecting Metrics - 3

- **Decide measurement goals up front**
 - **WHERE will data come from?**
 - **Tie to existing processes and tools**
 - **Reviews**
 - **Action items**
 - **Requirements definition/DOORS**
 - **TBDs, changes, mapping to components**
 - **Software design/architecture**
 - **Complexity parameters**
 - **Requirements inspections/reading**
 - **Defect reports**



Measurement Infrastructure Goal/Question/Metric Approach

Mechanism for defining and interpreting operational, measurable goals



- Each metric supports multiple goals
- Questions focus metric selection and in-process analysis



Considerations in Analyzing Metrics - 1

- **Start analysis and reporting from Day 1**
 - **Initial reports may be activity measures**
 - **Move into progress analysis**
 - **Activity measures against a plan**
 - **Coming up with the plan**
 - **Historical baseline**
 - **Manager expertise**
 - **Expected % change in requirements**
 - **Expected % growth in requirements**
 - **Build on actuals from Phase to Phase**
 - **Begin to build a model from actuals**



Considerations in Analyzing Metrics - 2

- **Start analysis and reporting from Day 1**
 - **Use whatever sources exist**
 - **RM counts, status**
 - **CM counts, status**
 - **Use tools (if available) to collect and analyze the data**
 - **Don't wait for the perfect tool**
 - **Manual collection can be low cost, low impact**
 - **Plan to review metrics set from Phase to Phase**



Example - 1

Goal: to evaluate the impact of design changes in a system against a given software architecture

- **Evaluate the initial architecture to form the baseline (ideal design)**
 - **Characterize by types, number of interfaces**
 - **Identify design guidelines used to create initial architecture**
- **“Implement” design changes at high level**
- **Measure modified design to extract “actual” design and to compare against planned “ideal” design (look for discrepancies)**



Example - 2

- **Coupling guidelines**
 - **Coupling from component-based modules to library-based modules is desirable (more=better).**
 - **Coupling from library-based modules to other modules is undesirable.**
 - **Coupling among component-based modules is undesirable.**
 - **Coupling within a single component-based module is more desirable than coupling among component-based modules**
- **Measures**
 - **CBM – coupling between modules**
 - **CBMC – coupling between module classes**



Example - 3

- **Metrics guidelines**
 - Values chosen arbitrarily
 - “CBM should be less than or equal to 2”
 - The only exceptions will be component X”
 - “CBMC should be less than or equal to 10”
 - The Y module will be the only exception to this rule”
- **Results of case studies**
 - Evaluation process is a way of monitoring and steering the actual implementation of the software architecture.
 - Metrics are useful for measuring coupling and cohesion for a high-level architectural design, but need to be tailored
 - It’s cost-efficient and quick



Selecting and Using Software Measures

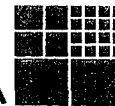
JPL SIM Workshop
January 29, 2004

Rose Pajerski



Topics

- **Metrics Selection**
- **Metrics Analysis/Collection**
- **Example**



Considerations in Selecting Metrics - 1

- **Decide measurement goals up front**
 - **WHO are the stakeholders?**
 - **Internal and external to project**
 - **Different perspectives may yield different measures**
 - **Quality measures**
 - **To system engineer: hw/sw interface requirements problems found/fixed**
 - **To software engineer: design and coding errors found/fixed**
 - **To system tester: requirements specification detailed and testable**
 - **Progress measures**
 - **Schedule performance is primary driver**
 - **Effort/budget performance is primary driver**
 - **Impact of changes to requirements is seen as high risk**



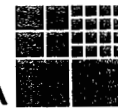
Considerations in Selecting Metrics - 2

- **Decide measurement goals up front**
 - **WHAT do they want to know?**
 - **Pose questions from different stakeholders' perspectives**
 - **Pose questions based on knowledge needs, e.g.,**
 - **Baseline/characterize**
 - **Model**
 - **Predict**
 - **WHEN do they need to know?**
 - **Weekly, monthly progress reporting**
 - **Link to milestone events**



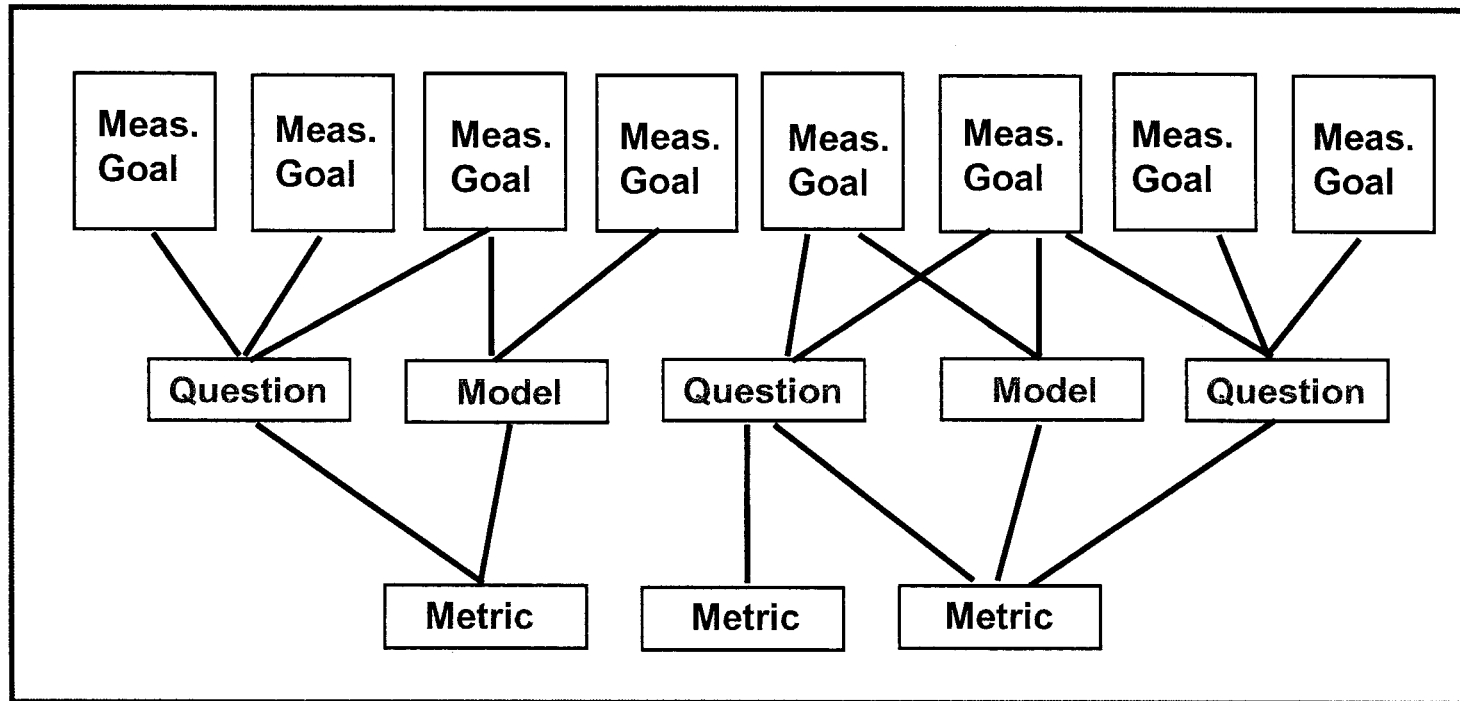
Considerations in Selecting Metrics - 3

- **Decide measurement goals up front**
 - **WHERE will data come from?**
 - **Tie to existing processes and tools**
 - **Reviews**
 - **Action items**
 - **Requirements definition/DOORS**
 - **TBDs, changes, mapping to components**
 - **Software design/architecture**
 - **Complexity parameters**
 - **Requirements inspections/reading**
 - **Defect reports**



Measurement Infrastructure Goal/Question/Metric Approach

Mechanism for defining and interpreting operational, measurable goals

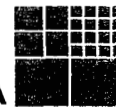


- Each metric supports multiple goals
- Questions focus metric selection and in-process analysis



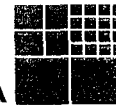
Considerations in Analyzing Metrics - 1

- **Start analysis and reporting from Day 1**
 - Initial reports may be activity measures
 - Move into progress analysis
 - Activity measures against a plan
 - Coming up with the plan
 - Historical baseline
 - Manager expertise
 - Expected % change in requirements
 - Expected % growth in requirements
 - Build on actuals from Phase to Phase
 - Begin to build a model from actuals



Considerations in Analyzing Metrics - 2

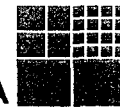
- **Start analysis and reporting from Day 1**
 - **Use whatever sources exist**
 - **RM counts, status**
 - **CM counts, status**
 - **Use tools (if available) to collect and analyze the data**
 - **Don't wait for the perfect tool**
 - **Manual collection can be low cost, low impact**
 - **Plan to review metrics set from Phase to Phase**



Example - 1

Goal: to evaluate the impact of design changes in a system against a given software architecture

- **Evaluate the initial architecture to form the baseline (ideal design)**
 - **Characterize by types, number of interfaces**
 - **Identify design guidelines used to create initial architecture**
- **“Implement” design changes at high level**
- **Measure modified design to extract “actual” design and to compare against planned “ideal” design (look for discrepancies)**



Example - 2

- **Coupling guidelines**
 - Coupling from component-based modules to library-based modules is desirable (more=better).
 - Coupling from library-based modules to other modules is undesirable.
 - Coupling among component-based modules is undesirable.
 - Coupling within a single component-based module is more desirable than coupling among component-based modules
- **Measures**
 - **CBM** – coupling between modules
 - **CBMC** – coupling between module classes



Example - 3

- **Metrics guidelines**
 - **Values chosen arbitrarily**
 - **“CBM should be less than or equal to 2**
 - **The only exceptions will be component X”**
 - **“CBMC should be less than or equal to 10**
 - **The Y module will be the only exception to this rule”**
- **Results of case studies**
 - **Evaluation process is a way of monitoring and steering the actual implementation of the software architecture.**
 - **Metrics are useful for measuring coupling and cohesion for a high-level architectural design, but need to be tailored**
 - **It’s cost-efficient and quick**

