Measurement and Analysis in software engineering

Practice of CMMI implementation

KRES-Consulting (ООО «КРЕС-Консалтинг»)

Moscow, 2004

http://www.kres.ru

Goals of the presentation

- To research why and when should measurements be used in software process
- Give you advises how to decrease complexity of measurements for people who will measure and use metrics

Nature of measurements

People need measurements if they want to “feel” entity’s characteristics but cannot feel it themselves

Measurement is …

Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world to describe them according to clearly defined rules

Measurements complexity

- High level of abstraction is barrier for understanding
- There is a difference between an object and measurement results
- High cost

Nature of measurements

In life people measure only very important, critical things …

and don't measure useful things!!!

Motives to measure

- *Single measuring:* to determine actual state (for example is situation “bad” or “good”) in order to undertake appropriate actions
- *Periodical measuring:* to monitor changes and deviations in order to make actions on time
- *Serial measuring:* to demonstrate regularity or to find exceptions

Key Reasons for Software Measuring

1. **Managing** Software Projects
2. **Understanding** Software Process
3. **Improving** Software Process

Measurements for ***management*** of a software project

Measurements for Managing of a Software Projects

1. Planning (estimating main project's parameters, predicting project evolution and planning preventive actions)

2. Monitoring and Control (monitoring project state and making corrective actions on time, predicting project evolution and making preventive actions)

Measurements for Managing: *Planning*

1. Measure ***current (present)*** attributes

2. With the help of models estimate main project's parameters and predict project evolution in order to create a ***contract***, make a ***plan*** and allocate ***resources***

Measurements for Planning: *Measuring current attributes*

- List of requirements – the only thing you have at the beginning of a project
- You can measure attributes of tasks and work products:
  - Size
  - Complexity
  - Connectivity

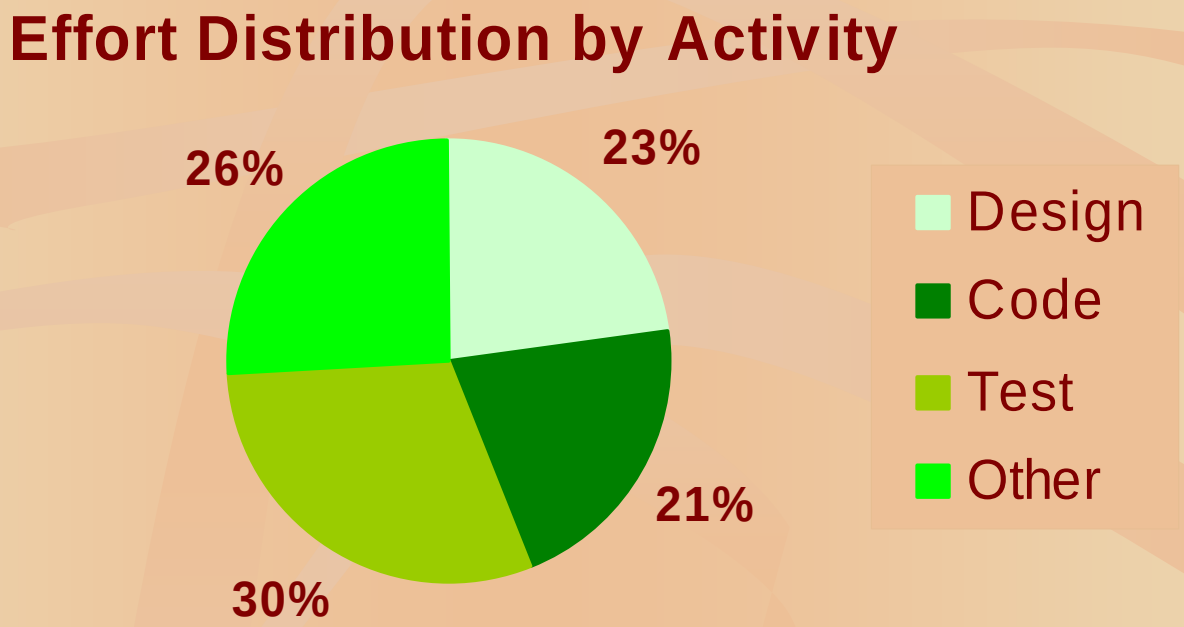
Measurements for Planning: *Estimating*

- Factors that are typically considered under estimating include:
  - Identified tasks and work products and their attributes (size, complexity, etc.)
  - Technical approach
  - Selected project life-cycle model
  - Models or historical data

Measurements for Planning: *Planning tasks*

- According to Project Planning Process Area you should make the following estimations:
  - ✓ Efforts
  - ✓ Cost
- According to Project Planning Process Area you should make plans for:
  - ✓ Schedule
  - ✓ Budget
  - ✓ Risks
  - ✓ Resources
  - ✓ Data
  - ✓ Knowledge and skills
  - ✓ Stakeholder involvement

Measurements for Planning: *Example 1*



Measurements for Planning: *Example 2*

$$\text{Effort (in staff} - \text{month)} = 1.48 \times (\text{KSLOC})^{0.98}$$

$$\text{Duration (in month)} = 4.6 \times (\text{KSLOC})^{0.26}$$

$$\text{Pages of Documentation} = 34.7 \times (\text{KSLOC})^{0.93}$$

Measurements for Managing of Software: *Monitoring and Control*

"You cannot control what you cannot measure" *(DeMarco, 1982)*

Measuring for Monitoring and Control: *Step by Step*

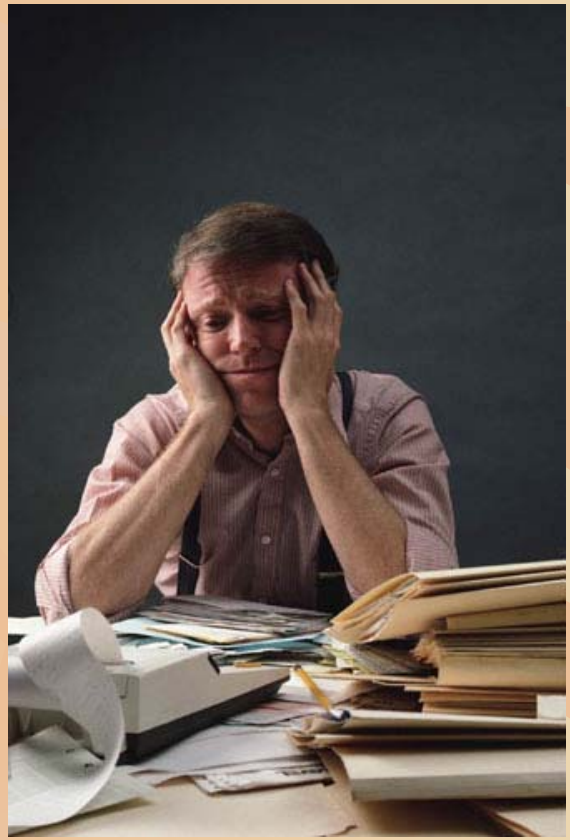
1. Choose *an object* (or situation) you need to manage
2. Determine *probable states* of an object (or situation)
3. Choose some of them (*special states*) that require special attention
4. Determine what *indicators* would signalize you that *the object* becomes *special state*

Measuring for Monitoring and Control: *Step by Step (continued)*

5. Establish a *metric*, a *unit* and a *scale* for measuring indicators
6. Establish a measurement procedure
7. Measure and understand what *place* does measured value occupy on your scale
8. With the help of models determine state of the object

Measuring for Monitoring and Control: *Example 1*

1. An object: an employee
2. Probable states of the object: working, not working, ill, unable to execute a task properly, underloaded, overloaded, etc.
3. Special states: unable to execute a task properly, underloaded, overloaded



Measuring for Monitoring and Control: *Example 1 (continued)*

Signalizing indicators:

- *Unable to execute a task properly* – makes a lot of mistakes, use a lot of help, doesn't achieve a result
- *Underloaded* – have free time
- *Overloaded* – cannot complete work on time

Measuring for Monitoring and Control: *Example 1 (continued)*

Metrics and Units:

- Makes a lot of mistakes (*number of mistakes*), use a lot of help (*hours of help per task, number of help request per task*)
- Have free time (*free hours*)
- Cannot complete work in time (*hours of delay*)

Measuring for Monitoring and Control: *Example 1 (continued)*

Scales:

- number of mistakes (absolute: 0, 1, 2, …), hours of help per task (ratio: 0, 0.5, 1, …), number of help request per task (absolute: 0, 1, 2, …)
- free hours (ratio: 0, 0.5, 1, …)
- hours of delay (ratio: 0, 0.5, 1, …)

Measuring for Monitoring and Control: *Example 2*

1. An object: a project
2. Probable states of the object: started, in progress, finished, there are problems, there are no problems, etc.
3. Special states: there are problems

Measuring for Monitoring and Control: *Example 2 (continued)*

Signalizing indicators:

- There are problems – there are deviations from plan, there are deviations from baseline, unexpected negative circumstances appeared

Measuring for Monitoring and Control: *Example 2 (continued)*

Metrics and Units:

- There are deviations from planned schedule *(hours of delay)*, milestones *(hours of delay)*, budget *(quantity of money overdraft)*, knowledge and skills management plan *(level of skills)*, stakeholder involvement plan *(number of interactions with stakeholder)*
- There are deviations from baseline: project size *(number of team members, number of business requirements, quantity of money, duration in month)*

Measuring for Monitoring and Control: *Example 2 (continued)*

Scales:

- hours of delay (ratio: 0, 0.5, 1, …), quantity of money overdraft (ratio: 0, 0.01, 0.02, …), level of skills (ordinal: low, normal, high), number of interactions (absolute: 0, 1, …)
- number of team members (absolute: 1, 2, …), number of business requirements (absolute: 1, 2, …), quantity of money overdraft (ratio: 0, 0.01, 0.02, …), duration in month (ratio: 0, 1, 2, …)

Measuring for Monitoring and Control: *Example 3*

1. An object: a requirements management process; related object: requirements

2. Probable states of the object: derived, analyzed, inconsistent, not full, changed, etc.

3. Special states: inconsistent, not full, changed

Measuring for Monitoring and Control: *Example 4*

1. An object: a product

2. Probable states of the object: works, doesn't work, works as appropriate, works as not appropriate, with defects (faults and failures), without defects

3. Special states: don't work, work as not appropriate, with defects (faults and failures)

Accents in achieving goals of measurements

1. Usefulness of metrics
2. Knowledge
3. Culture
4. Motivation

Measurements for ***understanding*** software process

Measurements for Understanding Software Process

1. Establishing baseline
2. Finding relationships
3. Creating models

Measurements for Understanding: *Baseline*

1. Establish baseline based on metrics you need to manage a project

2. Exclude specific metrics

Measurements for Understanding: *Relationships and Models*

Is reliability a function of testing time?

?

If I use C#, will I increase productivity and reduce cost?

?

How long will it take to finish if we add more functionality?

?

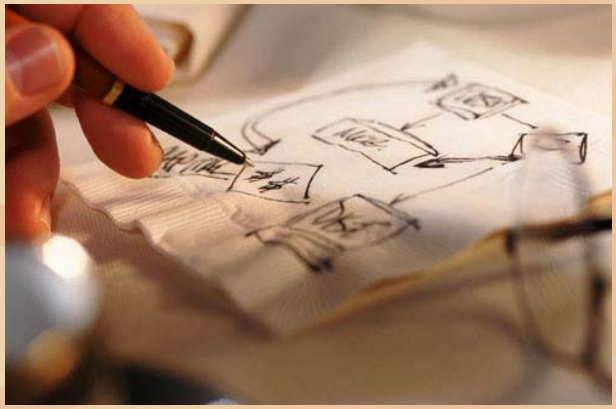
If I change the testing standards, will we find more errors?

?

If I add more staff, how much can I mpress the schedule?

?

Measurements for Planning: *Models*



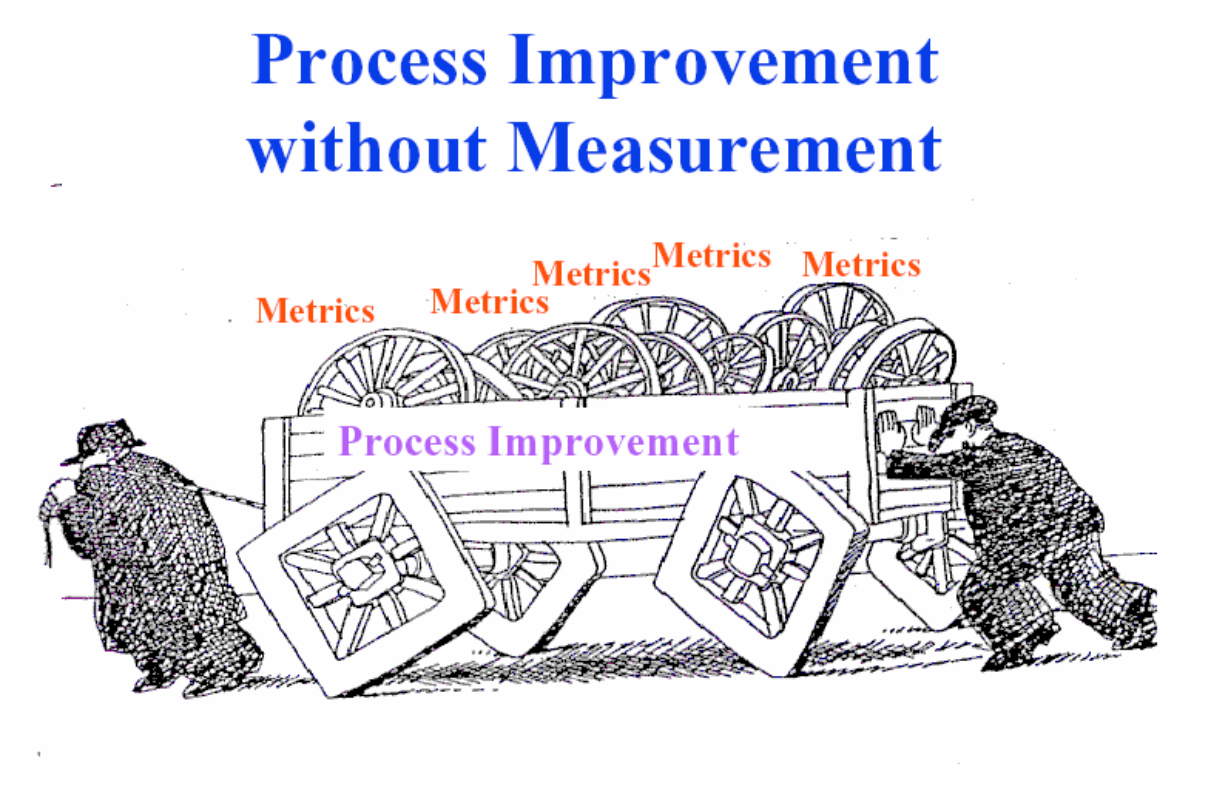
- Build models and investigate relationships based on management needs
- Derive software engineering models from measurement results of similar completed projects

Accents in achieving goals of measurements

1. Determining misunderstanding
2. Determining how to measure
3. Studying scientific approaches

Measurements for ***improving*** software process

Measurements for Improving Software Process



1. Discover problems and causes
2. Check improvements hypotheses

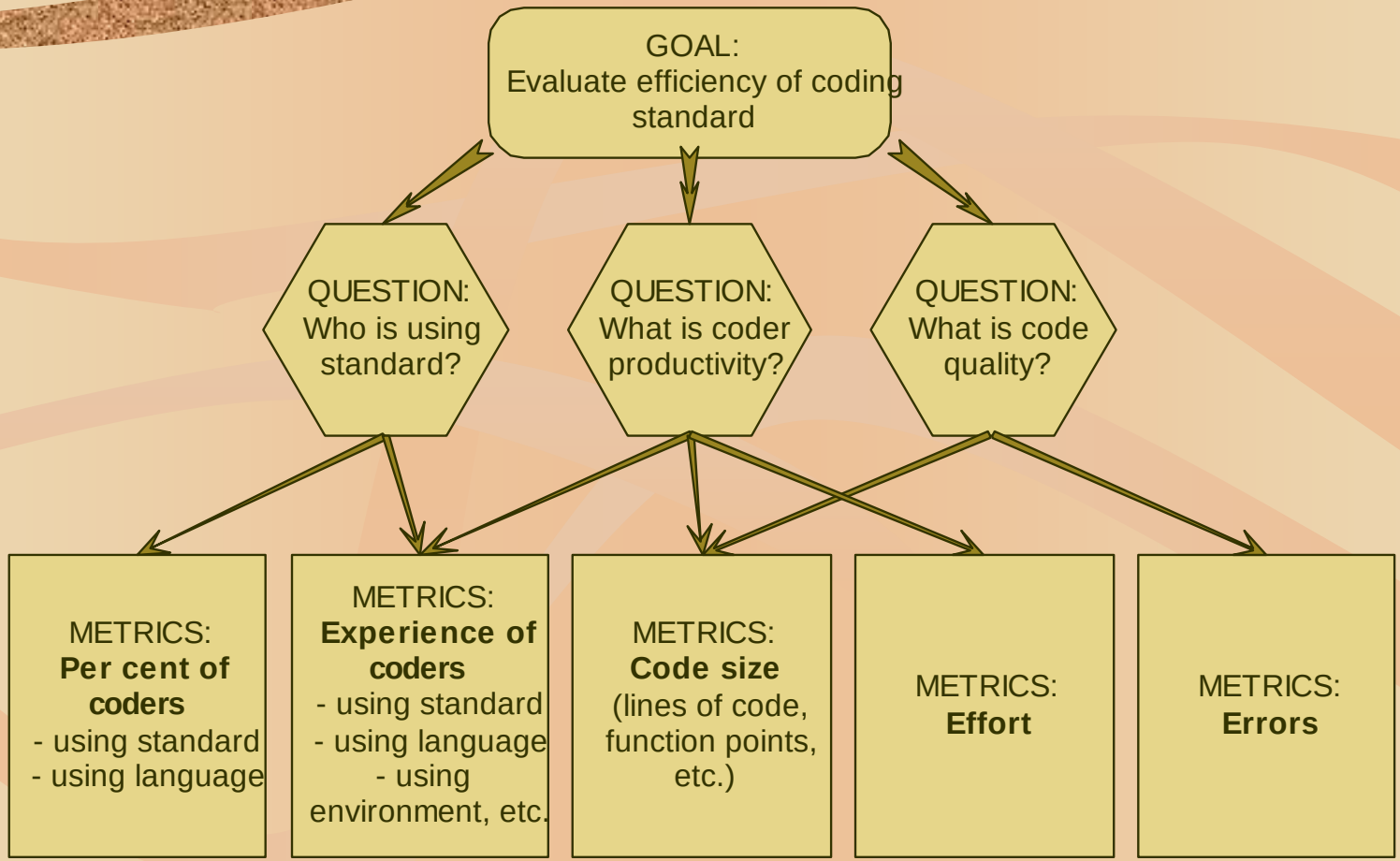
Measurements for Improving: *Discover problems*

- In order to understand "is a situation good or bad?" you should define what does "good" and "bad" mean
- Nobody can tell you what is "good" and "bad" for your organization
- You can compare your current level with your previous level, with world's famous standards, with level of your competitors, with level established by clients or top management

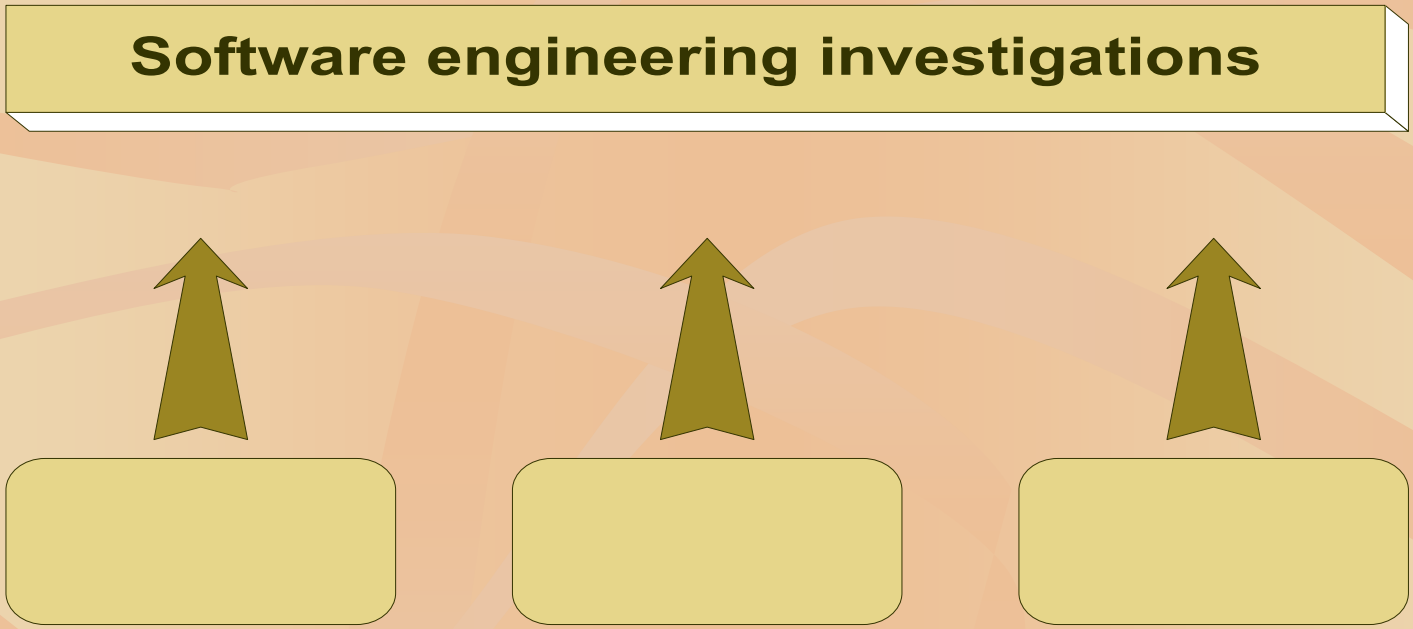
Measurements for Improving: *GQM approach*

1. Express the overall goals of your organization
2. Generate questions, answers to which may help you determine if your goals are met
3. Analyze each question from the perspective of measurements needed to answer it

Measurements for Improving: *GQM Example*



Measurements for Understanding and Improving: *Using investigations*



Accents in achieving goals of measurements

1. Studying scientific approaches
2. Finding resources

Measurements in *CMMI*

Process maturity levels and measurements

(based on work of Pfleeger and McGowan, 1990)

Level	Characteristics	Goals for measuring
Optimizing	Processes are continually improved based on a quantitative understanding of the common causes of variation inherent in processes	Make systematically improvements on ranking improvement proposals based on rigorous cost and benefits quantitative analysis
Quantitatively managed	Subprocesses are selected that significantly contribute to overall process performance. Selected subprocesses are controlled using quantitative techniques	Establish baselines and models and manage a process statistically
Defined	Standard processes are used to establish consistency across the organization. Projects establish their defined processes by tailoring the organization's set of standard processes	Establish measurement process for improving goals, try to determine what is "good" and what is "bad"
Managed	Projects have ensured that requirements are managed and that processes are planned, performed, measured, and controlled	Establish measurement process, determine metrics for management, build primitive baseline and primitive model
Initial	Ad hoc. Success depends on the competence and heroics of the people and not on the use of proven processes	-

Measurements in *CMMI*

- Process areas based on measurements: *Measurement and Analysis, Organizational Process Focus, Organizational Process Performance, Quantitative Project Management, Organizational Innovation and Deployment*
- Process areas use measurements: *Project Planning, Project Monitoring and Control, Configuration Management, Technical Solution, Product Integration, Organizational Training, Integrated Project Management, Risk Management, Decision Analysis and Resolution, Causal Analysis and Resolution*
- Common practices: *Monitor and Control the Process, Collect Improvement Information, Objectively Evaluate Adherence*

Measurement program

Your measurement program must specify:

- Measurement objectives
- Metrics
- Data collection procedure
- Data storage procedure
- Data analysis procedure
- Data distribution procedure
- Data using procedure

Summary

Let’s start from something easiest and try to understand obtained profits!