

Software Metrics

Software Engineering 2007

Anton Litvinenko

SourceKibitzer



SourceKibitzer





**“You can’t control what you can’t
measure”**

Tom DeMarco

“Controlling Software Projects”

Measurement

- Assignment of quantitative indications to product's attributes
 - *"Mapping from real world to numbers"*
 - Preserves empirical relation
- Quantitative attribute – *„objective“*
 - Variety of magnitudes
 - Stand in relation to one another
- Not quantitative – *„subjective“*
 - Feelings, emotions

Example

- Empirical relation
 - I am higher than Bill Gates



Example

- Empirical relation
 - I am higher than Bill Gates
- Measurement
 - My height is 192cm
 - Bill's height is 178cm

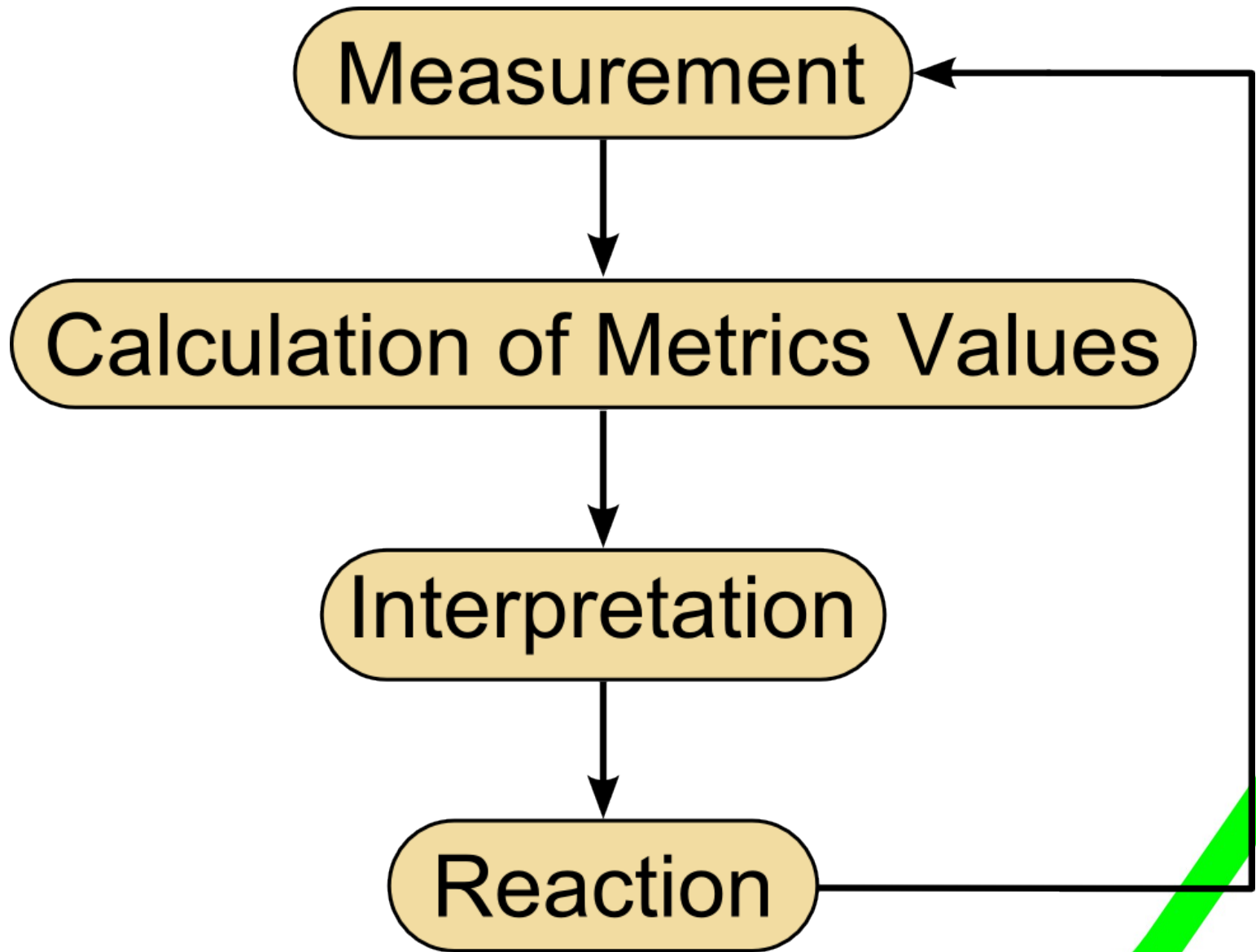


Metrics – **quantitative** representation
of otherwise vague attributes

GOAL – **IMPROVEMENT**

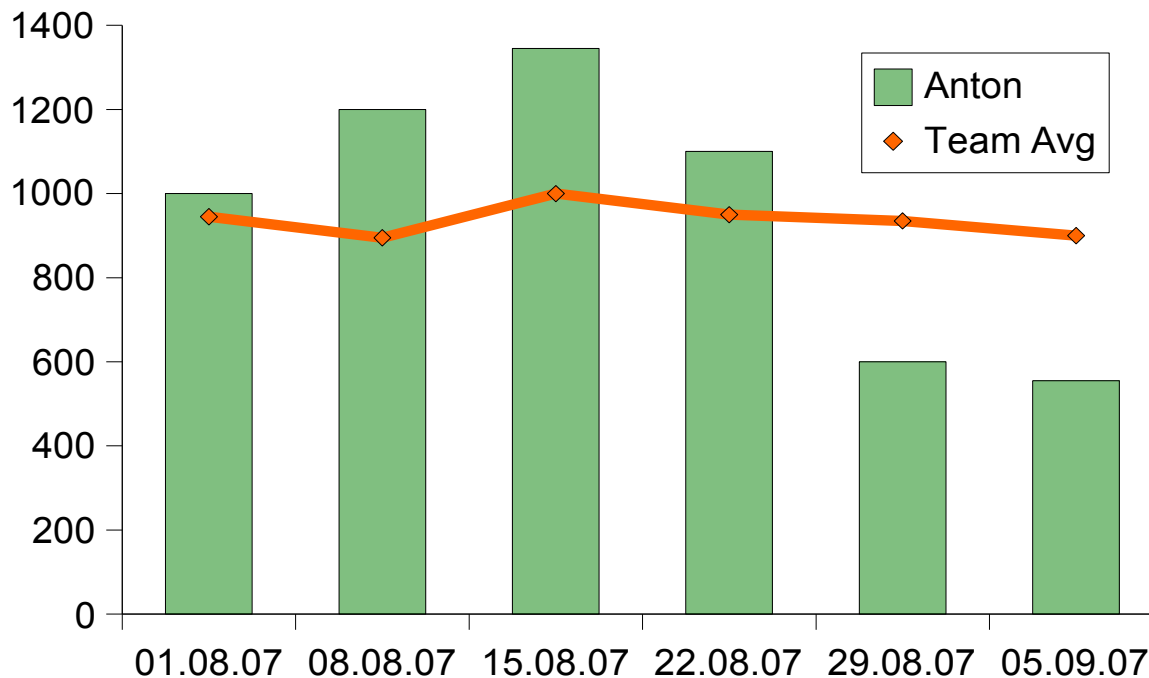
One-timer

- Contribution size
 - How much code developer has contributed during the last week
- 29. Aug. 2007 – Anton's contribution was 600 lines of code
 - What this number tells us?



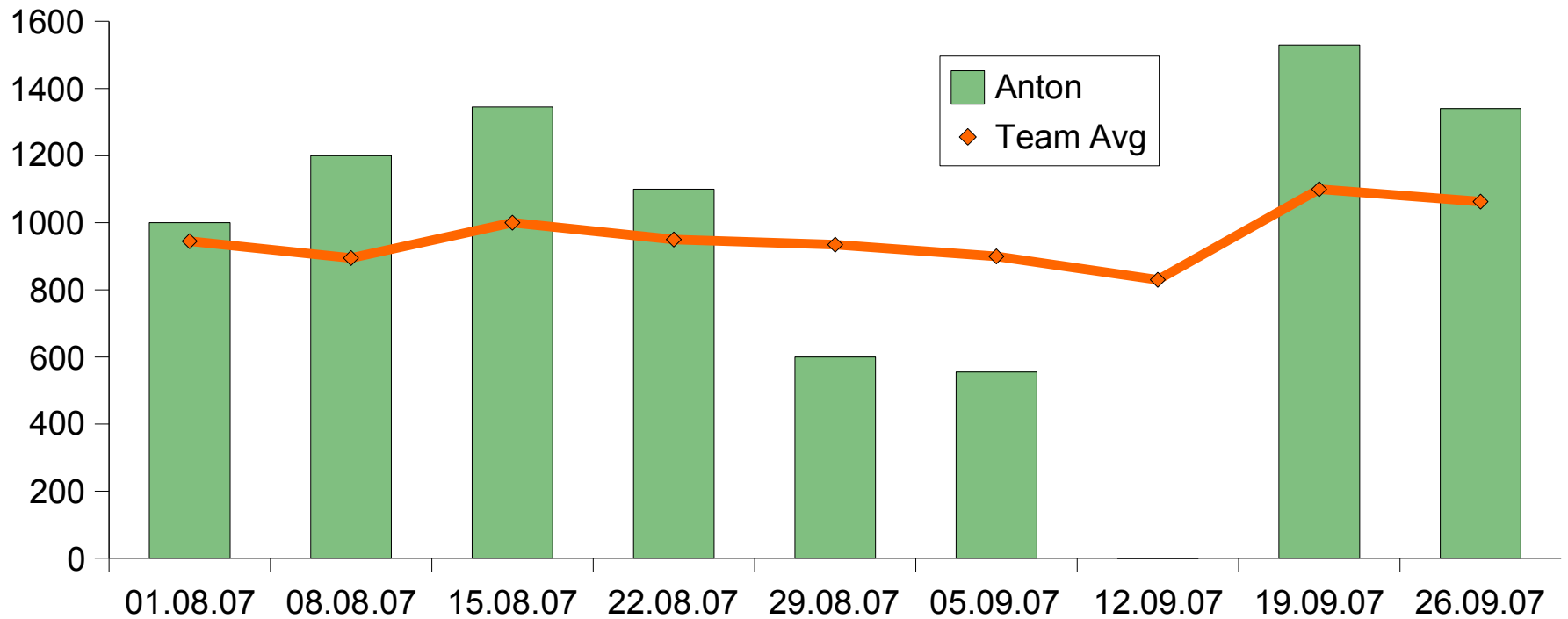
Iterative I

- Sequence of steps executed multiple times (*periodic, cyclic*)
 - *not iterative as iterative development*



Iterative II

- Gave Anton one week off (12.09.2007)



12/51

14.11.07

Software Metrics

Applications of Metrics

- Project progress estimation
 - Product size, velocity, ...
- Evaluation of tools and methods
 - Growth of productivity and quality, ...
- Product quality evaluation
 - Quality of design and code, number of bugs, ...
- Provide feedback to project team
 - Developer reputation, ...

What software metrics do you know?

Lines of Code

- Product size – LOC, SLOC, CLOC, ...

```
1 public static void main(String args[]) {
2
3     final SalesDomainController domainController =
4         new SalesDomainControllerImpl();
5
6     if (args.length == 1 && args[0].equals("console")) {
7         // a small console UI
8         ConsoleUI cui = new ConsoleUI(domainController);
9         cui.run();
10    } else {
11        // Swing UI
12        final SalesSystemUI ui = new SalesSystemUI(domainController);
13
14        ui.setVisible(true);
15    }
16
17    log.info("SalesSystem started");
18 }
```

Lines of Code II

```
1 public static void main(String args[]) {
2     /*
3      *   Creating instance of the controller.
4      *   Some more comments...
5      */
6     final SalesDomainController domainController =
7         new SalesDomainControllerImpl();
8     if (args.length == 1
9         && args[0].equals("console")) {
10        ConsoleUI cui =
11            new ConsoleUI(domainController); // a small console UI
12        cui.run();
13    } else {
14        // Swing UI
15        final SalesSystemUI ui =
16            new SalesSystemUI(domainController);
17        ui.setVisible(true);
18    }
19    log.info("SalesSystem started");
20 }
```


Lines of Code III

- Accurate, easy to measure
- How to interpret ...
 - Empty lines
 - Comments
 - Several statements on one line
- Depends on the language
- Doesn't respect complexity and content

Function Points

- Product size
- Use-Case evaluation
 - Each use-case screened for:
 - External Input: add new purchase
 - External Output: monthly purchase breakdown
 - External Inquiries: view client data
 - Internal Logical Files: DB table with client data
 - External Interface Files: warehouse interface
- For each such concept → complexity
 - Low, Average, High
- Number of F-points using formula

Function Points II

	Low	Average	High
External Input	_ × 3	_ × 4	_ × 6
External Output	_ × 4	_ × 5	_ × 7
External Inquiries	_ × 3	_ × 4	_ × 6
Internal Logical Files	_ × 7	_ × 10	_ × 15
External Interface Files	_ × 10	_ × 15	_ × 25
Unweighted Number of F-points:			Σ

Function Points II: Example

- Ökonoom saab hallata ladu
 - Võimalus lisada uut kaupa
 - Võimalus eemaldada kaupa süsteemist
 - Võimalus muuta kauba hinda, kogust ja nime

NB! Given example is simplified

Function Points IV

- Independent of programming language
- Can be applied earlier in the project
 - No source code needed
- Complexity estimation is subjective
 - Not accurate

Lines of Code and F-points

- Size and productivity estimation and evaluation
- Previous project
 - 1337 F-points and 24 man-months
- New project
 - 876 F-points -> ... man-months

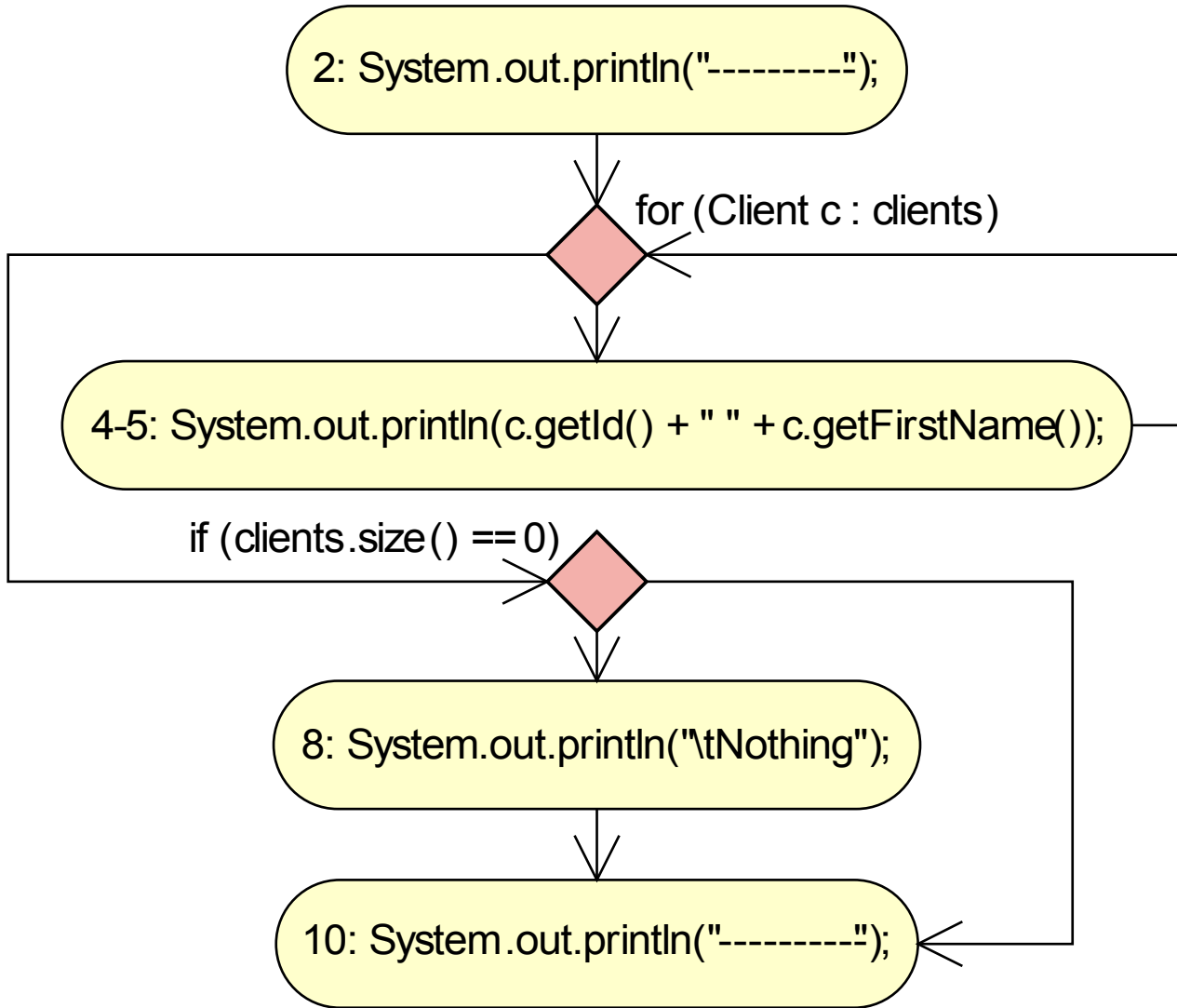
McCabe's Cyclomatic Complexity

- Thomas McCabe, 1976
- Measures complexity
 - Number of linearly independent paths through a program (or method)
 - Usually measured on the flow graph
- $V(G) = e - n + 2$
 - e – num of edges, n – num of vertices
- $V(G) = d + 1$
 - d – num of conditional statements
(if, for, throw(s), ...)

McCabe's Cyclomatic Complexity II

```
1 private void showClients(List<Client> clients) {
2     System.out.println("-----");
3     for (Client c : clients) {
4         System.out.println(c.getId() + " "
5             + c.getFirstName());
6     }
7     if (clients.size() == 0) {
8         System.out.println("\tNothing");
9     }
10    System.out.println("-----");
11 }
```


McCabe's Cyclomatic Complexity III



- $e = 7$
- $n = 6$
- $V(G) = 3$

McCabe's Cyclomatic Complexity IV

- Comprehensibility
 - $V(G) > 10 \rightarrow$ probability of bugs rises
- Relation to software testing
 - Lower bound for the number of paths through method
 - Upper bound for the number of branches in the method
- Doesn't respect other complexities
 - Data structure, data flows, interfaces

Software Metric Classification

Measurement Subject

Software Development Process

- Measuring the efficiency of process application
- Examples of metrics
 - Length of (development) iteration
 - Number of changes in requirements
 - Number of finished tasks

Resources

- Measuring resource usage and properties
- Examples of metrics
 - Personnel competency
 - Personnel fluctuation
 - Personnel know-how in the project*

Product

- Measuring product attributes
 - Size, complexity, performance, ...
- Examples of metrics
 - LOC, SLOC, CLOC, function points
 - (McCabe's) cyclomatic complexity
 - Code coverage (with tests)
 - Code stability*

Software Metric Classification II

Measuring vs Combining

Direct Metrics

- Directly measurable
- One measured attribute or entity
- Examples of Metrics
 - LOC, SLOC, CLOC
 - (entity: product)
 - Number of methods
 - Number of requirements
 - (entity: requirement specification)

Indirect Metrics

- Derived metrics – not possible to measure directly
- Combination of several metrics
 - Involves many attributes or entities
- Examples of Metrics
 - Requirement stability:
 - initial num of req.-s/total num of req.-s
 - Bug Density:
 - number of bugs/LOC

Software Metric Classification III

Internal vs External

Internal Attributes

- Measured in terms of only the entity itself
 - Measurement environment is not relevant
- Examples of Metrics
 - LOC, Density of Comments
 - Code coverage (with tests)
 - Code reuse

External Attributes

- Measured with respect to environment
- Examples of Metrics
 - Performance
 - Usability
 - Maintainability

Are software metrics good or evil?
Why?

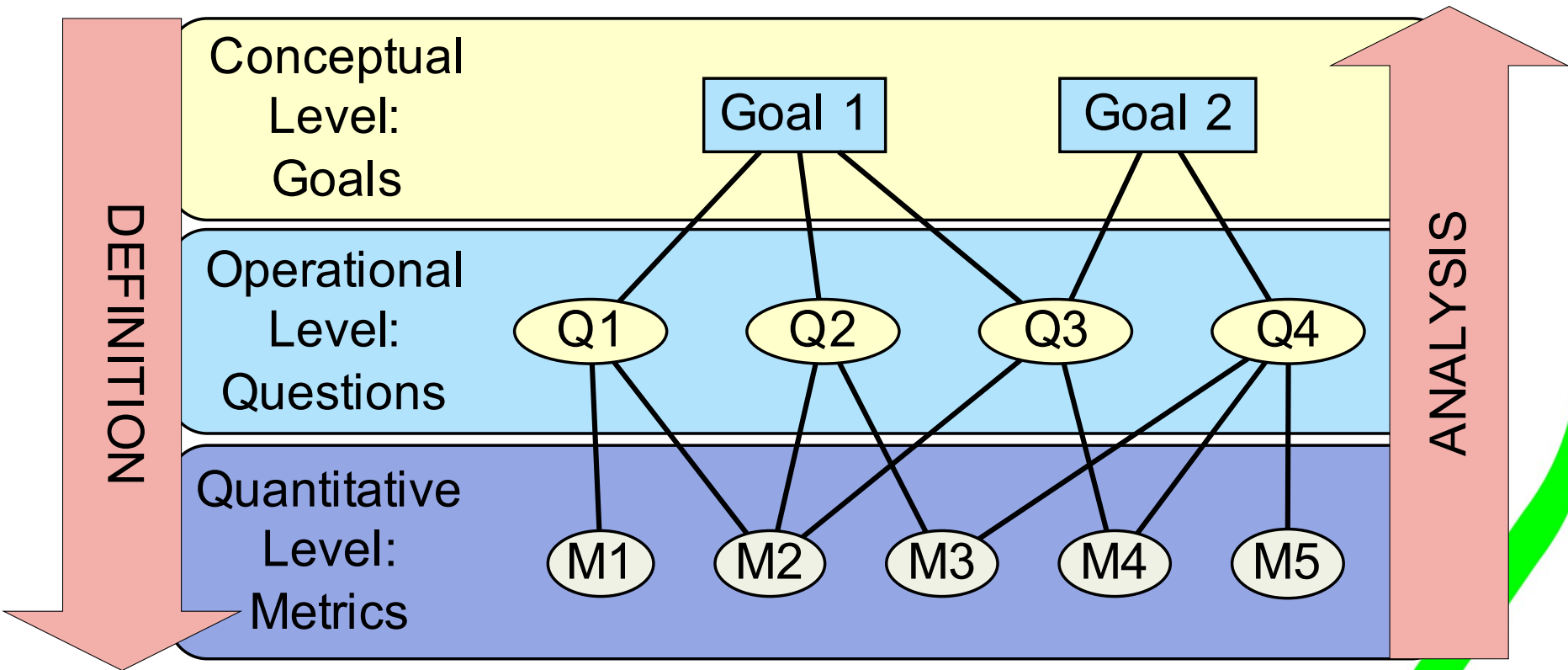
Critics

- People evaluation by numbers
 - Emotions, people adjust
- Easy to mistreat
- Either meaningful or accurate
 - Not both: LOC vs F-points
- Solution: set up clear goals and choose corresponding metrics

Software Metrics Frameworks

- GQM:
 - Goal – Question – Metric
- PSM:
 - Practical Software Measurement
- AMI:
 - Application of Metrics in Industry
- ...

Goal-Question-Metric Framework



GQM Exercise

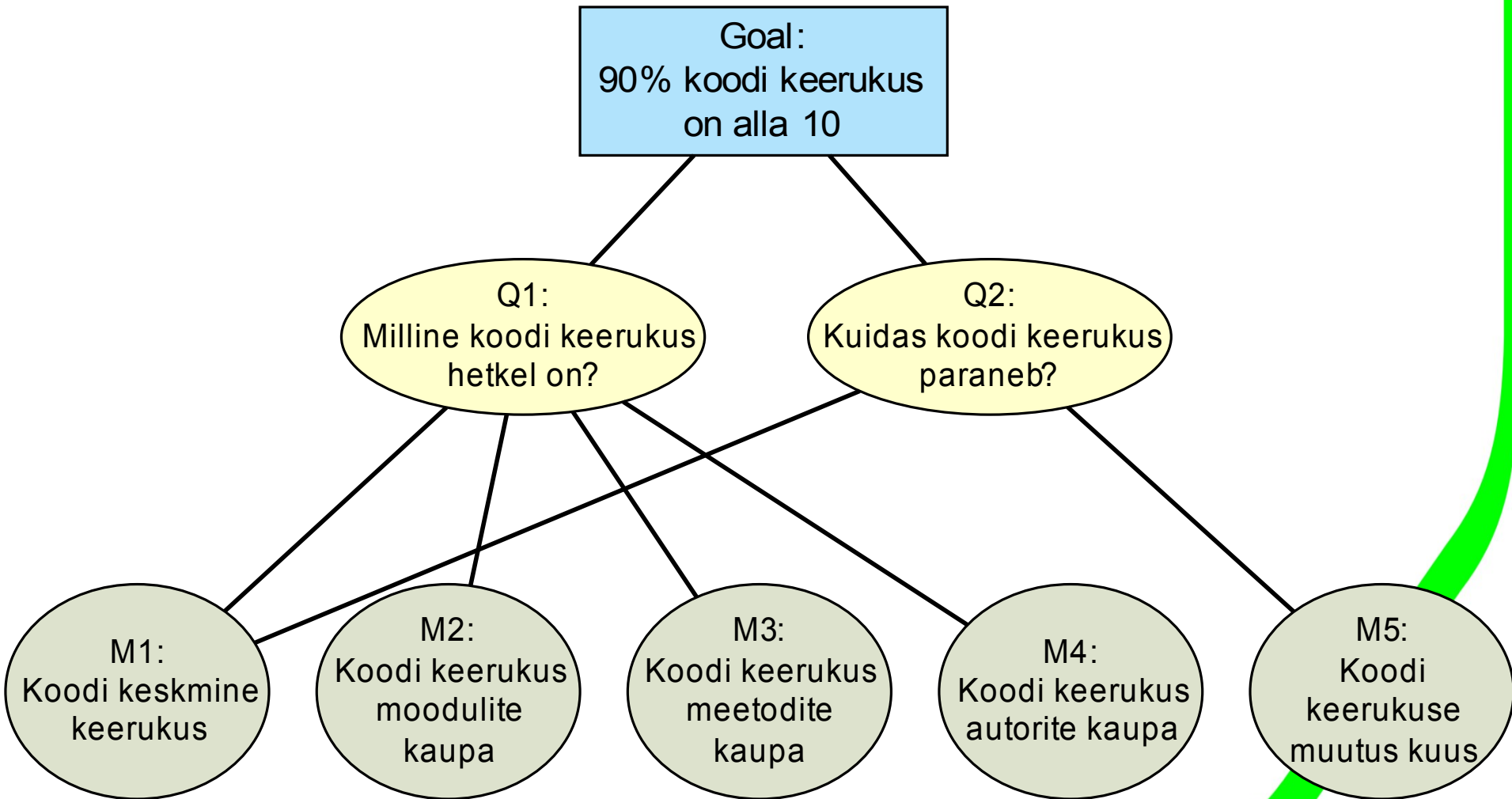
Any burning trouble in SE lab teams?

GQM Exercise

Any burning trouble in SE lab teams?

Some team members are much more committed than the others

GQM Naive Example



Some Tools

- Metrics for Java Open Source Teams*
 - <http://www.sourcekibitzer.org/>
- Metrics Eclipse Extension
 - <http://metrics.sourceforge.net/>
- JDepend
 - <http://clarkware.com/software/JDepend.html>
- Chidamber and Kemerer Java Metrics
 - <http://www.spinellis.gr/sw/ckjm/>
- RefactorIT
 - <http://refactorit.com/>
- ...

Exercise

- Calculate following metrics for the given code sample
 - LOC, SLOC
 - McCabe's Cyclomatic Complexity

CODE SAMPLE

- Solution
 - McCabe's Cyclomatic Complexity

FLOW GRAPH

Question

- Where is the highest probability of undetected bugs? (*Given that all modules are tested equally well*)
 - in modules showing **LARGEST** number of known bugs
 - in modules showing **SMALLEST** number of known bugs

Exercise

- Make up 10 metrics for evaluation of these slides:
 - Metric
 - Attribute of measurement
- Example
 - Number of slides -> *Presentation Size*
 - Ratio of pictures to the number of slides -> *Presentation Complexity*

Software Metrics

- Metric is ...
 - Goal: ...
 - The most important aspect of application: ...
- If software metrics topic seems interesting to you:
 - Anton Litvinenko
 - email: anton.litvinenko@sourcekibitzer.com
 - skype: [anton.a.litvinenko](https://www.skype.com/people/anton.a.litvinenko)

Please Keep in Mind

Software Metrics is a tool helping us to perform better

- BUT it is NOT a SILVER BULLET!
- You have to be careful and know the context of measurements
 - Otherwise you might end up with wrong interpretation of metrics values

References

- **Wikipedia**
 - http://en.wikipedia.org/wiki/Software_metric
- **C. Lange, Metrics in Software Architecting**
 - <http://www.win.tue.nl/~mchaudro/sa2007/Metrics%20Architecti>
- **M. Gökmen, Software Process and Project Metrics**
 - <http://www3.itu.edu.tr/~gokmen/SE-lecture-2.pdf>
- **H. Nestra, Metrics, Software Engineering 2005**
 - http://courses.cs.ut.ee/2005/tvt/uploads/Main/software_enginee
- **GQM Approach**
 - <http://www.goldpractices.com/practices/gqm/index.php>
- **More...**
 - <http://www.laynetworks.com/Software%20Engineering.htm>
 - <http://www.parlezuml.com/metrics/OO%20Design%20Principles>

References II

- **Lines of Code**
 - http://en.wikipedia.org/wiki/Source_lines_of_code
- **McCabe's Cyclomatic Complexity**
 - http://en.wikipedia.org/wiki/Cyclomatic_complexity
 - http://www.sei.cmu.edu/str/descriptions/cyclomatic_body.html
 - <http://www.stsc.hill.af.mil/crosstalk/1994/12/xt94d12b.asp>
- **Function Points**
 - http://www.csee.umbc.edu/~mikeg/cm645/function_point.html
 - [http://www.softwaremetrics.com/Function%20Point%20Training%](http://www.softwaremetrics.com/Function%20Point%20Training%20)
 - <http://www.codeproject.com/gen/design/Softwarecosting.asp>
 - <http://www.ifpug.com/fpafund.htm>