CS2 Software Quality

Models of Quality

Goal-Question-Metric

Defining Metrics

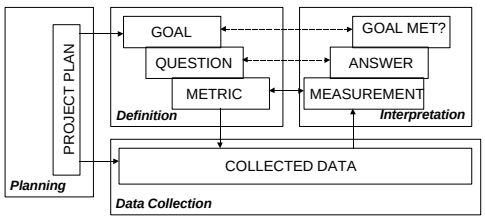
- Quality factors
  - ➪ FCM (McCall)
- Metrics
  - ➪ GQM (Basili)

Goals, Questions & Metrics

- ♦ List Goals
  - Purpose (what)
  - Perspective (who)
  - Environment (impact)
- ♦ Derive questions
  - How would I know if goals are being met?
- ♦ Measures to answer questions
  - all good properties of metrics
  - direct / indirect

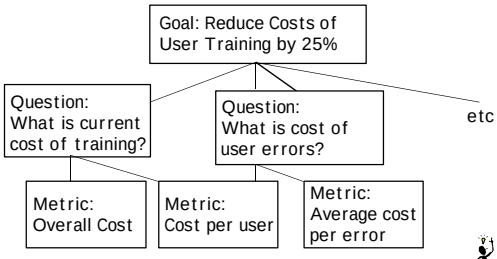
What makes a useful measure?

Objectivity	The results should be the same regardless of who takes the measure.
Reliability	The results should be precise and repeatable.
Validity	The metric must measure the required characteristic.
Standardisation	The metric must be unambiguous and allow for comparison.
Comparability	The metric must be comparable with other measures of the same criterion.
Economy	The measure should be simple and inexpensive to collect.
Usefulness	The measure must address a need, not simply measure a property for its own sake.
Consistency	The measure should be dimensionally consistent.
Automation	The measure should be capable of being collected automatically.



The 4 Phases of GQM

Goal-Question-Metric (GSM) Model



Goal Identification in GQM

Analyse	the "thing" under measurement
For the purpose of	understanding, controlling or improving the "thing"
With respect to	the quality focus of the "thing" that the measurement focuses on
From the viewpoint of	the people that measure the "thing"
In the context of	the environment in which the measurement takes place

RELIABILITY

Analyse	the delivered product and development process
For the purpose of	understanding
With respect to	reliability and its causes
From the viewpoint of	the project team
In the context of	project A

REUSE

Analyse	the delivered product
For the purpose of	understanding
With respect to	effectiveness of reuse
From the viewpoint of	the project team
In the context of	project A

Questions	**Metrics**
How many faults per module (total)?	Total number of faults per module
How many original faults per module?	Total number of original faults per module
How many faults introduced after modification?	Total number of faults introduced after modification
How much code in each module has been reused from other applications?	Percentage of code reused from other applications
Was the module code subject to peer review?	yes/no
What is the size of each module?	KSLOC
How complex is each module?	McCabe's Cyclomatic complexity

Data sorted by number of faults

Module	# Faults	Fault Ranking	Modification Faults	Original Faults	Reuse	Review	Size (KSLOC)	Complexity
1	17	1	10	7	> 20%	yes	4.56	624
2	13	2	8	5	> 20%	yes	3.10	560
3	12	3	4	8	> 20%	yes	4.10	624
4	8	4	0	8	none	no	2.10	207
5	8	4	0	8	none	no	0.68	210
6	7	6	0	7	none	yes	2.10	245
7	6	7	0	6	none	yes	0.50	70
8	5	8	3	2	>20%	yes	4.68	420
9	4	9	0	4	none	yes	0.46	55
10	4	9	0	4	none	yes	0.45	8
11	4	9	0	4	< 20%	no	0.39	39
12	3	12	1	2	< 20%	yes	1.80	303
13	3	12	0	3	none	no	2.24	309
14	3	12	0	3	none	no	1.20	150
15	3	12	0	3	none	yes	1.20	132

Data sorted by number of original faults

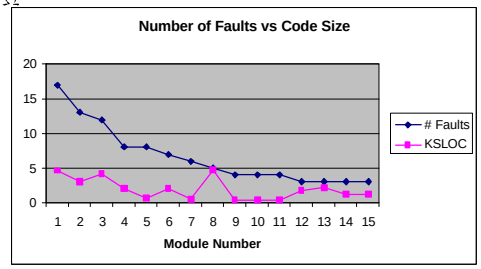
Module	# Faults	Fault Ranking	Modification Faults	Original Faults	Reuse	Review	Size (KSLOC)	Complexity
3	12	3	4	8	> 20%	yes	4.10	624
4	8	4		8	none	no	2.10	207
5	8	4		8	none	no	0.68	210
1	17	1	10	7	> 20%	yes	4.56	624
6	7	6		7	none	yes	2.10	245
7	6	7		6	none	yes	0.50	70
2	13	2	8	5	> 20%	yes	3.10	560
9	4	9		4	none	yes	0.46	55
10	4	9		4	none	yes	0.45	8
11	4	9		4	< 20%	no	0.39	39
13	3	12		3	none	no	2.24	309
14	3	12		3	none	no	1.20	150
15	3	12		3	none	yes	1.20	132
8	5	8	3	2	>20%	yes	4.68	420
12	3	12	1	2	< 20%	yes	1.80	303

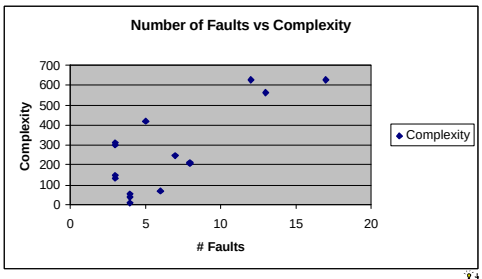
Data sorted by number of modification faults

Module	# Faults	Fault Ranking	Modification Faults	Original Faults	Reuse	Review	Size (KSLOC)	Complexity
1	17	1	10	7	> 20%	yes	4.56	624
2	13	2	8	5	> 20%	yes	3.10	560
3	12	3	4	8	> 20%	yes	4.10	624
8	5	8	3	2	>20%	yes	4.68	420
12	3	12	1	2	< 20%	yes	1.80	303
4	8	4		8	none	no	2.10	207
5	8	4		8	none	no	0.68	210
6	7	6		7	none	yes	2.10	245
7	6	7		6	none	yes	0.50	70
9	4	9		4	none	yes	0.46	55
10	4	9		4	none	yes	0.45	8
11	4	9		4	< 20%	no	0.39	39
13	3	12		3	none	no	2.24	309
14	3	12		3	none	no	1.20	150
15	3	12		3	none	yes	1.20	132

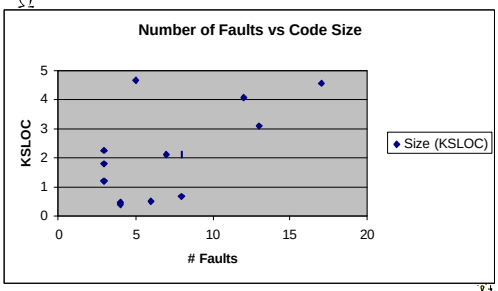
Data sorted by reuse and number of original faults

Module	# Faults	Fault Ranking	Modification Faults	Original Faults	Reuse	Review	Size (KSLOC)	Complexity
4	8	4		8	none	no	2.10	207
5	8	4		8	none	no	0.68	210
6	7	6		7	none	yes	2.10	245
7	6	7		6	none	yes	0.50	70
9	4	9		4	none	yes	0.46	55
10	4	9		4	none	yes	0.45	8
13	3	12		3	none	no	2.24	309
14	3	12		3	none	no	1.20	150
15	3	12		3	none	yes	1.20	132
8	5	8	3	2	>20%	yes	4.68	420
3	12	3	4	8	> 20%	yes	4.10	624
1	17	1	10	7	> 20%	yes	4.56	624
2	13	2	8	5	> 20%	yes	3.10	560
11	4	9		4	< 20%	no	0.39	39
12	3	12	1	2	< 20%	yes	1.80	303





Correlation = 0.7719



Correlation = 0.6361

Goal-Question-Metric (GSM) Model

Goal: Reduce Module Testing Time

Questions …

Metrics …

Other Sources:

Rini Van Solingen & Egon Berghout, *The Goal/Question/Metric Method*, McGraw-Hill, 1999.

http://www.gqm.nl

Watts Humphrey, *A Discipline for Software Engineering*, Addison-Wesley, 1995.

Brian Hambling, *Managing Software Quality*, McGraw-Hill, 1996.

Stephen Kan, *Metrics and Models in Software Quality Engineering*, Addison-Wesley, 1995.

http://www.adeptsys.com/Pages/goal_questions_metric.html

http://www.iese.fhg.de/Services/Projects/Public-Projects/Perfect/GQM/GQM.html

Yasuhiro Mashiko & Victor R. Basili, "Using the GQM Paradigm to Investigate Influential Factors for Software Process Improvement", *Journal of Systems and Software*, Vol. 36, No. 1, January 1997.