Using Quality Factors and Measures to Focus on Customer Satisfaction

Karen L. Owens and Martha Ann Griesel

For Software Engineering Process Group National Meeting
Costa Mesa, California
April 28-29, 1993

JPL
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109
(818) 306-6205

JPL

Acknowledgements

- This work was done through an agreement with the National Aeronautics and Space Administration, by Jet Propulsion Laboratory California Institute of Technology Pasadena, California

- We thank
  - Barbara Anderson, Tal Brady Mickey Bramble, Mark Francis, Dorothy Huffman, Shirlev Jeane, Cindv Kahn, Brian Larman, Marilyn Oifer, Melvin Pinck, Bill Robison, and Rex WeUs for their participation in discussing and ranking the quality factors and their patience through several iterations.
  - Tom Fouser, Allen Nikora, and Fra: ık Singleton for reviewing related work.

Why Force Developers to Think About Quality, Factors?

- Defining quality factors helps develop understanding of customer needs
- Attending to quality factors results in capturing quality requirements and design goals in the Software Management Plan (e.g., 2167A Software Development Plan) and in the Software Requirements Document
- Using measures for quality factors supports follow up (getting quality engineered into products)
  - o Where are we now? Where do we want to go? Are we getting there?

This presentation reports the experiences of 2 teams

Team Backgrounds

- Team A
  - Participants included Software Manager, Lead Engineers, Software Product Assurance, and SEPG member
  - Project characteristics
    - Onboard software for robotic spaceflight mission with many instruments and many critical events

- Team B
  - Participants included Manager, System Engineer, Test Engineer, Configuration Manager, Software Quality Engineer, SEPG members
  - Project characteristics
    - Ground data system software supporting many missions
    - Large inheritance, multiple platforms over history
    - Greater portability is needed along with new capabilities

Quality Goals of Each Team

- The Team A goal was to pre-rank the quality factors so that the lead engineers could then use the ranked quality factors as criteria in making design trade-off decisions.

- Team B goals were to
  - Improve quality within their budget and schedule
  - Measure current quality
  - Determine how to allocate resources based on current quality
  - Give rewards based on quality factor improvements

JPL

Generic Process

- Assemble the team members with all areas and disciplines represented

- Rank the quality factors

- Propose candidate measures

- Determine cost and values of the supporting measures

- Select the measures
  - o Collect, analyze, and refine the measures
  - o Improve the process
  - o Collect, analyze, and refine the measures based on their costs and values

JPL

Rank the Quality Factors

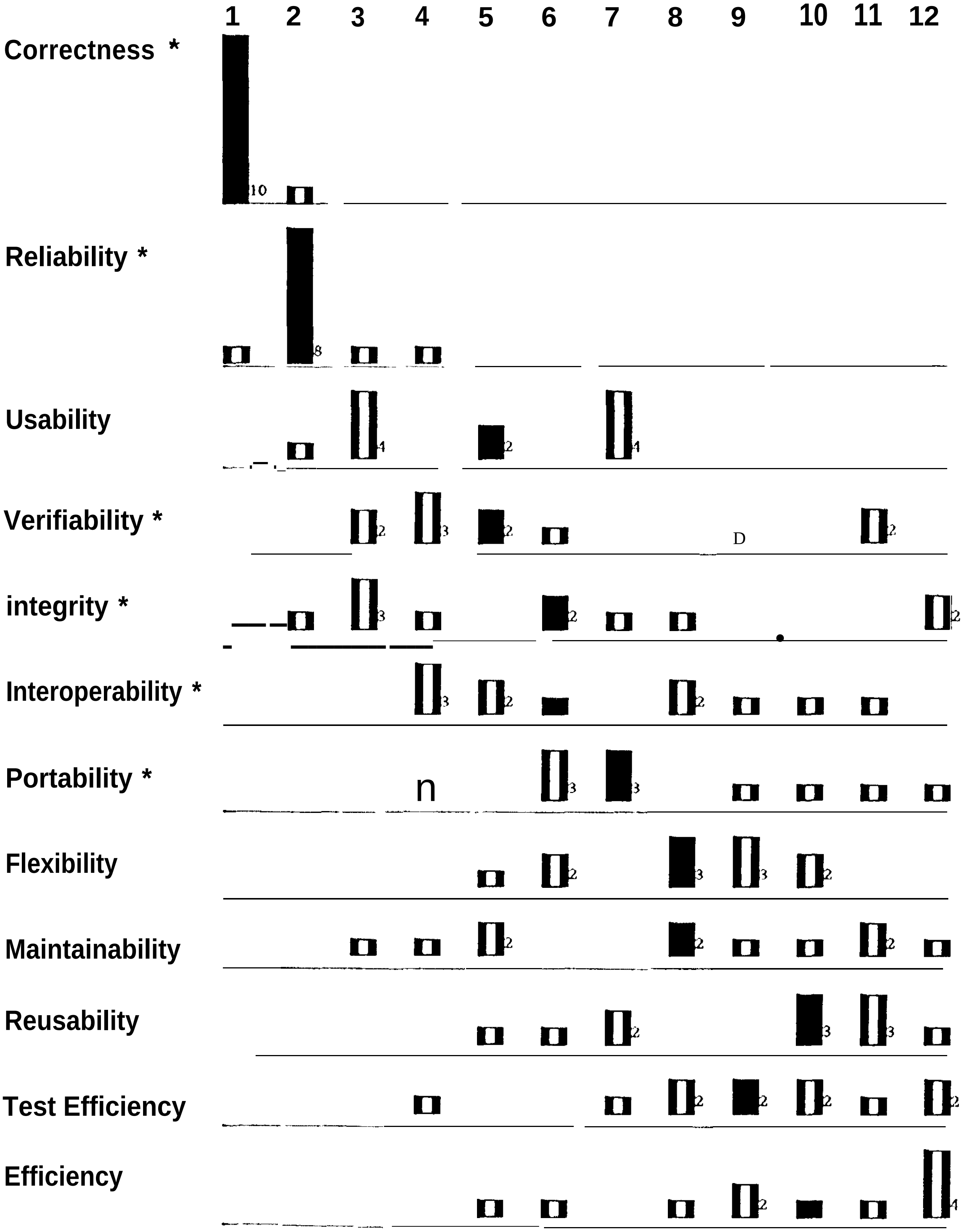
- Determine the definitions for your customer
  - o Discuss the book definitions (e.g., Deutsch and Willis)
  - o Customize those definitions
  - o Add other quality factors or delete those irrelevant to the customer

- Each individual ranks the quality factors
  - o Sometimes team members bring consensus from their areas back to team
  - o Skipping this step can incline group to follow the leader

- Display the range of votes
  - o Look at the minimum, the maximum, the median, and the modes

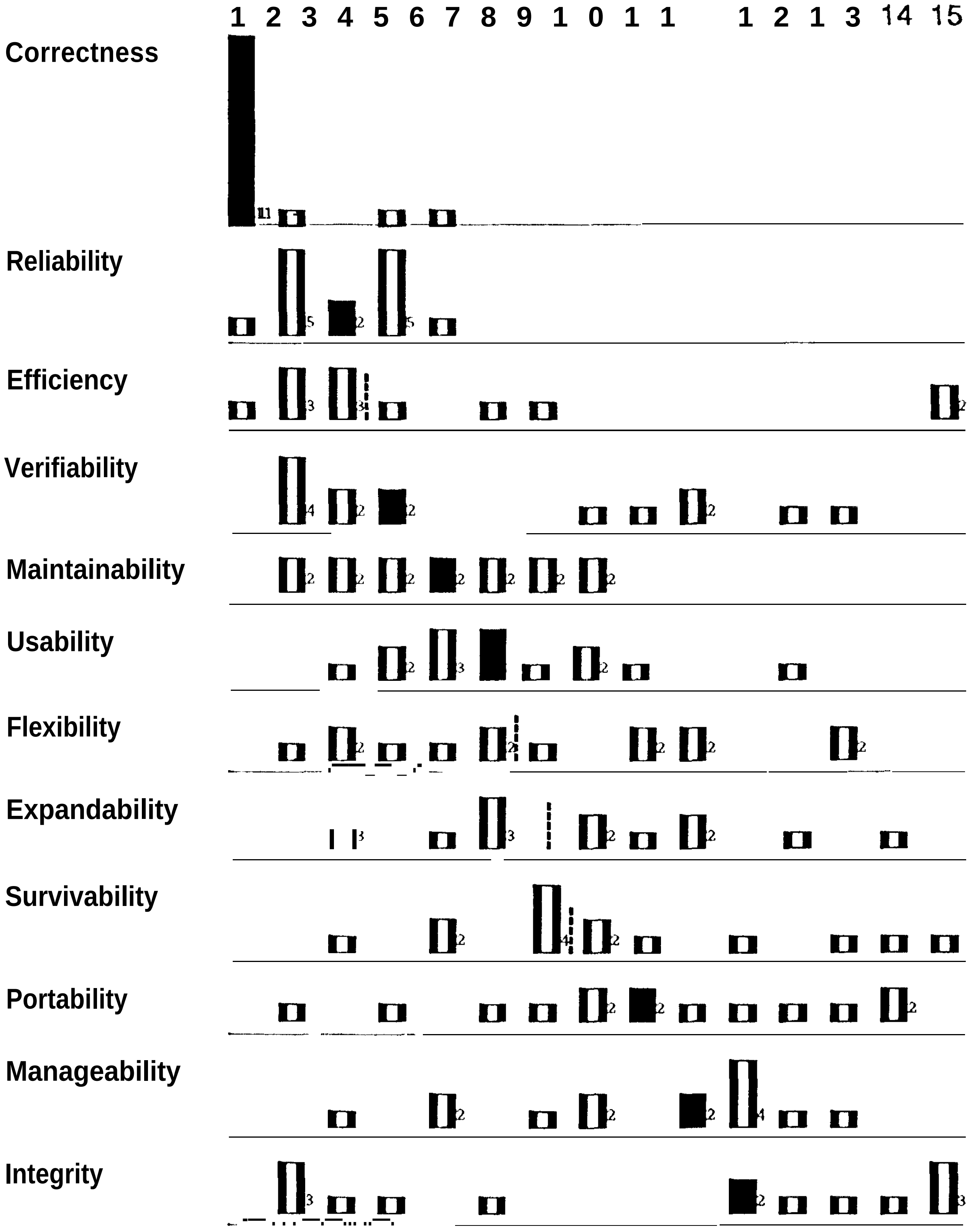
JPL

Team B -- Ranking Summary



Solid bars and dotted lines represent medians. * indicates required quality factor,

Diverse Students -- No Discussion



Solid bars and dotted lines represent medians. * indicates required quality factor,

Use Patterns to Focus Discussion

- Arrange counts for each quality factor by decreasing medians to see patterns

- Is apparent agreement real? (e.g., Correctness)

- Why did people hold split views? (e.g., Usability)

  o Are different people using different definitions?

- Why were views so distributed? (e.g., Maintainability)

  o Is same definition applied differently by different people to the project?

- Do close "averages" mean equally important factors?

  0 1234 . . . . 5678 . 0 . 0 .0.0...0. . 00 . . . .0... 9101112

  0 Are some quality factors requirements and others are design goals?

Propose Candidate Measures: Some Examples

- #Errs. In. Interface.Test = Number of errors found during interface testing
- #Errs.by.Phase.Found = Number of errors by phase found
- #Extrnl.Interfcs.to. Test = Number of external interfaces to test
- #Failure. Reports = Number of failure reports (FRs) per time period (e.g., month or delivery)
- #FRs.Call.Cockpit.Errs = Number of failure reports called cockpit errors
- #Unt.W/NStd.Lang. Feat = Number of units using non-standard language features

Example Scales for Determining Cost and Value

- cost
  - o □ LOW if it is something we already have or do
  - o ⬓ MEDIUM if is something not done now, but easy and well understood
  - o ■ HIGH if it is something new or something hard to do

- Value scales roughly with the strength of the correlation between the quality factor and the measure
  - o ■ HIGH if the measure is directly related to the quality factor
  - o ⬓ MEDIUM if it is one degree removed from the quality factor (e.g., Complexity correlates with Reliability)
  - o □ LOW if it is more than one degree removed from the quality factor

Determine Costs and Values

- Assess COSTS before VALUES
    - It is easier and gives an established scale to work against

- Pre-Assess VALUES
    - Discuss the meanings of the values with whole team
    - Two people each make an initial assessment of the values, then discuss the values with each other
    - Adjust the values and record differences of opinion (e.g., HIGH/MED)

- Re-Assess VALUES
    - Present preliminary values and their reasons to the group
    - Discuss and modify the pre-assessed values
    - Iterate as many times as it takes to reach consensus

JPL

Some Costs and Values According to Team B

Values Corresponding to Quality Factors

cost	Measure	CORR	RELI	USA	VERIF	INTEG	INTROP	PORT	FLEX	EXPAN	MAINT	REUSE
		▼	▼	v	▼	▼	▼	▼	▼	▼	▼	▼
❑	* #Errs. In. interface.Test	❑	■		⬓	.	■			❑	□	□
❑	#Errs. by. Phase. Found	■	■	.	.	.	.	.	.	.	□	□
■	#Errs. by. Phase. Injected	■	■	.	.	.	.	.	.	□/⬓	□/⬓	□/⬓
❑	* #Extrnl.Interfcs.to. Test	■	■	⬓	■	❑	❑		.	❑	⬓	❑
I - I	* #Failure. Reports	■	■	⬓	□	.	❑			|	.	|
□/⬓	* #FRs.Call.Cockpit. Errors	.	□	■	⬓	.	.	.	.	⬓	⬓	⬓
■	McCabe. Complexity/Unit	❑	⬓	.	■	.	.	.	n ⬓	⬓	■	⬓
❑	* %Reqs.Dmd.Testabl	■	.		■	.	•1	.	.	❑	⬓	❑
❑	* %Reqs.Wi.Tst.Cases	■	□	❑	■	❑	ı	.	□	❑	■	□

Costs and Values According to Team B

Values Corresponding to Quality Factors

		CORR	RELI	USA	VERIF	INTEG	INTROP	PORT	FLEX	EXPAN	MAINT	REUSE
Cost	Measure	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼
❑☛	%Comments/Unit	.	.	.	❑	.	.	□	❑	❑	■	■
❑	#Errs. Fnd/WF?by.Phs.Fnd	■	■	.	.		.		.	.	❑	❑
■	#Errs. Fnd/WP.by.Phs.Inj	■	■	.		.			.	□/⬓	□/⬓	□/⬓
⬓/■	#Errs. Forecast. Tot	.	■	.	⬓		.	.		.	.	
❑ *	#Errs.in.interface.Test	■	■		⬓		■		.	□	❑	□
❑ *	#Externi.interfaces. to.Test	■	■	⬓	■	❑	❑		.	❑	⬓	❑
❑ *	#Failure. Reports	■	■	❑	u	.	❑	.	.	.	.	.
□/⬓*	#FRs.Call.Cockpit. Errors	\	□	■	⬓	.		.	.	❑	⬓	W
❑	#Internl.interfaces.to. Test	■	■	.	■		.	.	.	□	❑	❑
■	McCabe. Complxty/Unit	❑	⬓	.	■	.	.	.	❑	⬓	■	⬓
u *	#NonPortablty. Msgs/Unit		.	.		.	□	■	❑	❑	⬓	❑
❑	#NSLOC/Unit	❑	□	.	❑		.	ı	❑	□	⬓	❑
•1	#NSLOC/Program	ı	□	.	❑	.	.	ı	.	❑	❑	⬓
U	Operator. Actions/Task	ı	□	■	⬓	.	⬓		.	ı	❑	
□	#Progs.In.Mixed. Langs	ı	.	.	⬓	.	□	■	⬓	❑	■	⬓
❑ *	%Reqs.Dmd.Testabl	■	.	.	■	.	❑	ı	.	❑	❑	⬓
❑ *	%Reqs.Wi.Tst.Cases	■	□	❑	❑	❑	.	ı	•1	❑	■	❑
□	#Units/Program	ı	□	.	W	.	.	ı	❑	⬓	⬓	⬓
□	#Unts.Over.NSLOC. UCL	❑	□	.	⬓	.	.	ı	.	⬓	⬓	❑
□ *	#Unts.Wi.NStd.Lang. Feat	ı	.	.	❑	.	⬓	■	❑	⬓	■	■

For Cost: ■ (HIGH), ❑ (Meal), or ❑ (low) represents the cost of obtaining the measurement.
For Value: ■ (HIGH), ⬓ (Meal), or ❑ (low) indicates the value of how strongly the measure supports the quality factor.
"." indicates no recommended value of measure. "/" shows range. "*" indicates selected measures.

JPL

In Conclusion

- Defining quality factors and measures helped the teams understand their customers' needs and make better trade-off decisions

"You Can't Achieve ≤uality ... Unless You Specify It!"

--Michael Deutsch and Ronald Willis in "Software Quality Engineering", Pg. 8

JPL

References

Deutsch, Michael and Willis, Ronald R., "Software Quality Engineering: A Total Technical and Management Approach", Prentice-Hall, New Jersey 1988.

Gilb, Tom, with Finzi, Suzannah, "Principles of Software Engineering Management", Addison-Wesley 1988.

McCall, J.A. and Matsumoto, M, "Software Quality Metrics Enhancements, Vol. I", Rome Air Development Center, 1980, Chapter 8 (An Introduction to Software Quality Metrics).

"Using Quality Factors and Measures to Focus on Customer Satisfaction", Version 1.0, SORCE Technical Memorandum 14, JPL internal document, 1993.