Decision Matrix & Product Metrics

Jamal Madni

CECS 445

Lecture 4: February 4th, 2021

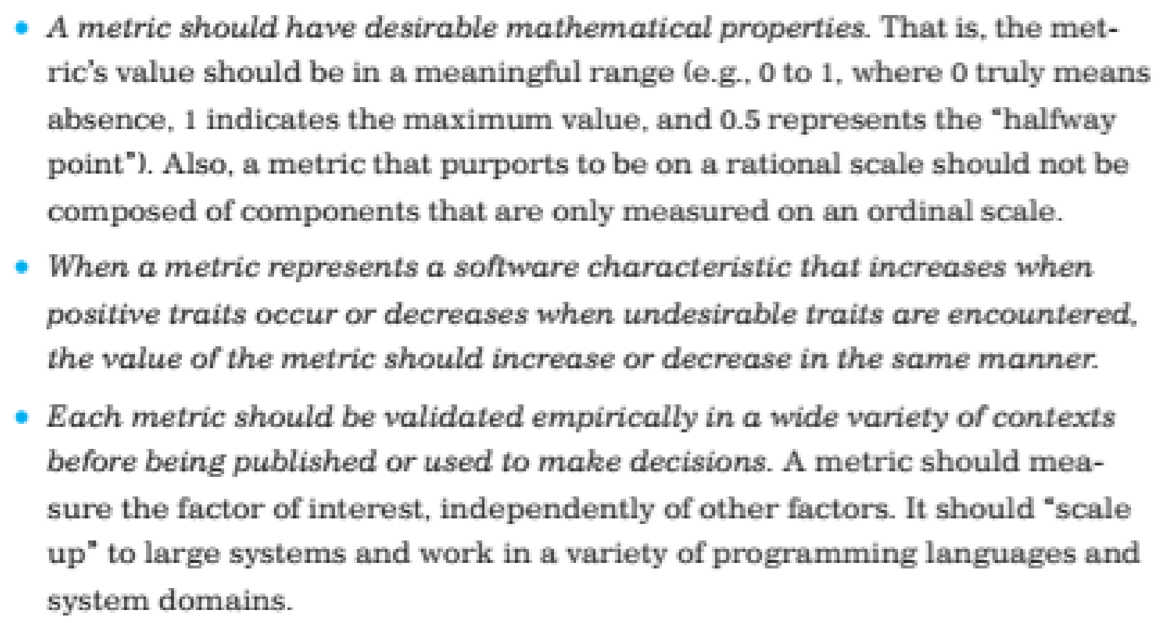
3 = Yes	
2 = Maybe	
1 = No	

Deciding On Your Project…

Team Member	**Scope** (6 pts max per person)	**Access** (6 pts max per person)	**Interest** (6 pts max per person)	**Total** (18 pts per person)
	Is the project achievable? Is the project modular?	Is your point of contact available every two weeks? Is your point of contact a decision-maker in the org?	Are you excited about the topic and/or organization? Are your skills being matched well to your piece?	Scope + Access + Interest

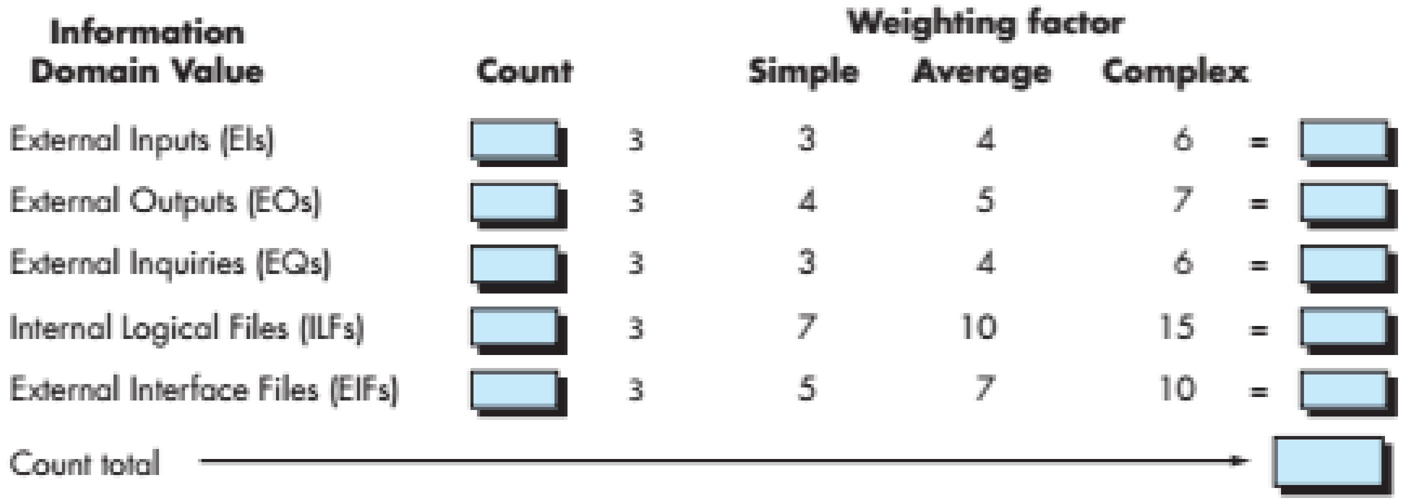
Product Metrics

YOUR USER REQUIRE-MENTS INCLUDE FOUR HUNDRED FEATURES.
DO YOU REALIZE THAT NO HUMAN WOULD BE ABLE TO USE A PRODUCT WITH THAT LEVEL OF COMPLEXITY?
GOOD POINT. I'D BETTER ADD "EASY TO USE" TO THE LIST.
www.dilbert.com
scottadams@aol.com

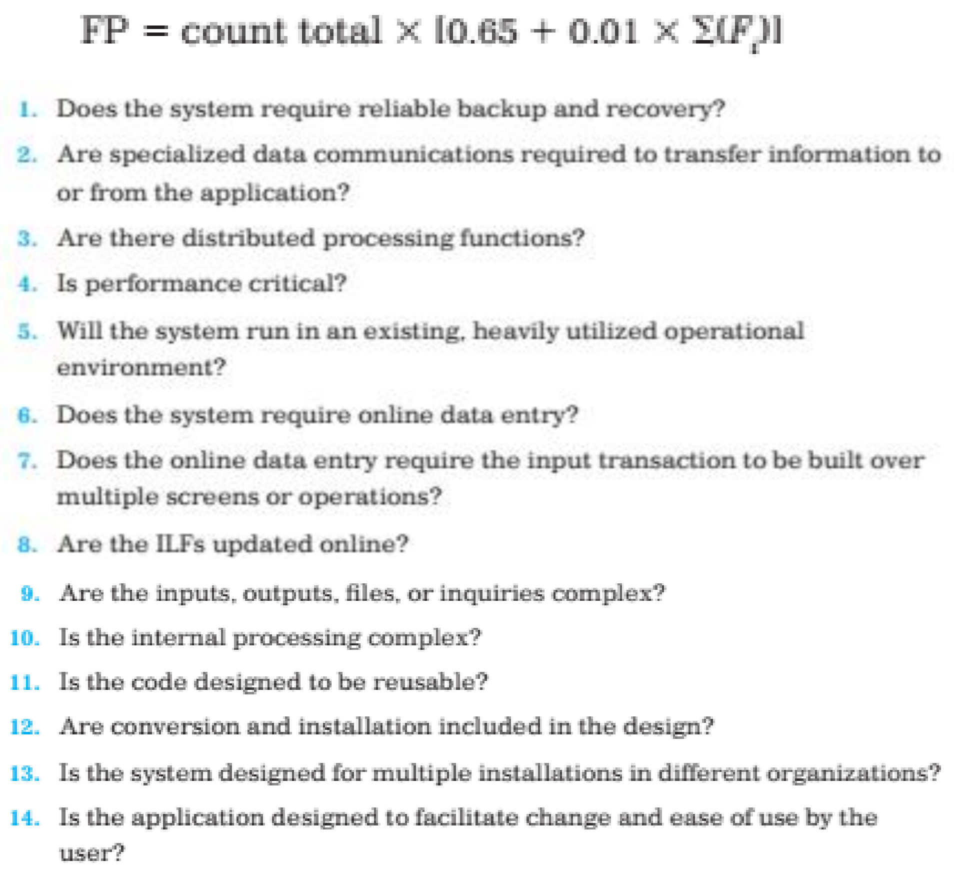


- *A metric should have desirable mathematical properties.* That is, the metric's value should be in a meaningful range (e.g., 0 to 1, where 0 truly means absence, 1 indicates the maximum value, and 0.5 represents the "halfway point"). Also, a metric that purports to be on a rational scale should not be composed of components that are only measured on an ordinal scale.
- *When a metric represents a software characteristic that increases when positive traits occur or decreases when undesirable traits are encountered, the value of the metric should increase or decrease in the same manner.*
- *Each metric should be validated empirically in a wide variety of contexts before being published or used to make decisions.* A metric should measure the factor of interest, independently of other factors. It should "scale up" to large systems and work in a variety of programming languages and system domains.

Metrics For Requirements



Information Domain Value	Count		Weighting factor Simple	Average	Complex		
External Inputs (EIs)		3	3	4	6	=	
External Outputs (EOs)		3	4	5	7	=	
External Inquiries (EQs)		3	3	4	6	=	
Internal Logical Files (ILFs)		3	7	10	15	=	
External Interface Files (EIFs)		3	5	7	10	=	
Count total							

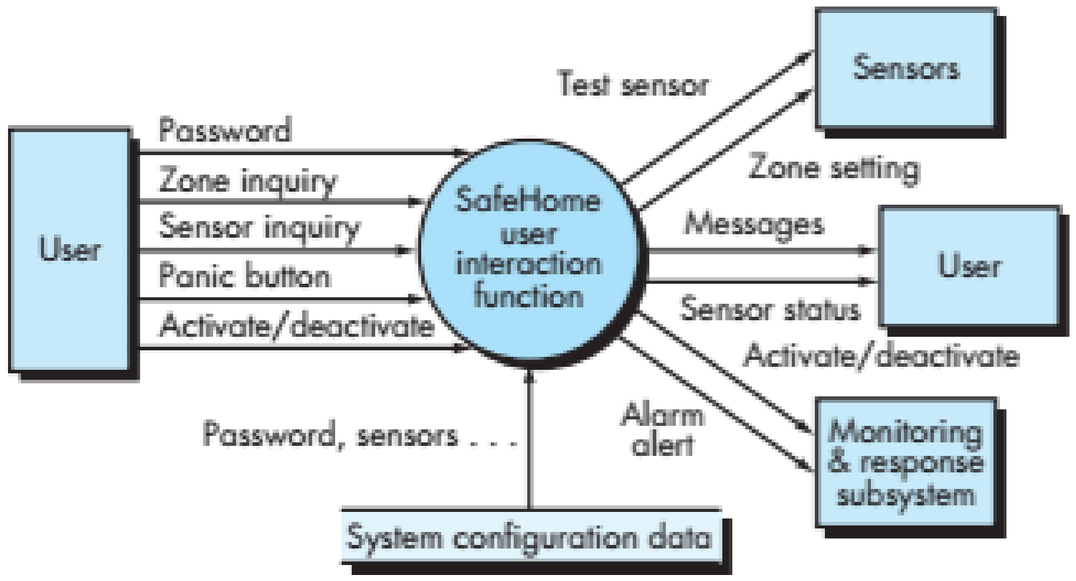


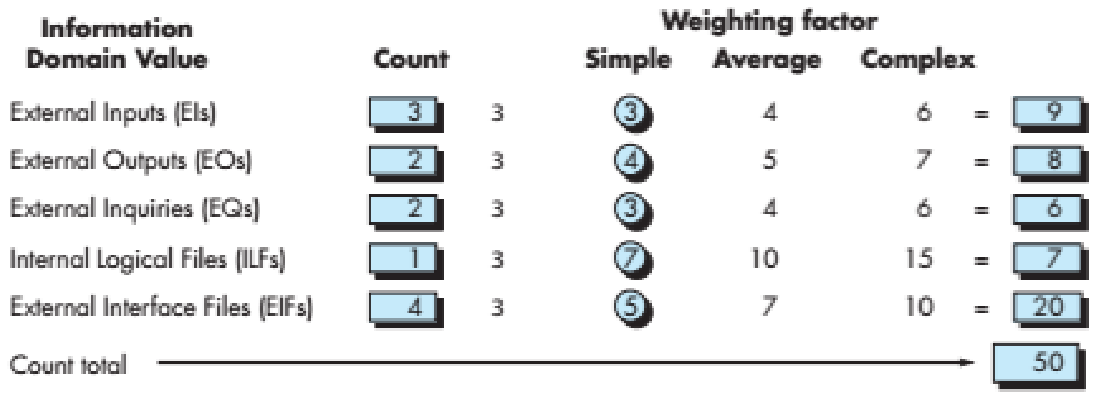
$$FP = \text{count total} \times [0.65 + 0.01 \times \Sigma(F_i)]$$

1. Does the system require reliable backup and recovery?
2. Are specialized data communications required to transfer information to or from the application?
3. Are there distributed processing functions?
4. Is performance critical?
5. Will the system run in an existing, heavily utilized operational environment?
6. Does the system require online data entry?
7. Does the online data entry require the input transaction to be built over multiple screens or operations?
8. Are the ILFs updated online?
9. Are the inputs, outputs, files, or inquiries complex?
10. Is the internal processing complex?
11. Is the code designed to be reusable?
12. Are conversion and installation included in the design?
13. Is the system designed for multiple installations in different organizations?
14. Is the application designed to facilitate change and ease of use by the user?

External Inputs	*Parameters*
External Outputs	*Return Types, Error Messages, UI*
External Inquiries	*Function Calls By Others*
Internal Logical Files	*Local Data Dependent On Global Variables / Files*
External Interface Files	*Global States Used By Function*

Metrics For Requirements: Example





Information Domain Value	Count		Weighting factor Simple	Average	Complex		
External Inputs (EIs)	3	3	[✓ 3]	4	6	=	9
External Outputs (EOs)	2	3	[✓ 4]	5	7	=	8
External Inquiries (EQs)	2	3	[✓ 3]	4	6	=	6
Internal Logical Files (ILFs)	1	3	[✓ 7]	10	15	=	7
External Interface Files (EIFs)	4	3	[✓ 5]	7	10	=	20
Count total						→	50

$$FP = 50 \times [0.65 + (0.01 \times 46)] = 56$$

Metrics For Code

n_1 = number of distinct operators that appear in a program
n_2 = number of distinct operands that appear in a program
N_1 = total number of operator occurrences
N_2 = total number of operand occurrences

$$N = n_1 \log_2 n_1 + n_2 \log_2 n_2$$

N = Length

$$V = N \log_2 (n_1 + n_2)$$

V = Volume

$$L = \frac{2}{n_1} \times \frac{n_2}{N_2}$$

L = Ratio of Ideal to Actual Volume

Metrics For Testing

n_1 = number of distinct operators that appear in a program

n_2 = number of distinct operands that appear in a program

N_1 = total number of operator occurrences

N_2 = total number of operand occurrences

$$PL = \frac{1}{[(n_1/2) \times (N_2/n_2)]}$$

$$e = \frac{V}{PL}$$

Percentage of testing effort $(k) = \frac{e(k)}{\Sigma e(i)}$

PL = Program Level

e = Testing Effort

k = individual module

Metrics For Maintenance

M_T = number of modules in the current release
F_c = number of modules in the current release that have been changed
F_a = number of modules in the current release that have been added
F_d = number of modules from the preceding release that were deleted in the current release

$$\text{SMI} = \frac{[M_T - (F_a + F_c + F_d)]}{M_T}$$

SMI = Software Maturity Index