The Making of the Complexity Score

ORACLE®

Introduction

The calculation of a complexity score for both hardware and software can be represented as:

$$C_n = \sum (F^3 + D^3)_n$$

where C is complexity, F is functionality, D represents the dependencies between functional units and n is the total number of systems being measured.
The beauty of this algorithm lies in the application to disparate entities - hardware entities, software entities, human resources and others. A derivation follows.

Derivation

Let C represent complexity and F represent functionality or number of functional units (note these can be pieces of hardware or software modules)

The governing assumption drives our model and may be expressed as: when we increase the number of functional units by 25% then our complexity doubles. This may be expressed as a power law:

$$C = F^x$$

Let's find x. From our governing assumption

$$2 * C = (1.25*F)^x$$

If we take the logarithms of both equations:

$$\text{Log } C = x * \text{Log } F$$

and

$$\text{Log } 2 + \text{Log } C = x * (\text{Log } 1.25 + \text{Log } F)$$

If we subtract the equations we have

$$\text{Log } 2 = x * (\text{Log } 1.25 + \text{Log } F) - x * \text{Log } F$$

or

$$\text{Log } 2 = x * \text{Log } 1.25$$

Divide each side by Log 1.25 to solve for x and we have:

$$x = \text{Log } 2/ \text{Log } 1.25$$

or x is approximately equal to 3.

Therefore complexity can be determined by the cubic of the number of funtional units or:

$C = F^3$

Using the same reasoning for dependencies, D, we derive the complexity score for one system:

$C = F^3 + D^3$

The addition operation is a conservative derivation as dependencies and complexity are multiplicative. As this is a relative score for before and after situations an addition operation will suffice.

However this calculation applies to one system; for multiple systems and disregarding the dependencies between systems, we re-formulate to:

$C = \sum (F^3 + D^3)_n$

Therefore to determine the **total** complexity score, calculate the complexity score for each system and sum them for n = 1 to n = total number of systems.

Conclusion

Here we showed the derivation of the calculation of a complexity score for both hardware and software, represented as:

$C_n = \sum (F^3 + D^3)_n$

We described the underlying assumptions and explained the inherent beauty of the algorithm.

We are mindful of entropy. Entropy is the measure of the uncertainty associated with a random variable and can be found in statistical thermodynamics, quantum physics as well as information theory. Here the variables are functional units and dependencies. Although we are cognizant of entropy that enters a system, we preferred to provide a conservative, believeable model all the while aware of the complexity barbarians just beneath the surface of our analysis.

You can't communicate complexity,
only an awareness of it.

Anonymous

Title: The Making of the Complexity Score
Authors: Peter Doolan and William Indest
Managers: Todd Lesser, Brian Nault

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com