Running Head: Weighting Matrix

Week 6 Application: Technical Requirements Weighting Matrix

Jered McClure

Walden University

Week 6 Application: Technical Requirements Weighting Matrix

The hardest decision within these weighting requirements was determining the raw values for the in-house build. Being that I do not know anything about the programmers, I cannot give an accurate assessment of their skills. As such, I went off of a rounding solution. Essentially, I took the raw scores for the other two systems, added them together, and divided them by two. This number was then rounded down to the nearest integer.

This raw score was placed in for the programming errors and quality of code categories, but a grain of salt must be taken when viewing this score as it assumes that the programmers in house are experienced and competent within their skillset. Note that the in-house programmers would truly need to be sub-par for them to have an effect on the score for Alternative 1. Being that the highest raw score can be 6, middle is still sitting at 3 or 4. This means the programmers would need to get a quality of code and programming errors score of 1 each to drop the in-house build below the acceptable threshold.

 The weights I gave to each category were based around a 1 to 5 score, where 1 is the lowest and 5 is the highest. The only fields to receive a 5 were Robustness, Documentation, and User-friendliness. These three fields epitomize the needs of any system development project, in my opinion.

The lowest weight, of 2, was given to easy installation. While this may or may not be a pain for the technicians to manage, installation is a onetime action. After the software is installed, this will no longer be a current issue. Therefore, it should not weigh heavily on the overall scheme when looking at a new system. Programming errors, quality of code, and flexibility were all placed firmly in the center with 3s. This is due to their overall need within any system, but the fact that their requirements will affect a smaller proportion of the user target base.

With all this in mind, the extended scores produced a verdict which put the in-house build firmly in the lead. However, this lead is subject to the capabilities of the programming team. With that in mind, the turnkey solution is the obvious choice at only 13 points behind the in-house build. For a full decision between the two, a thorough interview and analysis of the in-house skillset would need to take place. If their skills are truly up to par, then there is no reason not to stay in-house for the system solution.

IMS Technical Requirements Weighting Matrix (Average In-House)



IMS Technical Requirements Weighting Matrix (Sub-Par In-House)



Reference

Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2009). *Systems Analysis and Design in a Changing World* (5th ed.). Boston: Cengage Learning.