INTRODUCTION

Any well-established software undergoes revisions and overall development. These changes are usually dictated by some sort of change procedure, and many times by a formal change-request system. PHP change requests are made through the request for comments (RFC) process.

OVERVIEW OF RFC PROCESS

Those wishing for new features or changes in PHP must have an account at the Wiki PHP website, and must first send an email of their desired proposal. If negative feedback is not received, then the proponent of this change may proceed by creating an RFC at the Wiki PHP website, making it available for discussion by others. The requestor is encouraged to reply and resolve all issues others may have with their proposal. The RFC should then be revised to document all of the concerns brought up. When discussions have ended regarding the RFC, and at least two weeks have passed, the RFC can then receive “voting” status. Those that have already contributed to the development of PHP do the voting. If the change affects the language, then 2/3 of the votes are required, while others types require only a simple majority (Suraski, Joye, and Parra, 2011).

PROPOSED ADDITIONS

The first PHP RFC that I will be analyzing is the “Normalize increment and decrement operators”. The author of this RFC has made the claim that increment “++” and decrement “—“ operators are “not very intuitive” (Meesters, 2013). Currently, Boolean variables incremented or decremented are changed to their opposite values. A null value decremented stays null, while one that is incremented has a value of “int(1)”. Incremented empty strings change to “1”. The RFC’s author proposes either of the following:

* Boolean and null types should be restricted from incrementing and decrementing.
* Boolean and null types should be treated as integer values.

To me, this RFC is not very clear. In option 2, examples are given of incrementing or decrementing the same variable, but resulting in different values. I would advise the author to make this clearer.

The second PHP RFC that I will be analyzing is the “Loop+Else control structure”. The author of this RFC proposes that “else” be an optional block after a loop (Ravazin, 2012). If the loop is never executed, then the else block will execute. Ravazin (2012), states that it is very common that actions are required if a loop is not executed, and this implementation of this proposal would result in shorter and clearer code. If I were able to, I would vote “No” because I find it somewhat unintuitive, having always been accustomed to “if-else” statements. Also, from my understanding, the construct the author proposes exists in Python, however it has the opposite behavior. This will undoubtedly lead to much confusion to seasoned Python programmers.

COMPARISON CHANGE PROCEDURE FOR PUTTY

The change procedure for the open-source Telnet/SSH Client software PuTTY is much simpler than that of PHP. First of all, the team in charge of PuTTY admits to receiving an excessive amount of email, so they first recommend checking their wishlist page to see if the change being requested has already been made. If it is not, then the requestor is then instructed to send an “informative” report, consisting of software version number, detailed descriptions, and screenshots (PuTTY, 2014). While this method of requesting changes may seem more straightforward than that of PHP, it seems less coordinated, and there is not much of a sense of community, if any at all.

REFERENCES:

Meesters, T. (2013) ‘Php Rfc: Normalize Increment And Decrement Operator’ *PHP.net Wiki* [Online]. Available from: <https://wiki.php.net/rfc/normalize_inc_dec> (Accessed: 21 June 2014)

PuTTY (2014a) PuTTY Feedback and Bug Reporting [Online]. Available from: <http://www.chiark.greenend.org.uk/~sgtatham/putty/feedback.html> (Accessed: 21 June 2014)

Ravazin, D. (2012) ‘Request for Comments: Loop+Else control structure’, *PHP.net Wiki* [Online]. Available from: <https://wiki.php.net/rfc/loop_else> (Accessed: 21 June 2014)

Suraski, Z., Joye, P., & Parra, D. (2011) ‘Request for Comments: Voting on PHP features’, *PHP.net Wiki* [Online]. Available from: <https://wiki.php.net/rfc/voting> (Accessed: 21 June 2014)