

Bulldog Software Engineering
468 Broad Street
(318)-626-8591

To: L. P. Broadstreet, Senior Software Engineer
From: Dillon Welch, Junior Software Engineer
Subject: Details on the upcoming IEEE (Institute of Electrical and Electronics Engineers)
Computer Society

Date: 3/29/2010

While doing research for our presentation at the next IEEE Computer Society meeting, I found a series of intriguing articles written by the Editor-in-Chief of the IEEE Software journal on the development and implementation of modern software engineering ideas and how to ensure they succeed. The nurturing of ideas in software engineering has a definable process and there are processes called levers that can prevent an idea from regressing along the process (Erdogmus 4). As ensuring that ideas become software is vital to the success of our company and the industry in general, I believe that we should use the upcoming meeting to discuss these ideas and how to implement them in our work.

The Life Cycle of an idea

An idea goes through ten distinct stages: conception, flatlining, revisiting, bundling, branding, hype, testing, mellowing, extraction, and steaming. An idea is conceived (conception), becomes dormant for any of a variety of reasons (flatlining), eventually is revisited (revisiting) and is placed in the context where it becomes viable to the target audience (bundling), is described in a way that is appealing to others (branding), becomes popular (hype), becomes adopted by early users (testing), its strengths and flaws becomes known and tweaked (mellowing), the modified and improved idea gets taken out (extraction), and is accepted by the target audience (steaming) (Erdogmus 4-5).

Preventing Reversion

This life cycle is filled with regressive loops that can cause an idea to go backwards on the development process. There are five levers that can help prevent this recursion, and each one has a corresponding place in the life cycle. The first lever is unbiased reflection, and this occurs at the revisiting stage. This involves looking at the idea's pedigree and antipedigree, or the previous versions of the idea and previous competing ideas. The second lever is neat bundling, and this occurs at the bundling stage. This means to make sure that it is more cost effective to adopt the new idea over the current approach, as the new idea is neatly put together and more efficient. The third lever is effective branding, and this occurs at the branding stage. This involves ensuring that the idea will be adopted by the target audience using appropriate vocabulary and fact-based support. The fourth lever is easy sandboxing, and this occurs at the testing stage. This involves ensuring making sure the idea is testable and understandable to the point where testers will want to adopt it. The fifth lever is optimal extraction, and this occurs at

the extraction stage. Deciding on optimal extraction time is based on feedback from testers, timing, and the size of the extraction (Erdogmus 4-7).

The understanding of the life cycle of a software engineering idea and the preventing its recursion is essential to the success of the industry, and therefore I feel that we should use this as the topic at the upcoming IEEE Computer Society meeting. Thank you for considering this new idea. If this topic leads to a successful meeting, perhaps it will inspire someone else to think up even better ways to enhance the process of software engineering.

Works Cited

Erdogmus, Hakan. "Déjà Vu: The Life of Software Engineering Ideas." *IEEE Software* 27.1 (2010): 2-5. 31 Dec. 2009. Web. 28 Mar. 2010.

Erdogmus, Hakan. "Regress or Progress? Seeing Good Software Engineering Ideas Through." *IEEE Software* 27.2 (2010): 4-7. 25 Feb. 2010. Web. 28 Mar. 2010.