

Build Notifications in Agile Environments

Ruth Ablett, Frank Maurer, Ehud Sharlin, Jörg Denzinger, Craig Schock

1 Introduction

Agile methods [1] are becoming popular in the software industry. In agile software development projects, it is imperative that all software written by each developer integrates properly into the entire project. To this end, most agile teams adopt *Continuous Integration (CI)*. CI is the practice of automatically compiling, deploying and testing the entire codebase against a suite of prewritten tests. This occurs after any change to the codebase, usually multiple times per day.

When integration is finished, it is important for the developers to become aware of the result so that any problems can be immediately fixed. Undetected bugs can cause further problems as other developers may synchronize with a broken version of the codebase, and this may result in increased effort required to fix the problem and delays in integrating their changes to the latest build. Thus, awareness of the build status is essential, especially immediately after submitting new code to the codebase.

2 Previous Work

A study by Saff and Ernst [2] evaluated continuous integration when used by a single developer to ensure new code passed regression and unit tests. They found that continuous integration had a positive effect on the completion of programming tasks. Our research focuses on how agile teams can be notified when build breakages occur.

Alberto Savoia [3] created a build notification system using peripheral awareness [4] to inform developers of the build state. The system uses two lava lamps, one red and one green. The continuous integration tool turns on the green lamp when the build was successful, and the red when it was broken.

3 Experimental Setup

The goal of the experiment described herein is to evaluate three notification mechanisms - e-mail (virtual), lava lamps (ambient), and BuildBot (active) - within the context of a shared project. Notifications were sent out when code is committed. The goal was to determine which of the three modes would be most effective.

Java Lava Lamps were used as an ambient device in this study. An ambient display is a way of keeping people informed about the build state without disrupting them.

The BuildBot [5] robotic notification device was designed as an active, ambient build notification tool to study the effect of such a device on an agile team in the

context of a shared project. If the build fails, BuildBot follows a network of lines to reach the responsible developer's workstation and kindly barks until the build is fixed.

The three-week experiment was set up as follows: During the first week, email was sent only to the developer responsible for a build breakage. During the second week, a pair of *Java Lava Lamps* were installed and showed the build status. BuildBot was used as the physical notification device instead of the lava lamps for the third week.

4 Results and Discussion

Email was preferred by some developers because each message is nearly instantaneous, simple, not location dependent, not obtrusive to others, and shows the entire build break message such as tests failed. However, too many can become spam.

Some participants liked the lava lamps because they were simple, unobtrusive, and fun. However, most participants did not notice the lamps because of the cubicle walls, and thought the information was too limited on its own (only red or green). And finally, a developer must be present in the same room to see the lava lamps.

BuildBot's popularity among some developers and observers may be due to the fun and the novelty factor. Others, however, did not like the fact that the robot notifies everyone. Also, some expressed concern of the robot's singling out of one developer.

5 Conclusion and Future Work

The results of this evaluation show that the social nature of the group must be considered when introducing any continuous integration notification device.

The Java Lava Lamps used in this study were well-received in that they were fun and unobtrusive, but we believe they would be better in a more open environment.

Introducing something as potentially disruptive as BuildBot can cause friction. Since we found that to the developers, email was the most popular, followed by the lava lamps, we conclude that the most effective for an agile development team would be a combination of an openly visible but unobtrusive ambient and a virtual one.

The results presented here are those of a small-scale, short-term study. A longer-term evaluation (months or years) is needed involving many more developers. Also, there are also many kinds of alert mechanisms that have yet to be evaluated, such as ceiling-mounted rope lighting, system tray alerts, or a visit from a project manager.

References

1. *Manifesto for Agile Software Development*, 2005. Accessed 5 June 2007. <http://agilemanifesto.org>
2. Saff, D., and Ernst, M.D.: An Experimental Evaluation of Continuous Testing During Development. In *International Symposium on Software Testing and Analysis (ISSTA '04)*.

3. Savoia, A.: On Java Lava Lamps and other eXtreme Feedback Devices. 26 August, 2004.
<http://www.artima.com/weblogs/viewpost.jsp?thread=67492>
4. Cadiz, J.J., Venolia, G.D., Jancke, G., Gupta, A.: Sideshow: Providing peripheral awareness of important information. *Microsoft Research Technical Report MSR-TR-200181*, 2001.
5. Ablett, R., Sharlin, E., Maurer, F., Denzinger, J., and Schock, C.: "BuildBot: A Robotic Self-Supervision Mechanism for Agile Software Engineering Teams", IEEE RO-MAN 2007.