BLACKBOARD OPEN LMS APPROVED PLUGIN PROGRAM

Code Review Process

SEPTEMBER 27, 2018

I. Introduction

In order to provide the most stable and secure infrastructure for its clients, the Blackboard Open LMS Development Team reviews all third-party code before it can be included into the central codebase used by all Blackboard Open LMS hosted instances. Blackboard has designed and developed this Code Review Process to assist client and partner developers in creating custom code for Moodle that will have a high probability of passing Blackboard Open LMS code review for eventual inclusion to the central codebase.

II. When is a code review required?

A code review is required for the upload of any third-party code to the Blackboard Open LMS codebase. This could include:

- Code developed by a Vendor, Client or third-party authorized by the Client. This includes code developed by other Moodle partners;
- Integrations with external third-party tools;
- Plugins found on moodle.org or on related sites.

III. What is the code review process?

1. Requesting party should submit the code to be reviewed by the Blackboard Open LMS Development team.
    a. If you are an existing Blackboard Open LMS client, this request can be made to your Engagement Manager via a support case.
    b. If you are a third-party plugin vendor interested in integrating with Blackboard Open LMS, please contact Kai Lou, VP of Business Development at BbPartnerTeam@blackboard.com.
    c. If you are a prospect, please let your sales engineer or sales director know you are interested in integrating with third-party code.
2. Blackboard will evaluate, at no cost, the level of effort required to review the code and generate from this a Statement of Work. The level of effort – and henceforth cost – will be wholly based on three factors: length, complexity and quality of the code.
3. Upon receiving a signed Statement of Work back, Blackboard will notify you of the timeline for reviewing the code and of the project start. Code reviews take approximately 4-6 weeks to complete.
4. Blackboard will then review the code and determine if it meets Blackboard's review standards and performs properly within our hosted installation.
    a. If the code passes review:
        i. Blackboard will notify the requesting party and make the module available on a sandbox site accessible by the Blackboard Open LMS client upon the next release.

            1. Please refer to the publicly available release schedule - https://community.blackboard.com/groups/blackboardopenlms/pages/welcome
            2. Plugin vendors are not provided with a Blackboard Open LMS site, but can work with mutual customers to test, or license a site as desired.
        ii. The code will then be made available on the Blackboard Open LMS production build at the time of the next release.
        iii. In this scenario, installation of the code on Blackboard Open LMS staging site constitutes project release.
    b. If the code fails review:
        i. Blackboard will supply the requesting party with a code review summary outlining identified issues. In this scenario, provision of the summary constitutes completion of the project. The summary will outline all major and minor code violations; major violations must be fixed before the plugin can be added to the Client's codebase.
    c. The requesting party may make those changes on its own and/or sub-contract with an outside vendor to make them (Blackboard or otherwise) and re-submit the plugin for code review by Blackboard. If the requesting party wishes for Blackboard to make the fixes, this is handled through a separate Statement of Work.

5. The initial code review Statement of Work will allow the requesting party to re-submit the plugin up to two times annually for re-review, so long as no more than ten percent of the code has changed.
    a. These re-reviews can be used for the review of code fixed after a failed review and/or updates to the plugin.
    b. Additional reviews beyond the two included, or reviews that involve more than ten percent of the code, will require a change order and additional fees.

IV. Why does the code require a review?

Blackboard Open LMS uses a single codebase for all of our clients, which enables us to push out security patches and upgrades almost simultaneously with their release by moodle.org (and of course, by Blackboard). Because all clients are using the same codebase, we have to be extremely careful about only pushing out code that is secure and tested. We cannot make client X happy by introducing risk to client Y.

The variance in the quality of Moodle plugin codes available in the community is drastic. Some are extremely well written; others involve major hacks and/or introduce potential real security risk. It is Blackboard’s responsibility to protect our clients, to ensure that the third-party code will work within the agreed upon SLA and perform within our stated uptime. Blackboard applies these requirements to all clients to prevent any client from developing code that degrades server performance or uses a high amount of resources. Blackboard’s unique server architecture also demands that certain coding practices are followed in order to work properly in the hosted environment. Blackboard inspects all code for security issues and performance issues. The cost of this process is not intended to be prohibitive, only to cover the cost and time required to review the Client's code for performance and security risks to the hosted instance.

V. I paid for a Code Review, and I even paid for Blackboard to fix the major findings from the review, but I still see bugs in the plugin. Why?

The code review process does not ensure that the code being reviewed will be bug-free. We do not run the plugin through functional Quality Assurance (QA); we do not even profess to understand the inner workings of its functionality. A code review is intended solely to ensure that the plugin does not introduce risk to our codebase. Blackboard does not endorse third-party code that is uploaded to the site; it merely is satisfied that the code will not introduce security or performance risks to our clientele. If the Client wishes to address bugs found in the plugin once it is uploaded to their site, Blackboard recommends one of the following options:

- Contact the original third-party developer to have him/her fix the code. Note that in this case, if the code has already been modified by Blackboard to meet our review guidelines, the revised code should be provided to the developer for fixes to be done on that.
- If the above is not possible, contract with Blackboard via a subsequent SOW to fix specific, named bugs in the plugin.

VI. How can I best ensure that code will pass the review?

Before submitting code, please request from the Blackboard Open LMS team our detailed Code Review Guidelines. We encourage you to check that the code meets these guidelines before submitting it to us for review. In addition to ensuring that code does not introduce any risk to the codebase, the other most important issue for clients to be aware of is that THE CODE MUST BE MODULAR. Code that involves changes to Core Moodle or Blackboard Open LMS files will not be approved. All code must be encased in a plugin (e.g. auth, enrollment, block, activity) that can be turned on/off for clients and that is separate from the Core code.

VII. I want to develop my own code. How does that work?

Blackboard is dedicated to providing tools and features that meet the unique needs of its clients. To ensure Blackboard Open LMS remains relevant to the evolving needs of its customers, Blackboard has developed a Client Contributor Program to allow clients to contribute to the development of Blackboard Open LMS.

Client contributors can develop custom features based on the Core Moodle code that is available for free and can be downloaded at www.moodle.org. They can then submit the code to Blackboard for a code review. If approved and accepted by Blackboard, client contributions may be incorporated into the standard codebase.