

Development Practices

Information

This contains information which details the development practices used for writing code for the evaluation system.

JIRA practices

1. Make sure you review the outstanding issues at least once a day to see if there is anything new or anything you can take care of.
2. Use the **In Progress** when you are working on an issue and assign it to yourself.
3. For new issues, leave them unassigned, assign them to yourself, or assign them to the primary for the area (the primary can be determined by looking at the first name in the author list and excluding me if it is in the presentation layer).
4. Check the issues you reported that were resolved by someone else and close them as soon as possible if you can verify they are resolved (always verify it first).

SVN practices

1. All SVN commits should compile and allow tomcat to startup without failure
 - a. If you **MUST** commit code that cannot compile please make sure you write this clearly in the commit comment and clear this commit with the project lead first
2. Always put a comment when committing to SVN. Most everyone is good about this however there were a couple commits recently with no comment. You know who you are so please include a comment on EVERY commit.
3. If your commit is to address a JIRA issue (which most should be) then please put the JIRA key (e.g. EVALSYS-3) in as the first word in the commit comment
 - Example: EVALSYS-6 corrected stale hibernate object exception

Code comments and authors

1. All java files must have the ECL license included with them.
This is an example code comment template from eclipse:

```
/*
 * *****
 * ${file_name} - created by aaronz@vt.edu
 *
 * Copyright (c) ${year} Virginia Polytechnic Institute and State University
 * Licensed under the Educational Community License version 1.0
 *
 * A copy of the Educational Community License has been included in this
 * distribution and is available at: http://www.opensource.org/licenses/ecl1.php
 *
 * Contributors:
 * Aaron Zeckoski (aaronz@vt.edu) - primary
 *
 * *****
 */
```

- You will want to replace the University name, the email address, and the contributors list
2. Javadoc comments on classes are required for all classes and methods. Variable comments are optional but should be included in most cases. Primary authors on classes are responsible for comments on code contained within.
 3. The standard for javadocs comments is like this:

```
/**
 * comment stuff
 */
```

- Insert a boilerplate javadoc comment above any method or class by putting the cursor on the name and hitting Alt-Shift-J
- These are not javadoc comments:
 - a. NOT javadoc comment

```
/*
 * comment
 */
```

b. NOT javadoc comment

```
// comment
```

- Please make sure any class, method, or variable comments are actual javadoc comments and not some other kind of comment.
4. **Authors:** The author line (e.g. @author) should only include one name. Add extra authors by adding multiple @author comments in the javadoc comment.
- If you add yourself as an author on a file I expect that means you have written at LEAST 20% of the code in that file or you were the primary author of the file. If you edit one method and then add yourself as an author for a file it is likely I will remove your name from the author list. The idea of the author list is that anyone in it should be able to understand and correct any issue in that file. If you are listed as an author on a file and there is a related error I will assign you to fix the issue with it so make sure you understand the entire file before you make yourself an author.