Nadiger, Sushruta

Sushruta.Nadiger@ellucian.com

Abstract

This document lists important tips and tricks for refactoring existing codebases

Refactoring

Tips and Tricks

Refactoring strategy

* Create a plan on what needs to be refactored
* Observe the existing code and understand what needs to be changed
* Analyze the dependencies
* Create a baseline code
* Refactor the code in small and controlled steps so that it is easy to revert to the baseline if things go wrong
* Perform the refactoring
* Test the codebase

Refactoring cycle

Identifying the problem areas

Code Smells: Any symptom in the source code that possibly indicates a deeper problem.

Some examples of Code Smells are:

* + Race conditions, Lockups, Slowdowns and Crashes
  + Experimental code that is still in the codebase
  + Inconsistent naming

Avoiding Code Smells:

* Keep an active log of issues and bugs
* Isolate code you know is not working correctly until you can fix
* Avoid hacks/quick fixes

Common Optimizations

* Loops are bad as they block other code from executing
* Nested loops are even bad
* Simplify Math:

1. Division, Modulus and Sqrt are all expensive

2. Multiply instead of dividing: 4/2 should be 4\*0.5

* Object Pooling:

1. Creating new instances of classes in expensive

2. Avoiding using 'new' keyword as much as possible

* Avoid Reflection
* Take the performance hit up first - Perform the complex calculations when the app is loading and cache the results

Code Readability

* Method names should be consistent and intuitive
* Naming conventions should be maintained consistently
* Classes should implement interfaces and be type to them
* Use clear property names
* Limit parameters for methods
* Document the code/Add comments

Testing

* Rely on your UTs
* Automate as much as possible
* Have a QA run through the automated tests

Visual Studio Tools for refactoring

* Rename – Consistently renames methods or properties
* Extract – Extracts Methods
* Encapsulate – Converts a field into a property
* Extract Interface – Extracts interfaces from a class
* Change Signature – Consistently changes the method signature

Version Control

* Always make sure that the baseline codebase is backed up and is under version control
* Let one person be responsible for merging changes into the baseline.
* Make sure to run the tests before and after merging
* Baseline must always be the stable version of the codebase
* Refactoring should always be performed on separate branches that are directly branched off from the baseline

===================================END OF DOCUMENT=================================