

Lab Practicum #3

Collaborative Software inspection

Background:

An inspection is one of the most common sorts of review practices found in software projects. **The goal** of the inspection is to **identify defects**. In an inspection, a defect is any part of the work product that will keep an inspector from approving it. For example, if the team is inspecting a software requirements specification, each defect will be text in the document which an inspector disagrees with.

Inspection roles:

- **Moderator** (1 student): This is the leader of the inspection. The moderator plans the inspection and coordinates it. In this practicum moderator should split all files among two other students and himself, collect inspection results, and produce inspection report.
- **Inspector** (2 students): The person that examines the work product to identify possible defects. Should inspect given files and share results to moderator, provide valuable suggestions for improving code quality.

Goal of the lab:

Inspect source code (code review) of real-world software application and produce inspection report.

Task description:

- 1) Download and install RSS Bandit application and its source code (see Input Materials section).
- 2) Open source code in Visual Studio 2008 (*RSS Bandit.2008.sln*).
- 3) Review following .cs files according to code guidelines (*Input Materials, 3*):

a. RssBandit/Core/CommandLineParser.cs

- b. RssBandit/Core/GuiInvoker.cs
- c. RssBandit/Core/RssBanditApplication.Commands.cs
- d. RssBandit/Core/RssBanditApplication.cs
- e. RssBandit/Core/RssBanditApplication.EventHandlers.cs
- f. RssBandit/Core/ RssBanditUpdateManager.cs

4) Additional tasks:

- a. Try to find duplicate code, propose refactoring;
- b. Try to find code blocks which should use patterns;

Outcome:

- 1) Collective inspection report prepared by Moderator,
- 2) Individual reports prepared by inspectors.

RSS Bandit inspection report

11/09/2009

Inspectors:

John Doe

Inspected Files:

RssBandit/Core/GuiInvoker.cs

RssBandit/Core/RssBanditApplication.EventHandlers.cs

Code conventions rules:

1.9 Naming conventions and rules,

...

Summary:

File / Line	Code	Recommendation
GuiInvoker.cs Line 61	<code>var d = (InvokeControl) state;</code>	Rule (9.a), Avoid single character variable names; Avoid usage of var keyword. Might be replaced with: <i>InvokeControl control = (InvokeControl)state;</i>
RssBanditApplication .Commands.cs Line 448	<code>if (body.Length + subject.Length > 900)</code>	Avoid explicit constants: <code>const int maxLength = 900</code> <code>if (body.Length + subject.Length > maxLength)</code>

Suggestions:

CommandLineParser.cs file contains 8 different classes. According to guidelines each file should contain one class with the same name as a name of the file.

Input Materials:

- 1) RSS Bandit application <http://rssbandit.org/>
 - a. <http://sourceforge.net/projects/rssbandit/files/rssbandit/v1.8.0.870/RssBandit.1.8.0.862.Src.zip/download>
 - b. http://sourceforge.net/projects/rssbandit/files/rssbandit/v1.8.0.870/RssBandit1.8.0.870_installer.zip/download

- 2) http://en.wikipedia.org/wiki/Software_inspection

- 3) **IDesign C# Coding guidelines**
<http://www.idesign.net/idesign/download/IDesign%20CSharp%20Coding%20Standard.zip>

- 4) [http://en.wikipedia.org/wiki/Design_pattern_\(computer_science\)](http://en.wikipedia.org/wiki/Design_pattern_(computer_science))

- 5) [Gamma, Erich](#); [Richard Helm](#), [Ralph Johnson](#), and [John Vlissides](#) (1995). *[Design Patterns: Elements of Reusable Object-Oriented Software](#)*. Addison-Wesley. ISBN 0-201-63361-2.