

Practical Refactoring: Making Bad Code Good

Presenter

Dan Wilson

Overall Goal

The goal of this session is to showcase several sections of code that technically function appropriately but leave plenty of room for better readability and extensibility. The presenter will walk the audience through a graduated set of steps to improve the code while maintaining technical accuracy.

Timeline:

5 Minutes	Presenter Bio/Opening Summary Explain the life cycle of an application Explain line potency and internal context Explain readability and the telephone game
15 Minutes	Define/Demonstrate Error Handling Example Setup: Have audience read code and shout out what the purpose of the code is Step 1: <ol style="list-style-type: none">1. sweep the code to understand intent2. remove obstacles to understanding3. make short term goals (stay focused) Step 1 Changes: Compare, mixed tag case, nested conditionals, add comments Step 2: <ol style="list-style-type: none">1. Sweep the code for intent and accuracy2. Read code aloud in head for smoothness Step 2 Changes: Negative boolean logic, duplicated code, more comments Step 3: <ol style="list-style-type: none">1. Review final product.2. Review final product without comments and whitespace for full impact.
15 Minutes	Define/Demonstrate Challenge Question Example Setup: Have audience read code and shout out what the purpose of the code is Step 1: <ol style="list-style-type: none">1. Analyze code2. Identify implementation groups3. Identify intent and add comments Step 1 Changes: Added Comments, refactored Login and Validation subprocesses

	<p>Step 2:</p> <ol style="list-style-type: none"> 1. Look for unnecessary branches or variables 2. Extract Methods where appropriate 3. Clean up extra variables <p>Step 2 Changes: HasChallengeQuestionSet, challengeInfoSubmitted, saveChallengeAnswer, local.result removed, extra logic branches removed</p> <p>Step 3:</p> <ol style="list-style-type: none"> 1. Review final product. 2. Review final product without comments and whitespace for full impact.
15 Minutes	<p>Define/Demonstrate Validation Example</p> <p>Setup: Have audience read code and shout out what the purpose of the code is</p> <p>Step 1:</p> <ol style="list-style-type: none"> 1. Find commonalities 2. Notate differences in comments <p>Step 1 Changes: Added comments to identify the particular differences in common implementations</p> <p>Step 2:</p> <ol style="list-style-type: none"> 1. Wrap up common implementations 2. Migrate to implementations 3. Punt on the hard stuff <p>Step 2 Changes: Most validation now runs through a common hook. Some validation was custom.</p> <p>Step 3:</p> <ol style="list-style-type: none"> 1. Review final product. 2. Review final product without comments and whitespace for full impact.
10 Minutes	Overflow/ Question&Answer