

Fundamentals: The Don't Repeat Yourself Principle Part 3

Steve Smith

<http://pluralsight.com/>



Outline

Part 1

- **DRY Defined**
- **Demo: Repetition in Code**
- **Analysis**
- **Demos: Refactoring to apply DRY**

Part 2

- Refactoring for greater testability
 - Adding Tests
- **Code Generation**
- **Repetition in Process**
- **Demo: Automation to apply DRY**
- **Summary and Related Fundamentals**

Analysis

Part 1

- ~~Magic Strings/Values~~
- ~~Duplicate logic in several methods~~
- ~~Repeated if-then logic~~
- ~~Conditionals instead of polymorphism~~
- Repeated Execution Patterns
- Lots of duplicate, probably copy-pasted, code

Part 2

- ~~Only manual tests~~
- ~~Static methods everywhere~~

Repeated Execution Patterns

```
17 Console.WriteLine("Beginning extract at " + DateTime.Now);
18 Extract();
19 Console.WriteLine("Finished extract at " + DateTime.Now);
20
21 Console.WriteLine("Beginning transform at " + DateTime.Now);
22 Transform();
23 Console.WriteLine("Finished transform at " + DateTime.Now);
24
25 Console.WriteLine("Beginning load at " + DateTime.Now);
26 Load();
27 Console.WriteLine("Finished load at " + DateTime.Now);
28
```

```
34 using (var myConnection = new SqlConnection())
35 {
36     myConnection.ConnectionString =
37         "Data Source=localhost;Initial Catalog=
```

```
53 }
54 }
55 using (var myConnection = new SqlConnection())
56 {
57     myConnection.ConnectionString =
58         "Data Source=localhost;Initial Catalog=
```

```
71 }
72 }
73 }
```

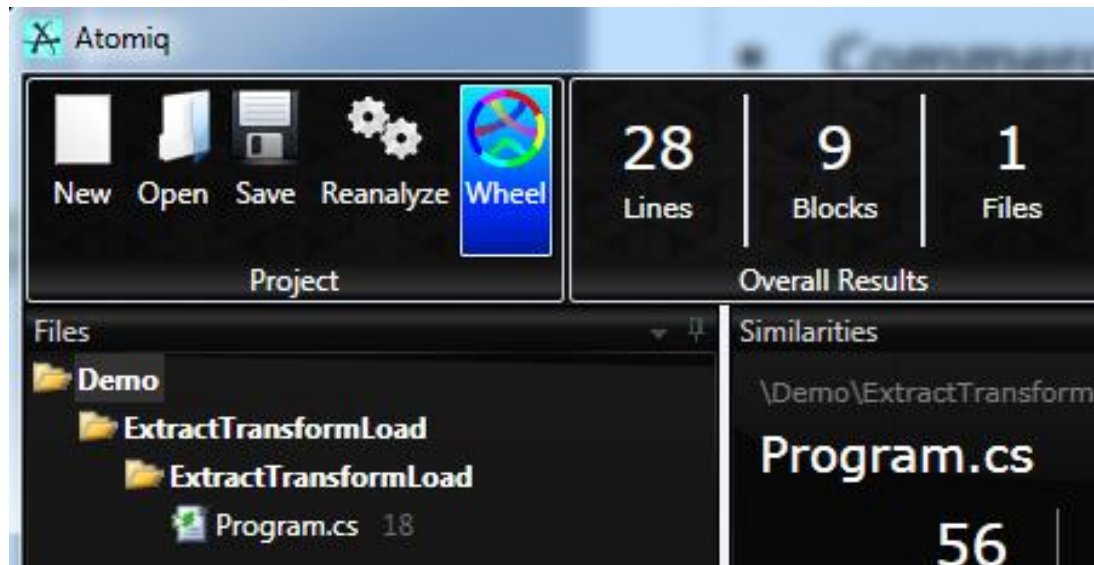
Demo

Refactoring Repeated Execution Patterns



Duplicate, Copy Pasted Code

- Too Numerous to List
- Commercial Tool (free version available)
 - Atomiq (<http://GetAtomiq.com>)
 - *Full disclosure: I'm one of the owners of the company that produces **Atomiq***



Demo

Find Duplicate Code using Tools



Consider Code Generation

- **T4 Templates**
 - Custom code generation templates
 - Built into Visual Studio 2010
- **ORM Tools (LINQ to SQL, Entity Framework, nHibernate, LLBLGen)**
 - Reduce repetitive data access code
 - Eliminate common data access errors
- **Commercial Tools:**
 - CodeSmith
 - CodeBreeze
 - CodeHayStack

Repetition in Process

- **Testing**
 - Performing testing by hand is tedious and wasteful
- **Builds**
 - Performing builds by hand is tedious and wasteful
- **Deployments**
 - Performing deployments by hand is tedious and wasteful
- **Are you seeing a trend here?**

```
String.Format("Performing {0} by hand is tedious and wasteful", action);
```

Demo

Automating Processes



Summary

- Repetition breeds errors and waste
- Use an ORM tool to eliminate repeated data access code
- Use Tools to Locate Repeated Code
- Automate repetitive actions in your processes
- Recommended Reading:
 - The Pragmatic Programmer: From Journeyman to Master <http://amzn.to/b2gJdK>
 - 97 Things Every Programmer Should Know <http://amzn.to/cAse1Y>
- Related Tool
 - Atomiq <http://GetAtomiq.com>

Final Code

**Check out the final version of the demo
in this module's demo zip file**

For more in-depth **online** developer **training** visit



on-demand content from authors you **trust**