On Software Quality

A Discussion

Steve Smith
SteveSmithBlog.com

What is Quality?

"Quality… you know what it is, yet you don't know what it is. But that's self-contradictory. But some things *are* better than others, that is, they have more quality. But when you try to say what the quality is, apart from the things that have it, it all goes *poof*! There's nothing to talk about."

- Zen and the Art of Motorcycle Maintenance

QUALITY

THE RACE FOR QUALITY HAS NO FINISH LINE-
SO TECHNICALLY IT'S MORE LIKE A DEATH MARCH.

Software Quality?

- How well is the software designed?
  - *Quality of Design*
- How well does the software conform to said design?
  - *Quality of Conformance*
- Quality of the software *Product*
  - “fitness of purpose”
- Quality of the software *Code*

External Quality

"External [quality] characteristics are characteristics that a user of the software product is aware of…

External characteristics of quality are the only kind of software characteristics that users care about."

- Steve McConnel, *Code Complete*

Characteristics of External Quality

- Correctness – Does the right thing
- Accuracy – Does the thing right
- Usability – Ease of use
- Efficiency – Resource requirements, scalability
- Reliability – (in)frequency of failures
- Integrity / Security – Prevents invalid use
- Adaptability – Usable in scenarios not designed for
- Robustness – Tolerance for stressful conditions
- Documentation – Presence, accuracy, and usefulness

Characteristics Internal Quality

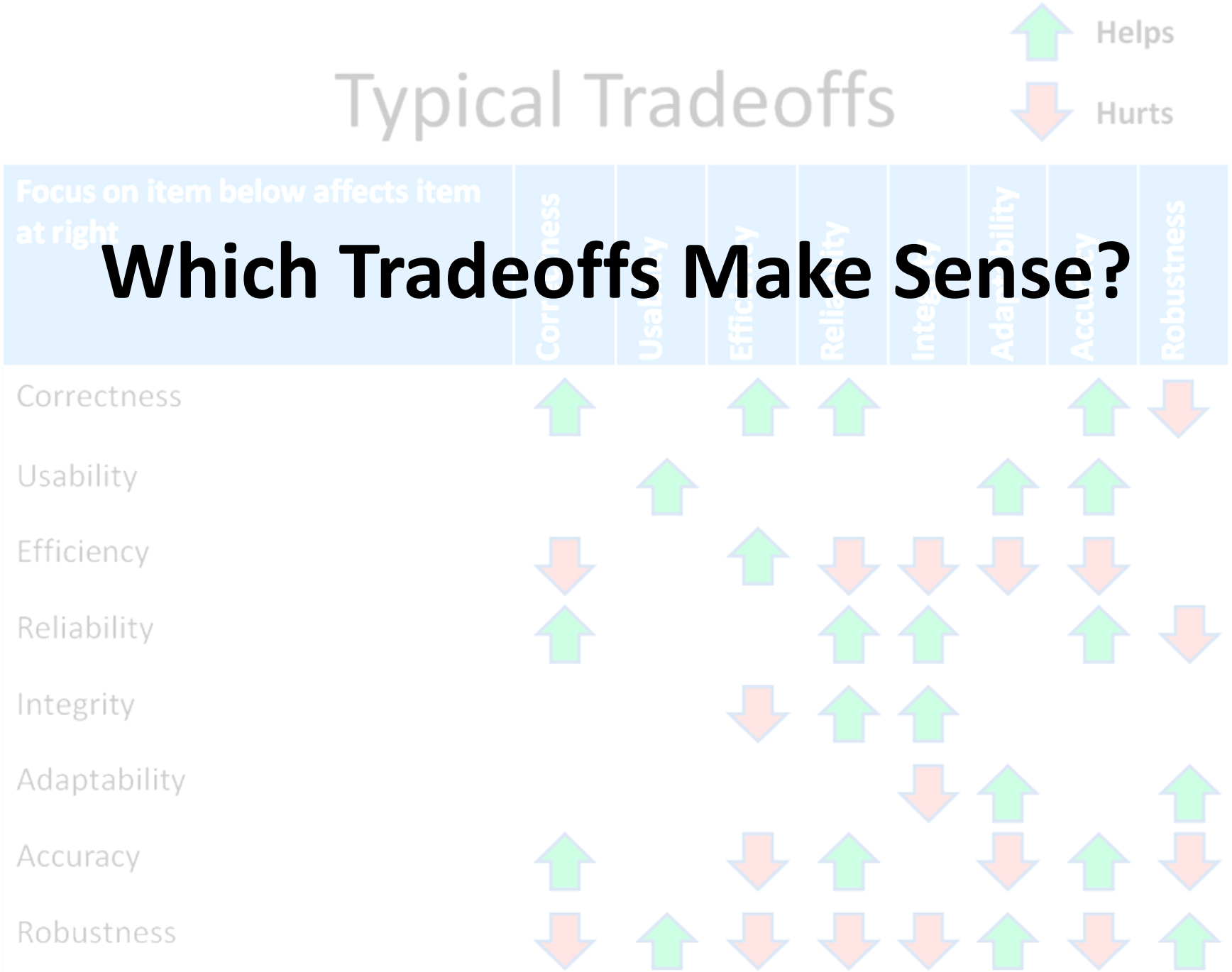
- Maintainability – Ease of modification
- Flexibility – Extent system can be made to work in new environments or uses
- Reusability – Can parts of this system be used in other systems?
- Readability – Ease with which detailed-statements can be read and understood
- Testability – Degree to which the system can be unit- and system-tested
- Understandability – Ease of which entire system can be understood, from organizational to detail level

Typical Tradeoffs

Focus on item below affects item at right	Correctness	Usability	Efficiency	Reliability	Integrity	Adaptability	Accuracy	Robustness
Correctness	↑		↑	↑			↑	↓
Usability		↑				↑	↑	
Efficiency	↓		↑	↓	↓	↓	↓	
Reliability	↑			↑	↑		↑	↓
Integrity			↓	↑	↑			
Adaptability					↓	↑		↑
Accuracy	↑		↓	↑		↓	↑	↓
Robustness	↓	↑	↓	↓	↓	↑	↓	↑

[McConnell, 2004]

Typical Tradeoffs



Which Tradeoffs Make Sense?

The General Principle

- “The General Principle of Software Quality is that improving quality reduces development costs.”

  - Steve McConnell, *Code Complete*

Process

"If your software development process produces defects, then you have a defective process."

Principles of Lean Software Development

- Build Quality In
- Optimize the Whole

Simple Design

1. Passes its tests
2. Minimizes duplication
3. Maximizes clarity
4. Has fewer elements

Rainsberger / Beck

Coupling and Cohesion

- Coupling
  - Refers to how tight the connection is between two classes
  - Looser is better
  - Occurs when abstraction is leaky, or encapsulation is broken
- Cohesion
  - How closely related a classes members are to one another
  - Good abstractions have strong cohesion

Heuristics

- Lines of Code
- Code Coverage (code exercised by tests)
- Cyclomatic Complexity
- NPath Complexity
- Halstead Metrics
- Henderson-Sellers Lack of Cohesion in Methods
- Defect Density (per project, class, method)

Cyclomatic Complexity

$C = E - N + p$

- C = Complexity
- E = Sum(Edges of the graph)
- N = Sum(Nodes in the graph)
- p = Number of connected components

Range (Software Engineering Institute):

Cyclomatic Complexity	Risk
1 to 10	Simple; little risk
11 to 20	Somewhat complex; moderate risk
21 to 50	Very complex; high risk
50+	Untestable; very high risk

Advantages of Cyclomatic Complexity

- Easily computed
- Provides good indicator of ease of code maintenance
- Can help locate
  - Code that requires more tests
  - Complex code for formal review procedures
- Recommendation:
  - **Keep Complexity Under 10 if possible**

Lack of Cohesion of Methods of a Type (Henderson-Sellers)

The intuition underlying the Henderson-Sellers method of calculating Lack of Cohesion of Methods (LCOM) is that in a cohesive class *C*, many methods access the same fields of *C*. Formally, let

- *M* = set of methods in class
- *F* = set of fields in class
- *r*(*f*) = number of methods that access field *f*
- *ar* = mean of *r*(*f*) over *f* in *F*

We then define LCOM of the class under consideration to be

LCOM = (*ar* - |*M*|) / (1 - |*M*|)

We follow Lance Walton (author of the State of Flow Eclipse Metrics Plugin) in restricting *M* to methods that read *some* field in the same class, and *F* to fields that are read by some method in the same class.

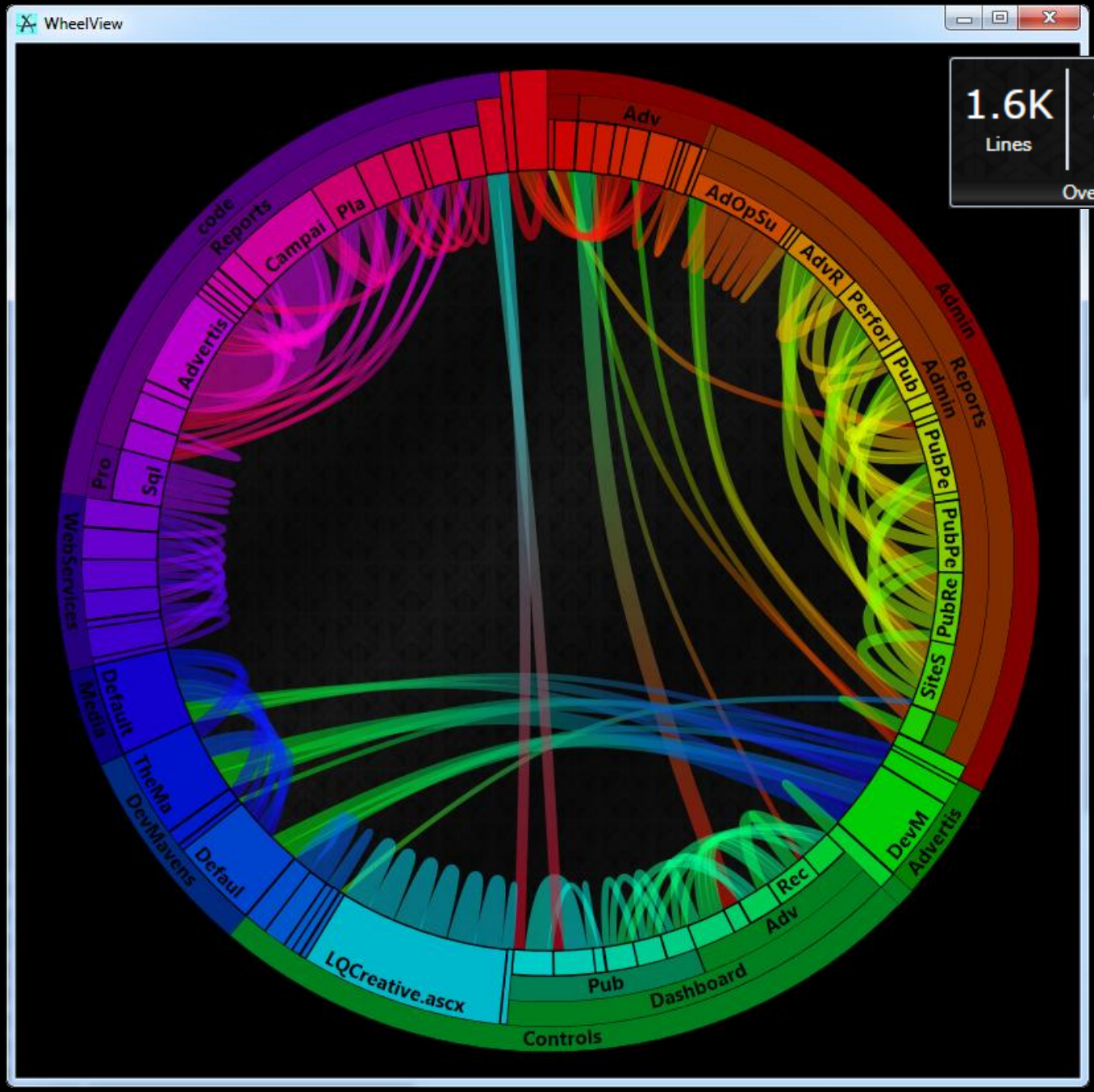
A value greater than 0.9 indicates a class that may deserve some further scrutiny.

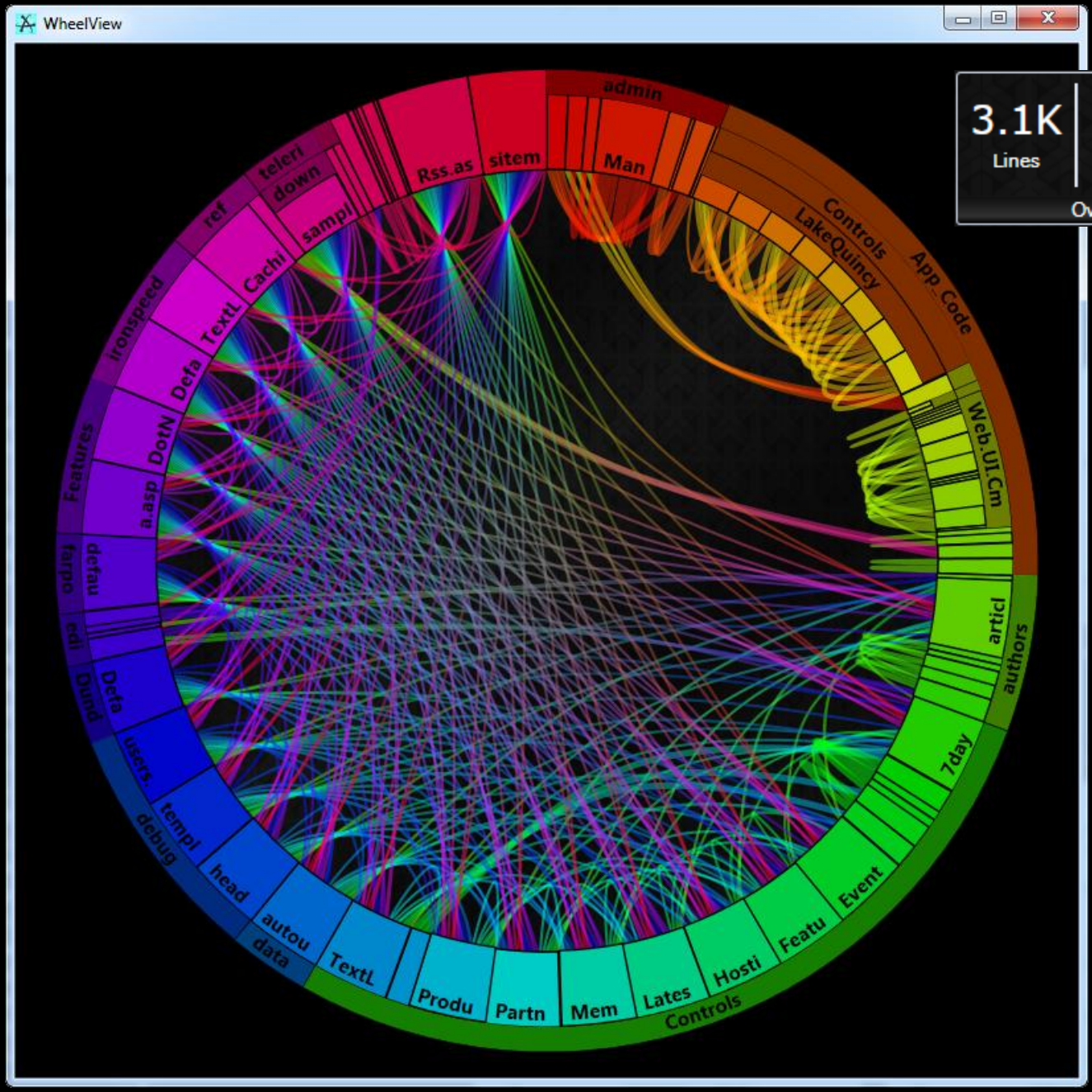
Boy Scout Rule

Leave your code better than you found it.

Tools

Tool	Price	URL
ReSharper	$349	JetBrains.com/Resharper
CodeRush	$249.99	DevExpress.com/CodeRush
Nitriq	$39.95	Nitriq.com
Atomiq	$30.00	GetAtomiq.com
NDepend	$414	NDepend.com


WheelView
1.6K
Lines
143
Blocks
82
Files
Overall Results
Adv
AdOpSu
AdvR
Perfor
Pub
PubPe
PubPe
PubRe
SiteS
Admin
Reports
Admin
DevM
Advertis
Rec
Adv
Dashboard
Pub
Controls
LQCreative.ascx
Defaul
DevMavens
TheMa
Defaul
Media
WebServices
Pro
Sql
Advertis
Campai
Pla
Reports
code


WheelView
3.1K
Lines
378
Blocks
84
Files
Overall Results
admin
Man
sitem
Rss.as
sampl
down
teleri
ref
Cachi
TextL
Deta
IronSpeed
Features
DotN
a.asp
detan
targe
ico
Deta
Dund
users
templ
debug
head
autou
data
TextL
Produ
Partn
Mem
Lates
Hosti
Featu
Event
Controls
7day
authors
articl
Web.UI.Gri
App_Code
Controls
LakeQuincy

Methods to Refactor

```
1  var results =
2  from method in Methods
3  where (method.Cyclomatic > 25 || method.PhysicalLineCount > 200 ||
4  method.TypesUsed.Count > 30 || method.ParameterCount > 7) && method.Type.IsInCoreAssembly
5  select new { method.MethodId, method.FullName, method.Cyclomatic,
6  method.PhysicalLineCount, OutTypes = method.TypesUsed.Count, method.ParameterCount };
7
8  Warn(results, 0);
```

General Stats

Core Assemblies

Assemblies:	8
Namespaces:	112
Types:	2333
Methods:	24291
Fields:	9418
Events:	93
Phys. Line Count:	68303

Used by Core Assemblies

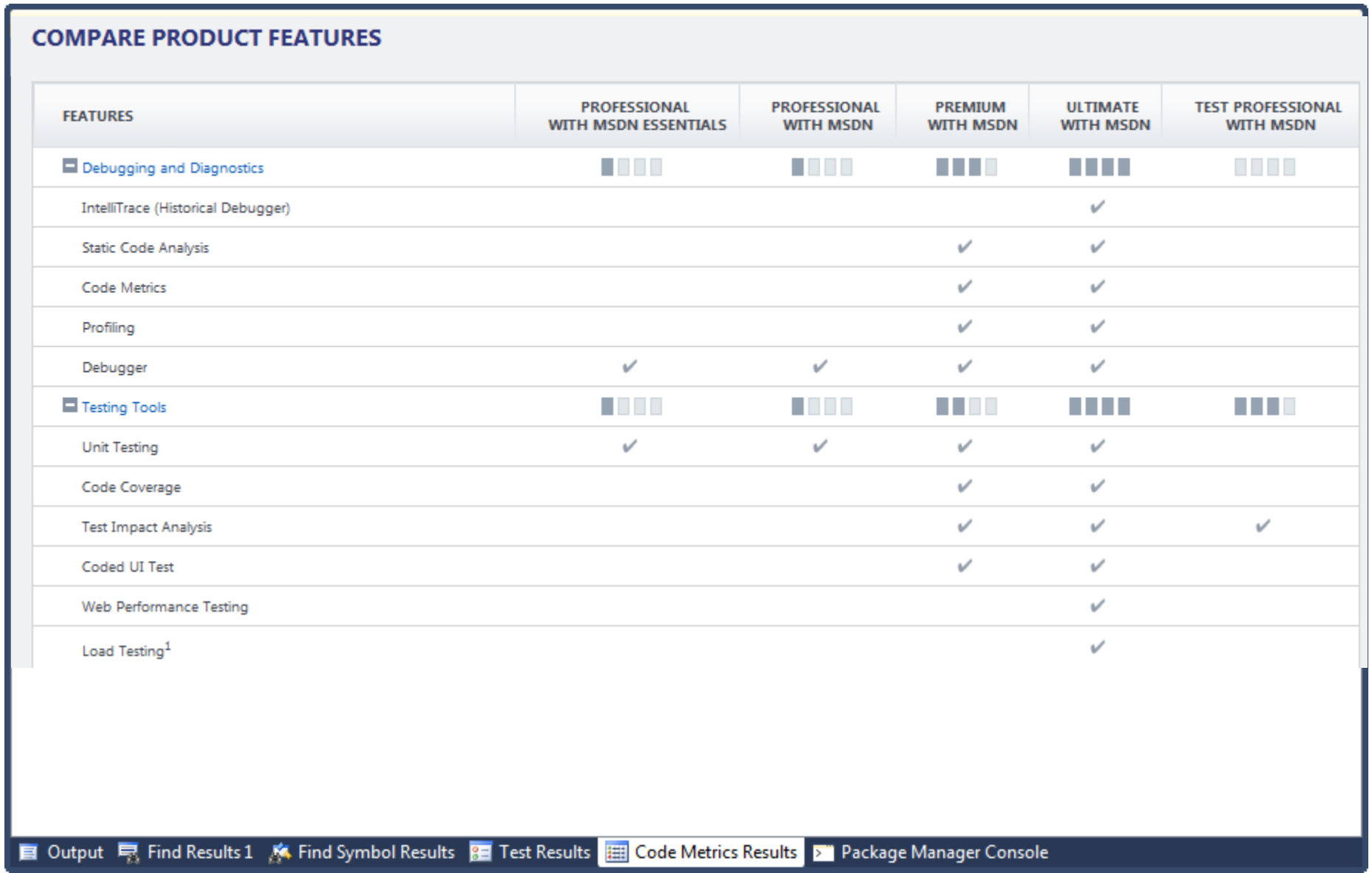
Assemblies:	50
Namespaces:	91
Types:	815
Methods:	2101
Fields:	16
Events:	0

Grid

199 Results

FullName	Cyclomatic	PhysicalLineCount	OutTypes	ParameterCount
LakeQuincy.Web.Admin.Adv.EditPlacement.SetStyles	34	106	21	2
LakeQuincy.Web.Controls_LQCreativeDisplay.SetDataSourceAndVisibility	32	83	10	0
LakeQuincy.Web.Profile.SqlTableProfileProvider.SetPropertyValues	31	175	33	2
LakeQuincy.Web.Admin.Reports.Adv.AdvertiserCampaignSummary.MonthlyDataRepeater_DataBi	27	84	17	2
.Controls_AddCreative.SubmitCreativeButton_Click	27	137	29	2
LakeQuincy.Web.Admin.Adv.EditPlacement.SaveAll	20	119	41	0
LakeQuincy.Web.Admin.Adv.RegisterAdvertiser.RegisterAdvertiserSubmitButton_Click	18	138	47	2
LakeQuincy.Web.Admin.Reports.Adv.ProposalReport.Page_Load	17	96	35	2
LakeQuincy.Business.CampaignRenewer.RenewCampaign	13	95	33	2
LakeQuincy.Web.Admin.Pub.Admin_Pub_UpdatePublisher.FillPublisherFields	13	84	37	0
LakeQuincy.Web.Admin.Pub.RegisterPublisher.SubmitButton_Click	12	104	49	2
LakeQuincy.Web.Admin.Pub.Admin_Pub_UpdatePublisher.SubmitButton_Click	12	112	48	2
LakeQuincy.Web.Contact.SendMessageButton_Click	12	80	31	2

Visual Studio



COMPARE PRODUCT FEATURES

FEATURES	PROFESSIONAL WITH MSDN ESSENTIALS	PROFESSIONAL WITH MSDN	PREMIUM WITH MSDN	ULTIMATE WITH MSDN	TEST PROFESSIONAL WITH MSDN
Debugging and Diagnostics	■□□□	■□□□	■■■□	■■■■	□□□□
IntelliTrace (Historical Debugger)				✔	
Static Code Analysis			✔	✔	
Code Metrics			✔	✔	
Profiling			✔	✔	
Debugger	✔	✔	✔	✔	
Testing Tools	■□□□	■□□□	■■□□	■■■■	■■■□
Unit Testing	✔	✔	✔	✔	
Code Coverage			✔	✔	
Test Impact Analysis			✔	✔	✔
Coded UI Test			✔	✔	
Web Performance Testing				✔	
Load Testing[1]				✔	

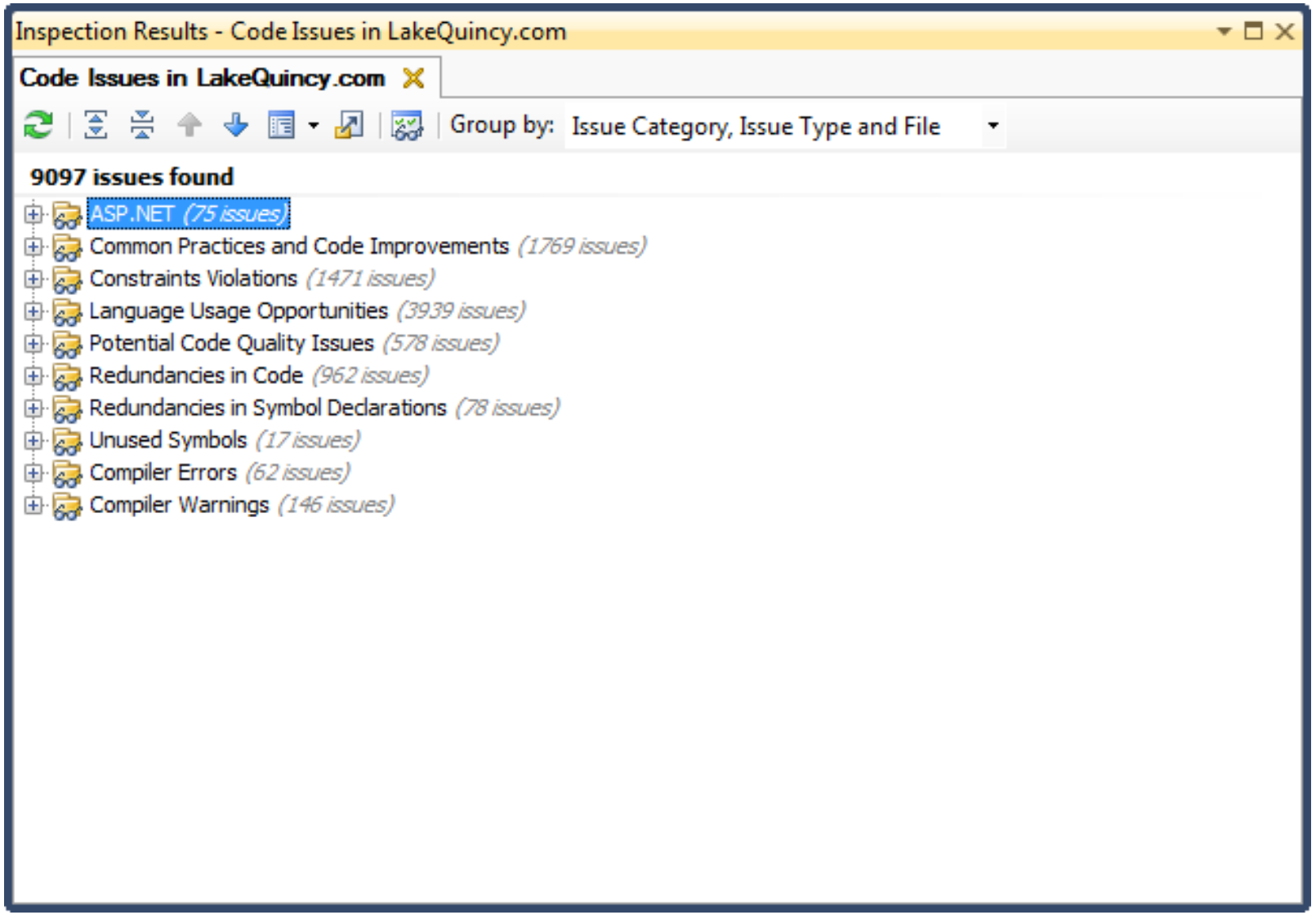
Output | Find Results 1 | Find Symbol Results | Test Results | Code Metrics Results | Package Manager Console

Visual Studio Versions

COMPARE PRODUCT FEATURES

FEATURES	PROFESSIONAL WITH MSDN ESSENTIALS	PROFESSIONAL WITH MSDN	PREMIUM WITH MSDN	ULTIMATE WITH MSDN	TEST PROFESSIONAL WITH MSDN
Debugging and Diagnostics	■□□□	■□□□	■■■□	■■■■	□□□□
IntelliTrace (Historical Debugger)				✔	
Static Code Analysis			✔	✔	
Code Metrics			✔	✔	
Profiling			✔	✔	
Debugger	✔	✔	✔	✔	
Testing Tools	■□□□	■□□□	■■□□	■■■■	■■■□
Unit Testing	✔	✔	✔	✔	
Code Coverage			✔	✔	
Test Impact Analysis			✔	✔	✔
Coded UI Test			✔	✔	
Web Performance Testing				✔	
Load Testing[1]				✔	

Resharper

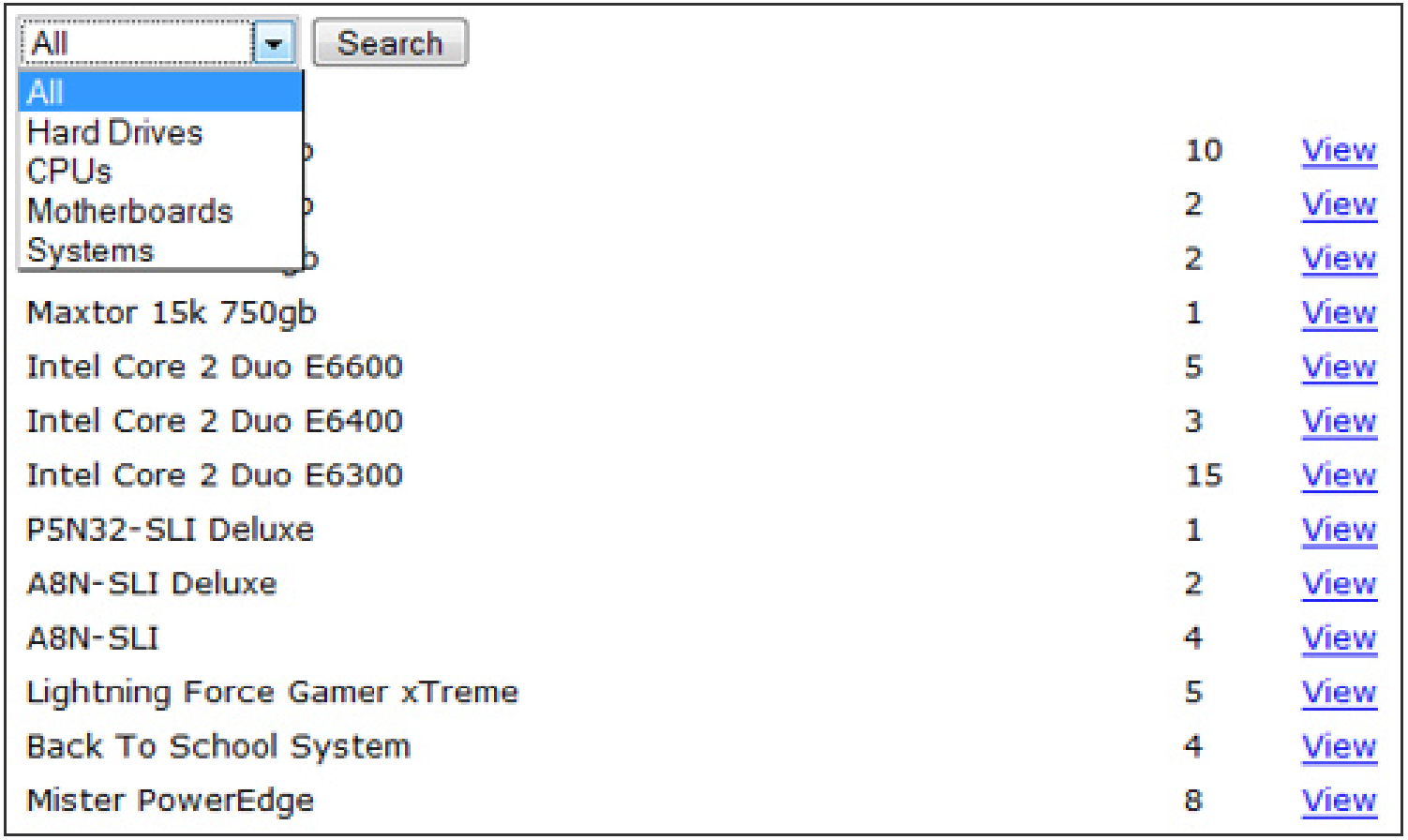

Inspection Results - Code Issues in LakeQuincy.com
Code Issues in LakeQuincy.com
Group by: Issue Category, Issue Type and File
9097 issues found
ASP.NET (75 issues)
Common Practices and Code Improvements (1769 issues)
Constraints Violations (1471 issues)
Language Usage Opportunities (3939 issues)
Potential Code Quality Issues (578 issues)
Redundancies in Code (962 issues)
Redundancies in Symbol Declarations (78 issues)
Unused Symbols (17 issues)
Compiler Errors (62 issues)
Compiler Warnings (146 issues)

Code Review

- ASP.NET Form

- ETL Processing

ASP.NET Form: Search Products



Codebehind Methods: OnLoad

```
protected override void OnLoad(EventArgs e)
{
    int defaultCategory;
    try
    {
        defaultCategory = Int32.Parse(Request.QueryString["CategoryId"]);
    }
    catch (Exception ex)
    {
        defaultCategory = -1;
    }

    Results.DataSource = GetResults(defaultCategory);
    Results.DataBind();

    if (!Page.IsPostBack)
    {
        CategoryList.DataSource = GetCategories();
        CategoryList.DataTextField = "Name";
        CategoryList.DataValueField = "Id";
        CategoryList.DataBind();
        CategoryList.Items.Insert(0, new ListItem("All", "-1"));
        CategoryList.SelectedIndex = CategoryList.Items.IndexOf(CategoryList.Items.FindByValue(defaultCateg
        base.OnLoad(e);
    }


}
```

Codebehind Methods: Search_Click

```
private void Search_Click(object sender, EventArgs e)
{
    Results.DataSource = GetResults(Convert.ToInt32(CategoryList.SelectedValue));
    Results.DataBind();
}
```

Codebehind Methods: GetCategories

```
private DataTable GetCategories()
{
    if (Cache["AllCategories"] != null)
    {
        return (DataTable) Cache["AllCategories"];
    }

    SqlConnection connection = new SqlConnection("Data Source=DB;Initial Catalog=Store;User Id=User;Passwo
    string sql = string.Format("SELECT * From Categories");
    SqlCommand command = new SqlCommand(sql, connection);

    SqlDataAdapter da = new SqlDataAdapter(command);
    DataTable dt = new DataTable();

    da.Fill(dt);
    Cache.Insert("AllCategories", dt, null, DateTime.Now.AddHours(1), System.Web.Caching.Cache.NoSlidingEx

    connection.Dispose();
    return dt;

}
```

Codebehind Methods: GetResults

```
private DataTable GetResults(int categoryId)
{
    SqlConnection connection = new SqlConnection("Data Source=DB;Initial Catalog=Store;User Id=User;Passwo
    string sql = string.Format("SELECT * FROM Products P INNER JOIN Categories C on P.CategoryId = C.Id WH
    SqlCommand command = new SqlCommand(sql, connection);

    SqlDataAdapter da = new SqlDataAdapter(command);
    DataTable dt = new DataTable();

    da.Fill(dt);

    connection.Dispose();
    return dt;
}
```

ETL - Original

FullName	Cyclomatic	PhysicalLineCount	OutTypes	ParameterCount
ExtractTransformLoad.Program.Load	4	70	17	0
ExtractTransformLoad.Program.Transform	9	60	17	0
ExtractTransformLoad.Program.Extract	3	41	18	0
ExtractTransformLoad.Program.Main	1	14	5	1
ExtractTransformLoad.Program..ctor	0	0	2	0
ExtractTransformLoad.Employee..ctor	0	0	2	0
ExtractTransformLoad.FreightByShipper..ctor	0	0	2	0

ETL - Refactored

FullName	Cyclomatic	PhysicalLineCount	OutTypes	ParameterCount
ExtractTransformLoad.Services.TransformService.Transform	3	26	15	1
ExtractTransformLoad.Impl.SqlFreightByShipperRepository.DeleteAndInsert	3	25	18	2
ExtractTransformLoad.Impl.SqlEmployeeBonusRepository.DeleteAndInsert	3	24	19	2
ExtractTransformLoad.Services.ExtractionService.Extract	4	18	15	0
ExtractTransformLoad.Services.LoadService.Load	3	18	18	1
ExtractTransformLoad.Impl.SqlEmployeeRepository.List	1	16	16	0
ExtractTransformLoad.Impl.SqlInvoiceRepository.List	2	15	15	1
ExtractTransformLoad.Domain.EmployeeFactory.Create	2	8	6	2
ExtractTransformLoad.Program.Main	2	7	10	1

Self-Improvement and Quality

- How fast can you produce:
  - Code you believe to be of high quality
  - Code that maybe gets the job done, but you believe to be of low quality

- Which one can you produce more quickly?
- Why?
- How can we develop our skills and our tools so that building quality is *natural* and *easier* than not doing so?

Software Craftsmanship Calendar

- http://bit.ly/SC_2011

unclebobmartin
Nov 05, 4:02pm via Twitter for iPhone
Calendar of craftsmanship principles: http://bit.ly/SC_2011

RonJeffries
Nov 05, 4:08pm via TweetDeck
RT @unclebobmartin: Calendar of craftsmanship principles: http://bit.ly/SC_2011 [cool]

alanstevens
Nov 05, 12:35pm via TweetDeck
I'm digging the software craftsmanship motivation calendar. Get one for your team room: http://bit.ly/c0WEmb

brianhprince
Nov 02, 11:46am via TweetDeck
Software Craftsmanship 2011 Motivational Calendar : http://bit.ly/c0WEmb : get yours today

elijahmanor
Nov 05, 11:04am via Chromed Bird
Pre-order your 2011 Software Craftsmanship Calendar today! by @nimblepros http://bit.ly/SC_2011 Stay SOLID all year long ;)

Additional Reading

- Pirsig, Robert M. *Zen and the Art of Motorcycle Maintenance: an Inquiry into Values*, New York: Morrow, 1974. Print.
- "Software Quality." *Wikipedia, the Free Encyclopedia*. Web. 8 Nov 2010 http://en.wikipedia.org/wiki/Software_quality
- McConnell, Steve. *Code Complete.* Redmond, WA: Microsoft, 2004. Print.
- "The Four Elements of Simple Design." Web. 8 Nov 2010 http://www.jbrains.ca/permalink/the-four-elements-of-simple-design

References

- http://semmle.com/semmlecode/documentation/semmlecode-glossary/lack-of-cohesion-of-methods-of-a-type-henderson-sellers/
- http://www.codeproject.com/KB/architecture/Cyclomatic_Complexity.aspx
- http://portal.acm.org/citation.cfm?id=42379
- http://en.wikipedia.org/wiki/Halstead_complexity_measures
- http://codebetter.com/blogs/karlseguin/archive/2006/12/01/How-to-hire-a-programmer-_2D00_-Part-2-_2D00_-Improve-this-code.aspx
- http://www.microsoft.com/visualstudio/en-us/products#compareTable

- The ETL Demos are available as part of this course (the DRY modules):
- http://www.pluralsight-training.net/microsoft/olt/Course/Toc.aspx?n=principles-oo-design