Task Refactoring

Maximizing Your Personal Throughput

Eric Lynn
DeltaPoint Solutions
@deltapoint

Platinum Sponsors

SyllogisTeks

Microsoft Azure

slalom

Team Daugherty Est. 1985

Insight Digital Innovation

Polaris Solutions

ArchitectNow

Celebration Sponsor

Team Daugherty Est. 1985

Lanyards Sponsor

Greenfield Creative

Notebook Sponsor

SyllogisTeks

Registration Sponsor

Microsoft

Gold Sponsors

Keyhole Software

Microsoft

MiTek

Adaptive Solutions Group

VantageLinks

ungerboeck

Express Scripts

Covenant Technology Partners

JetBrains

twilio

Algorand

((CENTRIC))

Fast Reports
Reporting must be Fast!

TriZetto Provider Solutions
A Cognizant Company

Allscripts
All possible

Robert Half Technology

TEKsystems
Own change

mongoDB.

STIFEL

Greenfield Creative

Silver Sponsors

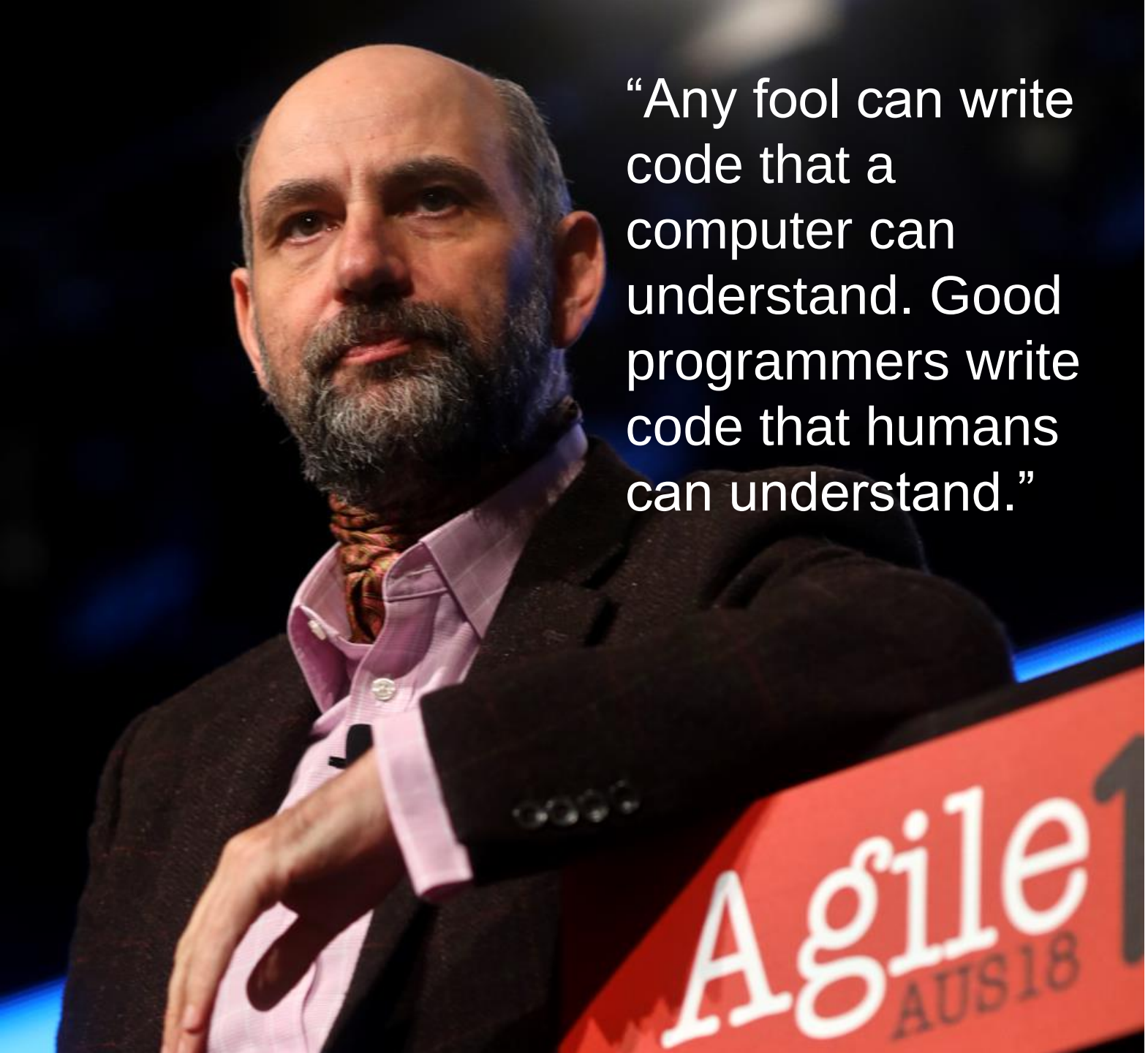
CODE Magazine

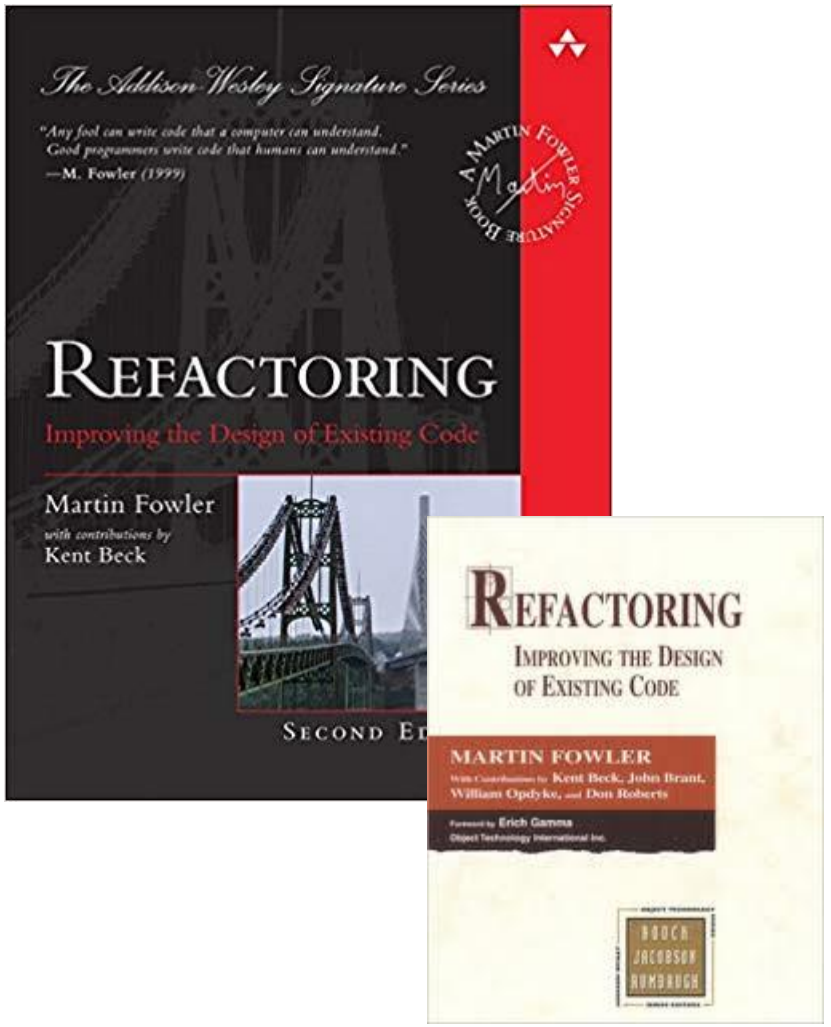
oasis digital

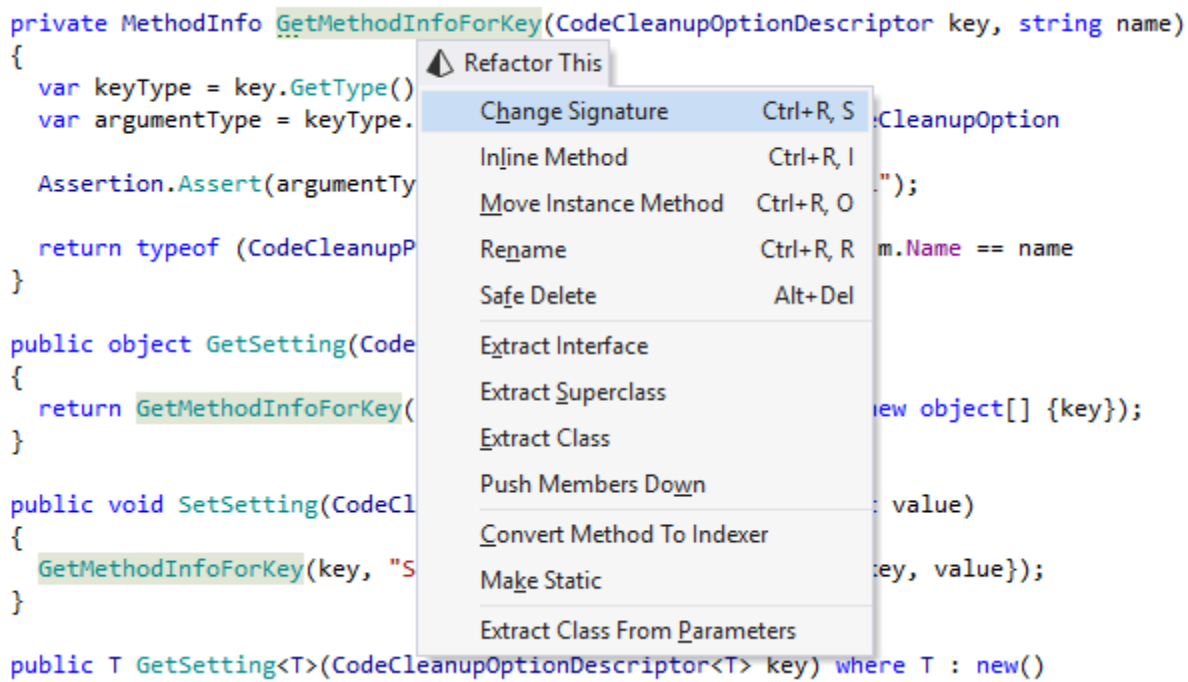
Hunter Engineering Company


SAINT GEORGE

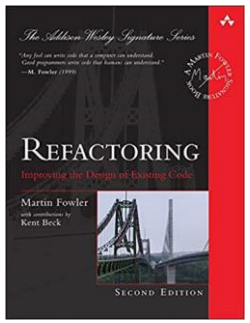
The Key Question


“Any fool can write code that a computer can understand. Good programmers write code that humans can understand.”
Agile
AUS 18


The Addison Wesley Signature Series
"Any fool can write code that a computer can understand. Good programmers write code that humans can understand."
—M. Fowler (1999)
A Martin Fowler Signature Book
REFACTORING
Improving the Design of Existing Code
Martin Fowler
with contributions by
Kent Beck
REFACTORING
IMPROVING THE DESIGN OF EXISTING CODE
MARTIN FOWLER
Kent Beck, John Brant, William Opdyke, and Don Roberts
Erich Gamma
BOOCH
JACOBSON
RUMBAUGH


private MethodInfo GetMethodInfoForKey(CodeCleanupOptionDescriptor key, string name)
{
var keyType = key.GetType()
var argumentType = keyType.
Assertion.Assert(argumentTy
return typeof (CodeCleanupP
}
public object GetSetting(Code
{
return GetMethodInfoForKey(
}
public void SetSetting(CodeCl
{
GetMethodInfoForKey(key, "S
}
public T GetSetting<T>(CodeCleanupOptionDescriptor<T> key) where T : new()
Refactor This
Change Signature Ctrl+R, S
Inline Method Ctrl+R, I
Move Instance Method Ctrl+R, O
Rename Ctrl+R, R
Safe Delete Alt+Del
Extract Interface
Extract Superclass
Extract Class
Push Members Down
Convert Method To Indexer
Make Static
Extract Class From Parameters

Value of Refactoring



Refactor to understand

Refactor to improve the design

Refactor to allow change to happen smoothly and efficiently

Others?

This is the online catalog of refactorings, to support my book Refactoring 2nd Edition.

This catalog of refactorings includes those refactorings described in my original book on Refactoring, together with the Ruby Edition.

Using the Catalog ▶

Tags

- basic
- encapsulation
- moving-features
- organizing-data
- simplify-conditional-logic
- refactoring-apis
- dealing-with-inheritance
- collections
- delegation
- errors
- extract
- parameters
- fragments
- grouping-function
- immutability
- inline
- remove
- rename
- split-phase
- variables

#

Change Function Declaration
Add Parameter • Change Signature • Remove Parameter • Rename Function • Rename Method

Change Reference to Value

Change Value to Reference

Collapse Hierarchy

Combine Functions into Class

Combine Functions into Transform

Consolidate Conditional Expression

Decompose Conditional

Encapsulate Collection

Encapsulate Record
Replace Record with Data Class

Encapsulate Variable
Encapsulate Field • Self-Encapsulate Field

Extract Class

Extract Function
Extract Method

Extract Superclass

Extract Variable
Introduce Explaining Variable

Hide Delegate

Remove Dead Code

Remove Flag Argument
Replace Parameter with Explicit Methods

Remove Middle Man

Remove Setting Method

Remove Subclass
Replace Subclass with Fields

Rename Field

Rename Variable

Replace Command with Function

Replace Conditional with Polymorphism

Replace Constructor with Factory Function
Replace Constructor with Factory Method

Replace Control Flag with Break
Remove Control Flag

Replace Derived Variable with Query

Replace Error Code with Exception

Replace Exception with Precheck
Replace Exception with Test

Replace Function with Command
Replace Method with Method Object

Replace Inline Code with Function Call

Task Refactoring Catalog

- Enable Task
- Decompose Project
- Delegate and Await
- Delegate Task
- Demote Project to Task
- Demote to Backlog
- Demote to Candidate
- Extract Project from Goal

- Promote Candidate to Task
- Promote to Project
- Pull from Calendar
- Push to Calendar
- Rename to Actionable
- Remove Low-Value Task
- Split Task


Wood-Mizer

Q & A

Platinum Sponsors

SyllogisTeks

Microsoft Azure

slalom

Team Daugherty Est. 1985

Insight Digital Innovation

Polaris Solutions

ArchitectNow

Celebration Sponsor

Team Daugherty Est. 1985

Lanyards Sponsor

Greenfield Creative

Notebook Sponsor

SyllogisTeks

Registration Sponsor

Microsoft

Gold Sponsors

Keyhole Software

Microsoft

MiTek

Adaptive Solutions Group

VantageLinks

ungerboeck

Express Scripts

Covenant Technology Partners

JetBrains

twilio

Algorand

((CENTRIC))

Fast Reports
Reporting must be Fast!

TriZetto Provider Solutions
A Cognizant Company

Allscripts
All possible

Robert Half Technology

TEKsystems
Own change

mongoDB.

STIFEL

Greenfield Creative

Silver Sponsors

CODE Magazine

oasis digital

Hunter Engineering Company

Task Refactoring

Maximizing Your Personal Throughput by Sharpening Your “To Do” List

Eric Lynn
DeltaPoint Solutions
@deltapoint