

Write Code
For The
Future You

@paulmgower

TITANIUM SPONSORS



Platinum Sponsors



Gold Sponsors



Full Disclosure...



@paulmgower

Full Disclosure...

- Summer 2000

@paulmgower



Full Disclosure...

- Summer 2000
- Obfuscated Perl Contest

@paulmgower



Full Disclosure...

- Summer 2000
- Obfuscated Perl Contest
- Mentality was to write “elite code” that was hard to understand.

@paulmgower



Perspective Shift



@paulmgower

Perspective Shift

- First job out of college, 2001



@paulmgower

Perspective Shift

- First job out of college, 2001
- Alltel Client/Server University



@paulmgower

Perspective Shift

- First job out of college, 2001
- Alltel Client/Server University
 - Textbook: Code Complete
 - Daily Lectures, Labs, and Code Reviews



@paulmgower

Real World Learning



@paulmgower

Real World Learning

- Added to project towards end of development

@paulmgower



Real World Learning

- Added to project towards end of development
- Fixed bugs in code I didn't write

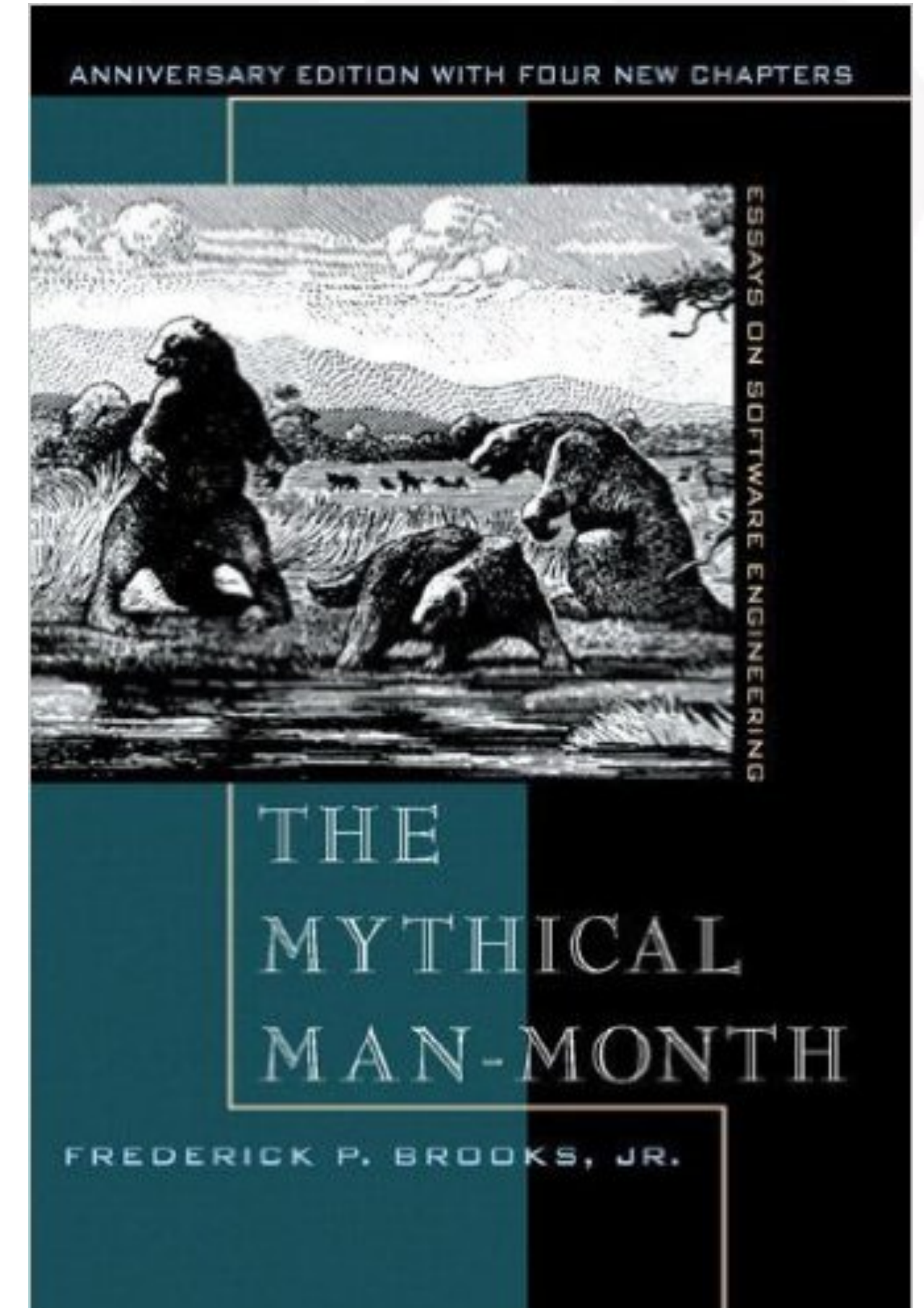
@paulmgower



Real World Learning

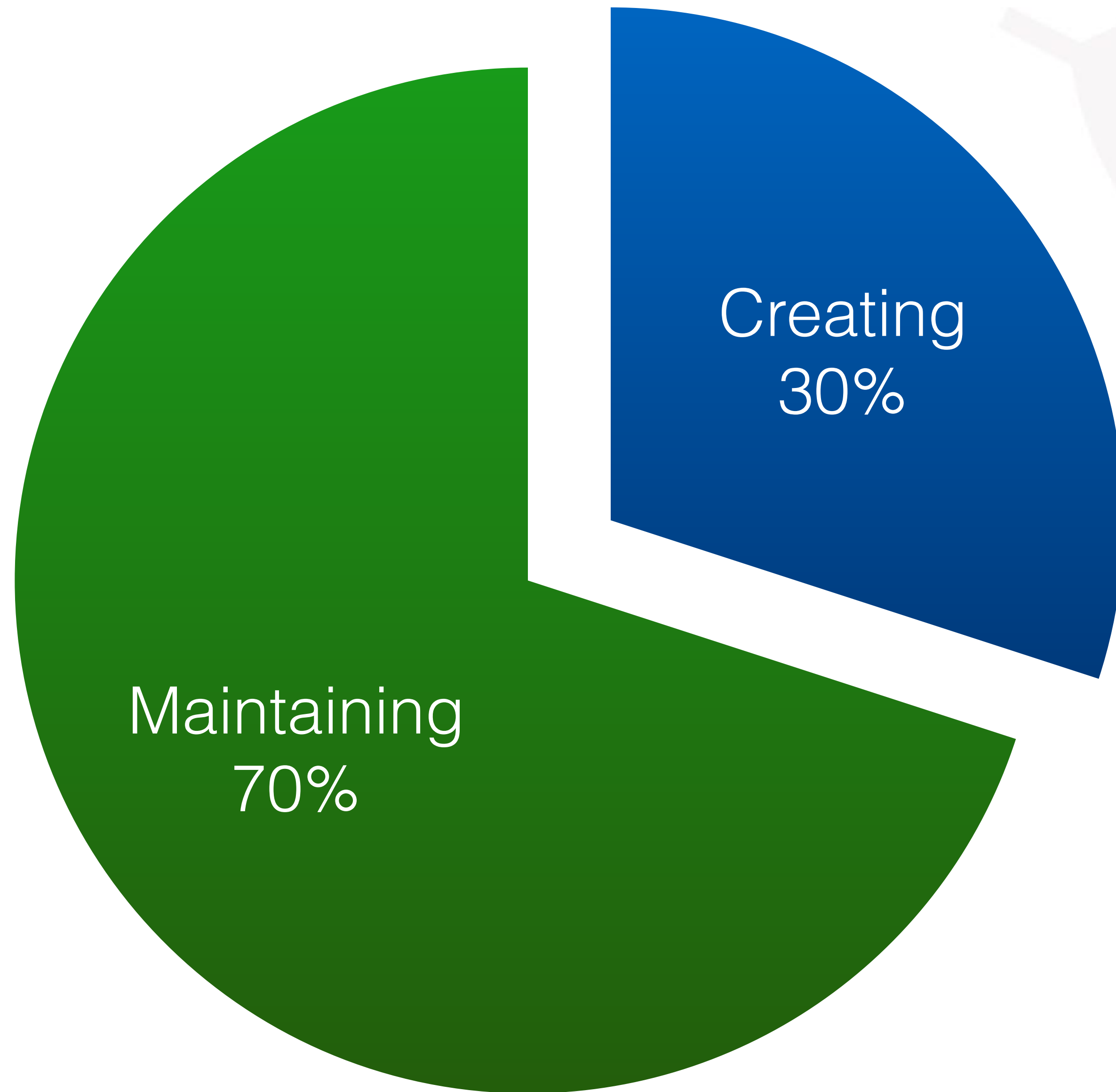
- Added to project towards end of development
- Fixed bugs in code I didn't write
- Reality of Mythical Man Month

@paulmgower



How much maintenance?

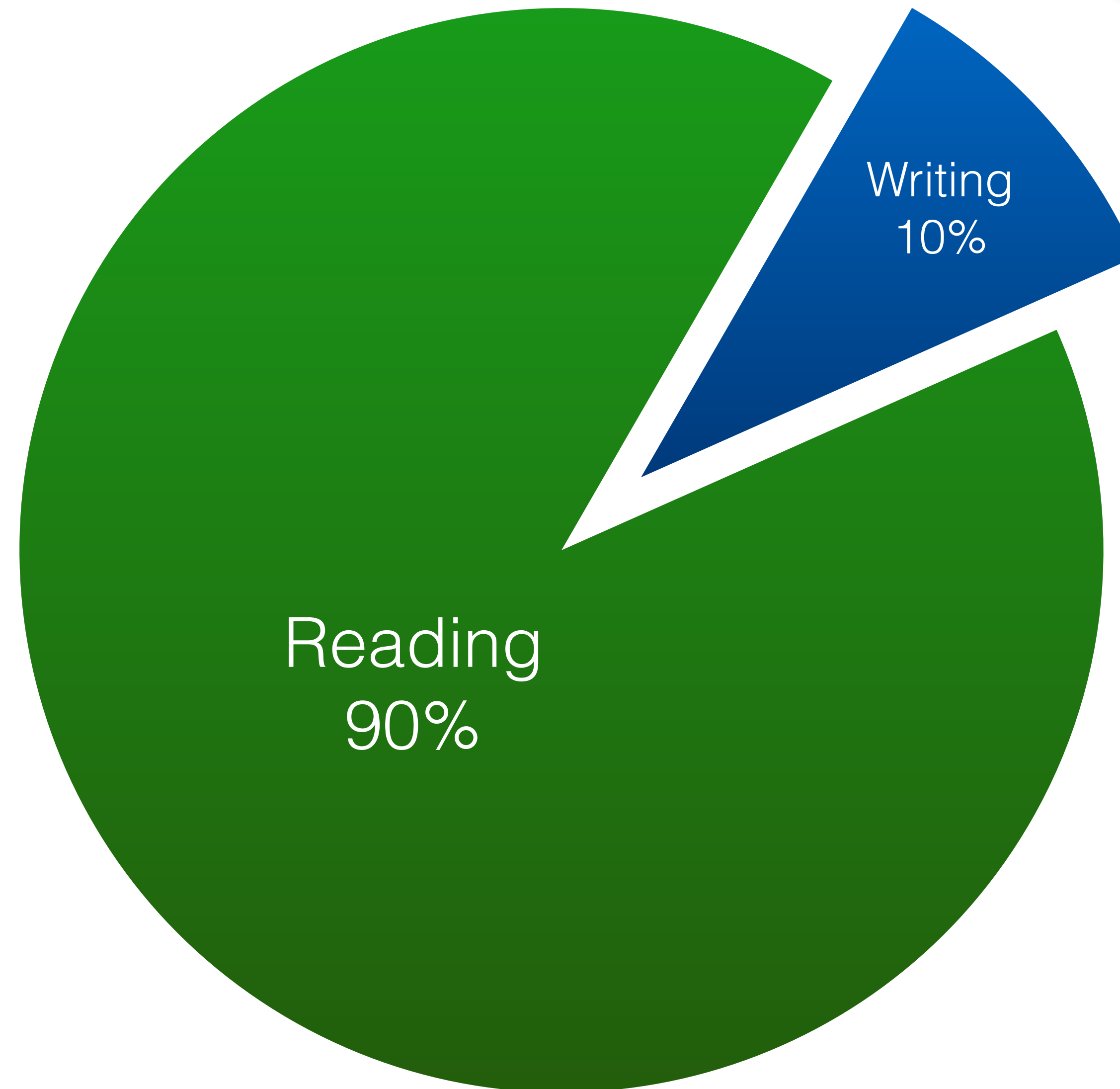
@paulmgower



@paulmgower

How much code reading?

@paulmgower



@paulmgower

Why code for reader?



@paulmgower

Why code for reader?

- Sept 1991, Comp.lang.c++ news group discussion

@paulmgower

Why code for reader?

- Sept 1991, Comp.lang.c++ news group discussion
- Why not use comma more?

@paulmgower

Why code for reader?

- Sept 1991, Comp.lang.c++ news group discussion
- Why not use comma more?
- Like this

```
if (condition)
    var = value, anothervar = anothervalue
```

@paulmgower

Why code for reader?

- Sept 1991, Comp.lang.c++ news group discussion
- Why not use comma more?


- Like this

```
if (condition)
    var = value, anothervar = anothervalue
```

- However consider this:

```
if (condition)
    var = value; anothervar = anothervalue
```

@paulmgower



“Always code as if the guy who ends up
maintaining your code will be a violent
psychopath who knows where you live”

John F. Woods

@paulmgower

How To Write Better Code?

@paulmgower

How To Write Better Code?

- Code Complete, 1993, 2004

@paulmgower

How To Write Better Code?

- Code Complete, 1993, 2004
- Manifesto for Agile Soft. Dev., 2001

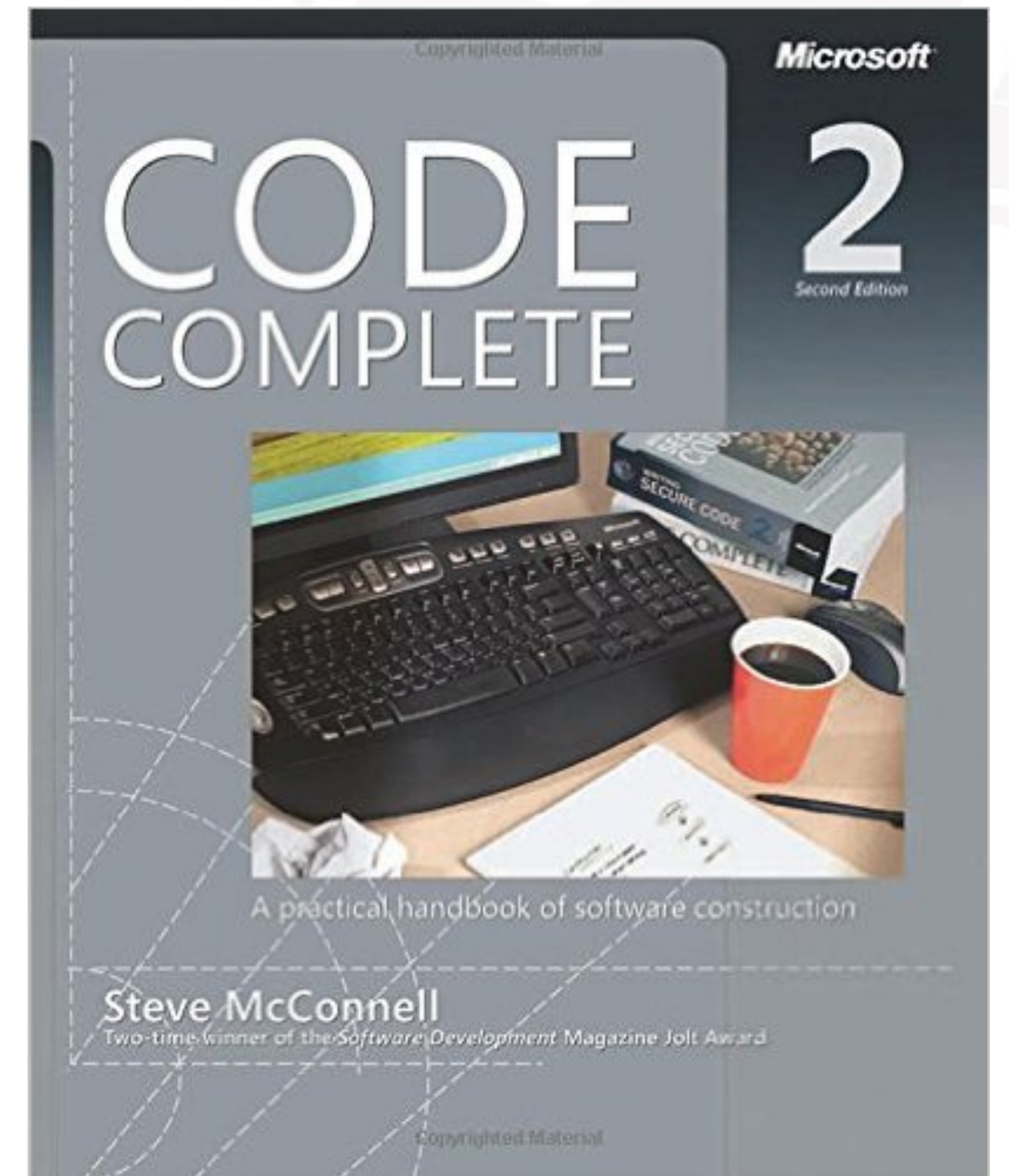
@paulmgower

How To Write Better Code?

- Code Complete, 1993, 2004
- Manifesto for Agile Soft. Dev., 2001
- Clean Code, 2009

@paulmgower

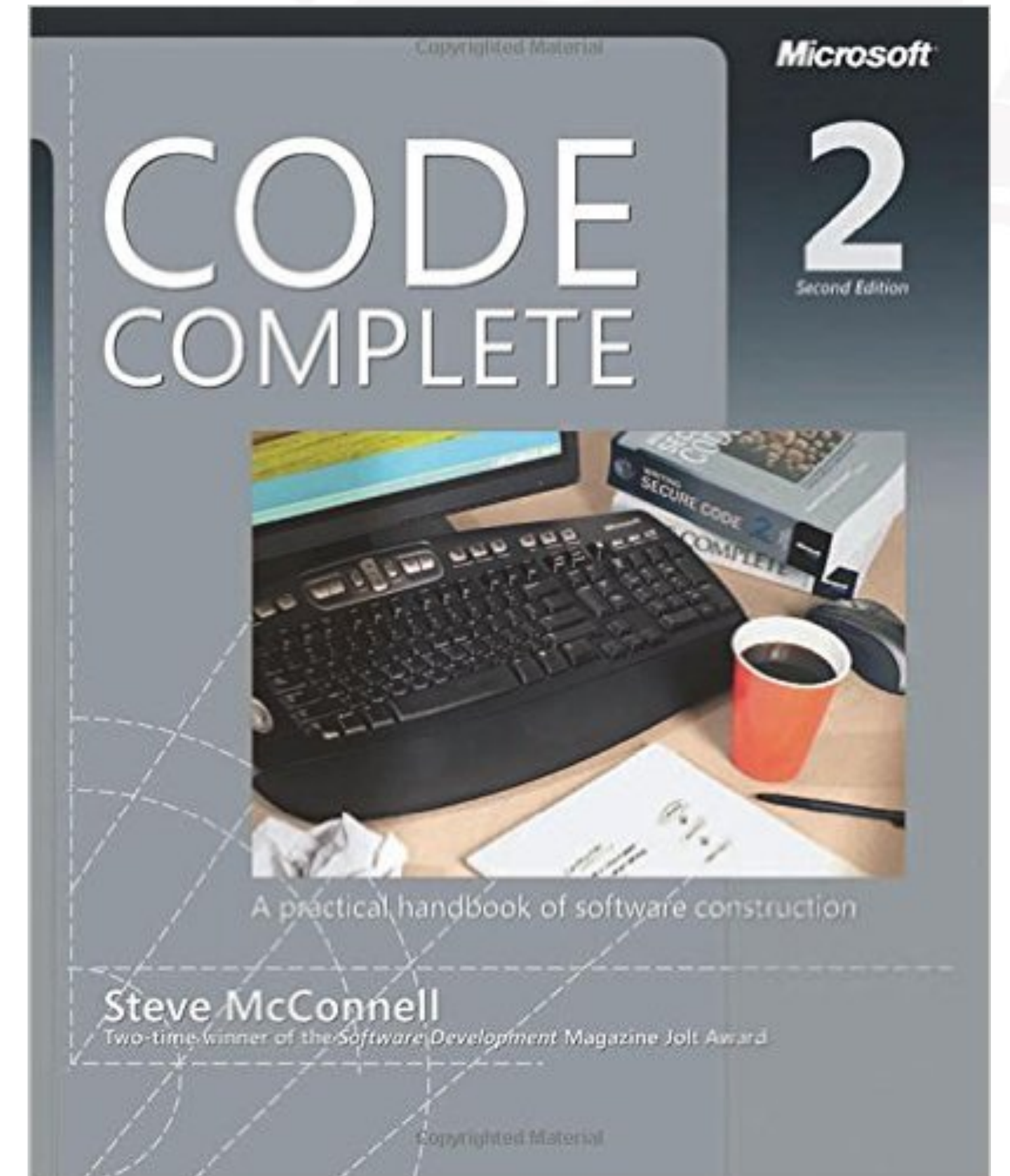
Variable Naming



@paulmgower

Variable Naming

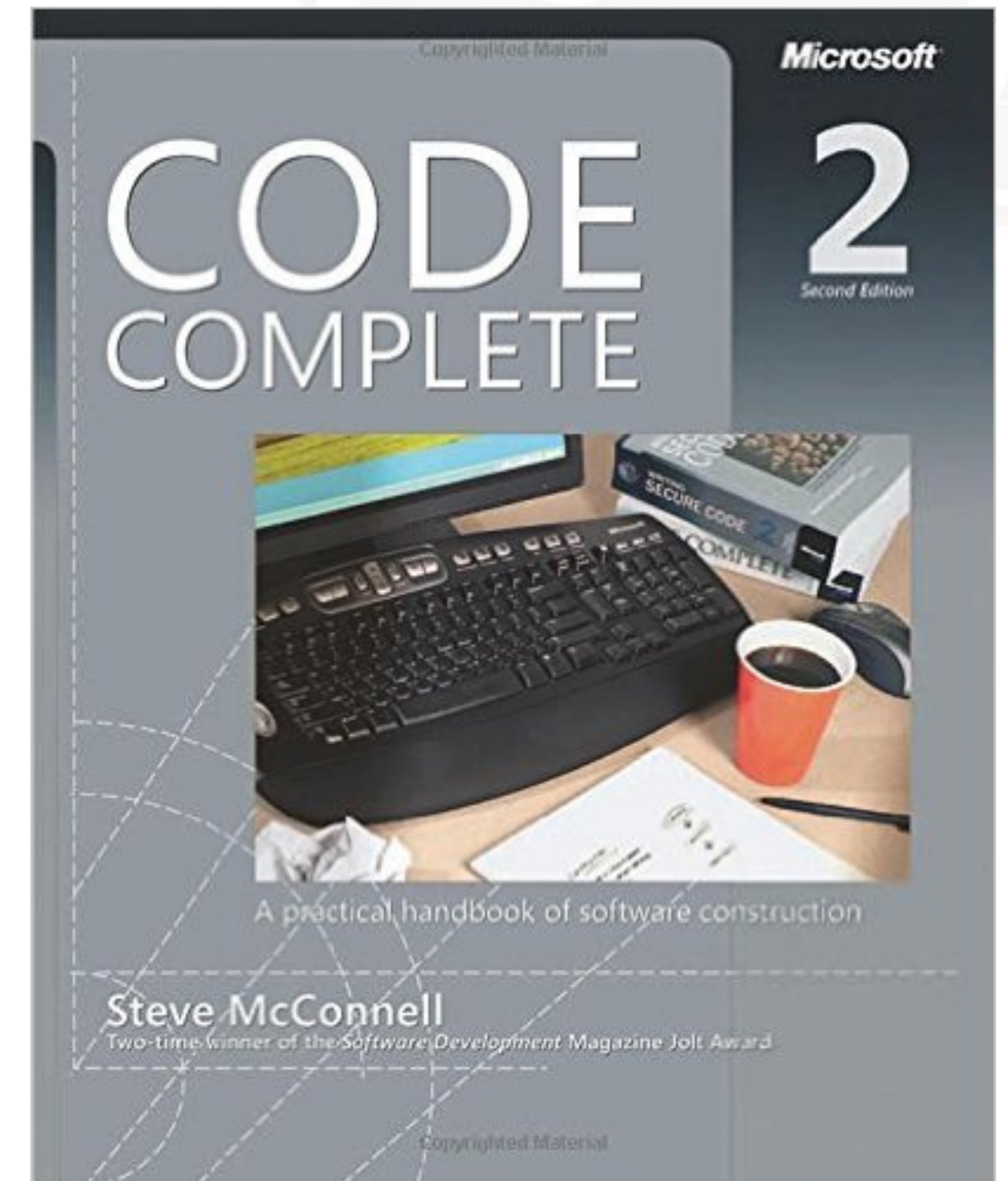
- Describe the entity it represents



@paulmgower

Variable Naming

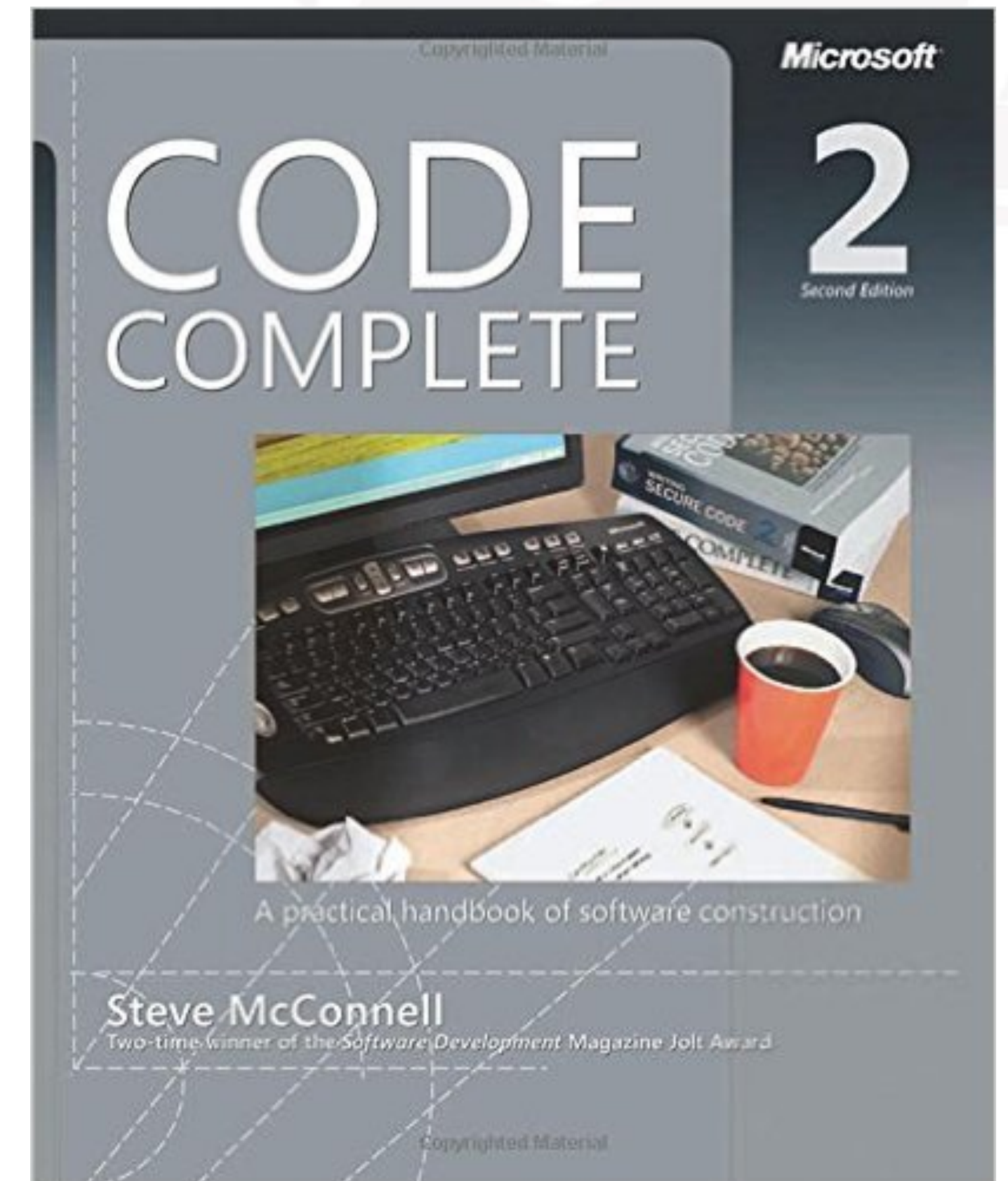
- Describe the entity it represents
- Easy to understand and read



@paulmgower

Variable Naming

- Describe the entity it represents
- Easy to understand and read
- Be specific as possible



@paulmgower

Variable Name Examples

Bad Names

- total, ct, checks, CHKTTL
- velt, tv, train
- cd, current, c, date
- lpp, lines

Good Names

- runningTotal, checkTotal
- trainVelocity, velocityInMph
- currentDate, today
- linesPerPage

Magic Numbers



@paulmgower

Magic Numbers



```
if (password.length > 12) {  
    ...  
}
```

Magic Numbers == BAD

- Don't use magic numbers

Magic Numbers == BAD

- Don't use magic numbers
- Use constants, enums, and/or config

Magic Numbers == BAD

- Don't use magic numbers
- Use constants, enums, and/or config
- Self documenting code

@paulmgower

Don't Do This



```
public void setPassword(String password) {  
    // don't do this  
    if (password.length() > 7) {  
        throw new IllegalArgumentException("password");  
    }  
}
```

@paulmgower

Do This

```
public void setPassword(String password) {  
    const int MAX_PASSWORD_SIZE = 7;  
    if (password.length() > MAX_PASSWORD_SIZE)  
    {  
        throw new IllegalArgumentException("password");  
    }  
}
```

@paulmgower

D.R.Y.



@paulmgower

D.R.Y.

- Duplication leads to maintenance



D.R.Y.

- Duplication leads to maintenance
- Y2K Culprit?



D.R.Y.

- Duplication leads to maintenance
- Y2K Culprit?

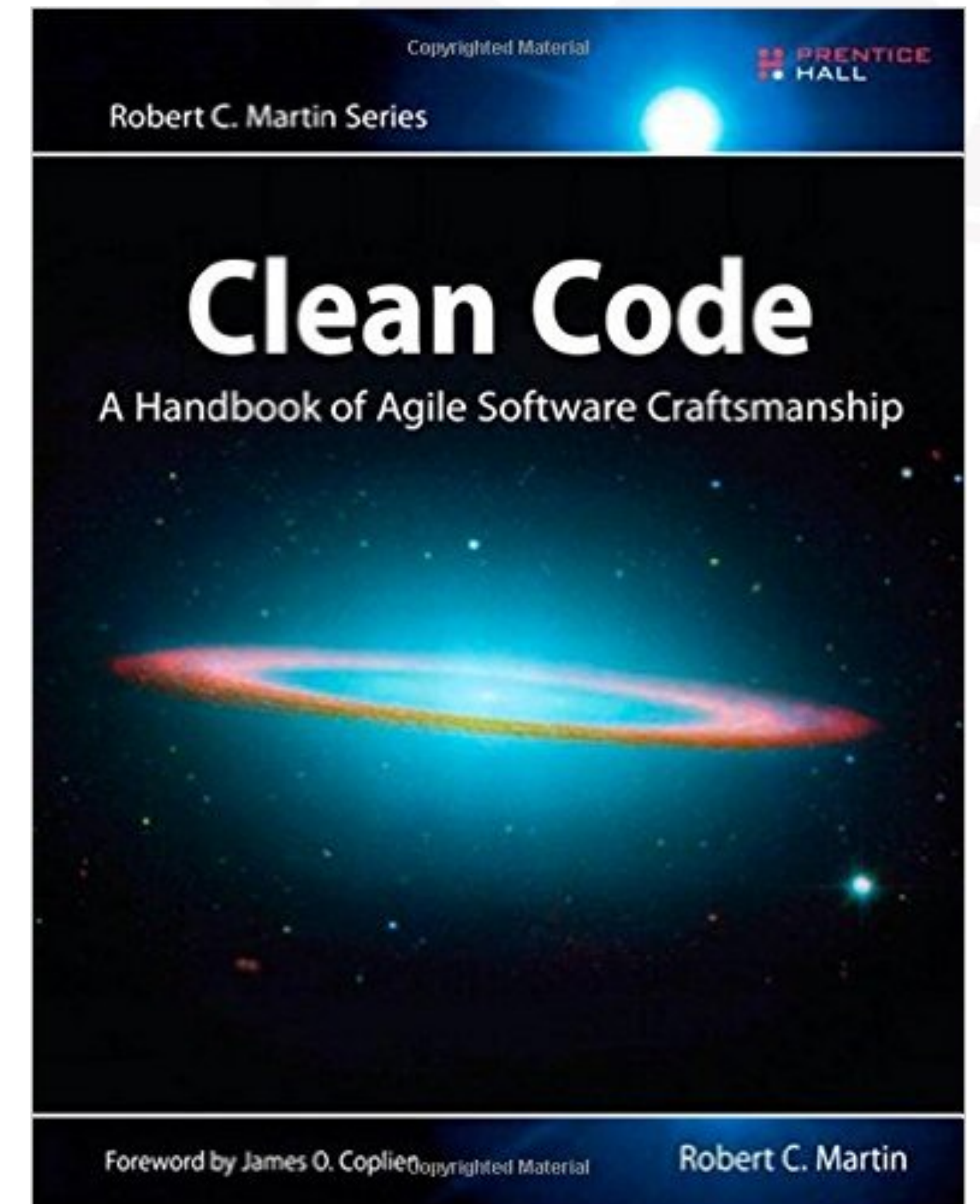
“Every piece of knowledge must have a single, unambiguous, authoritative representation within a system”

–Dave Thomas

@paulmgower



Single Responsibility

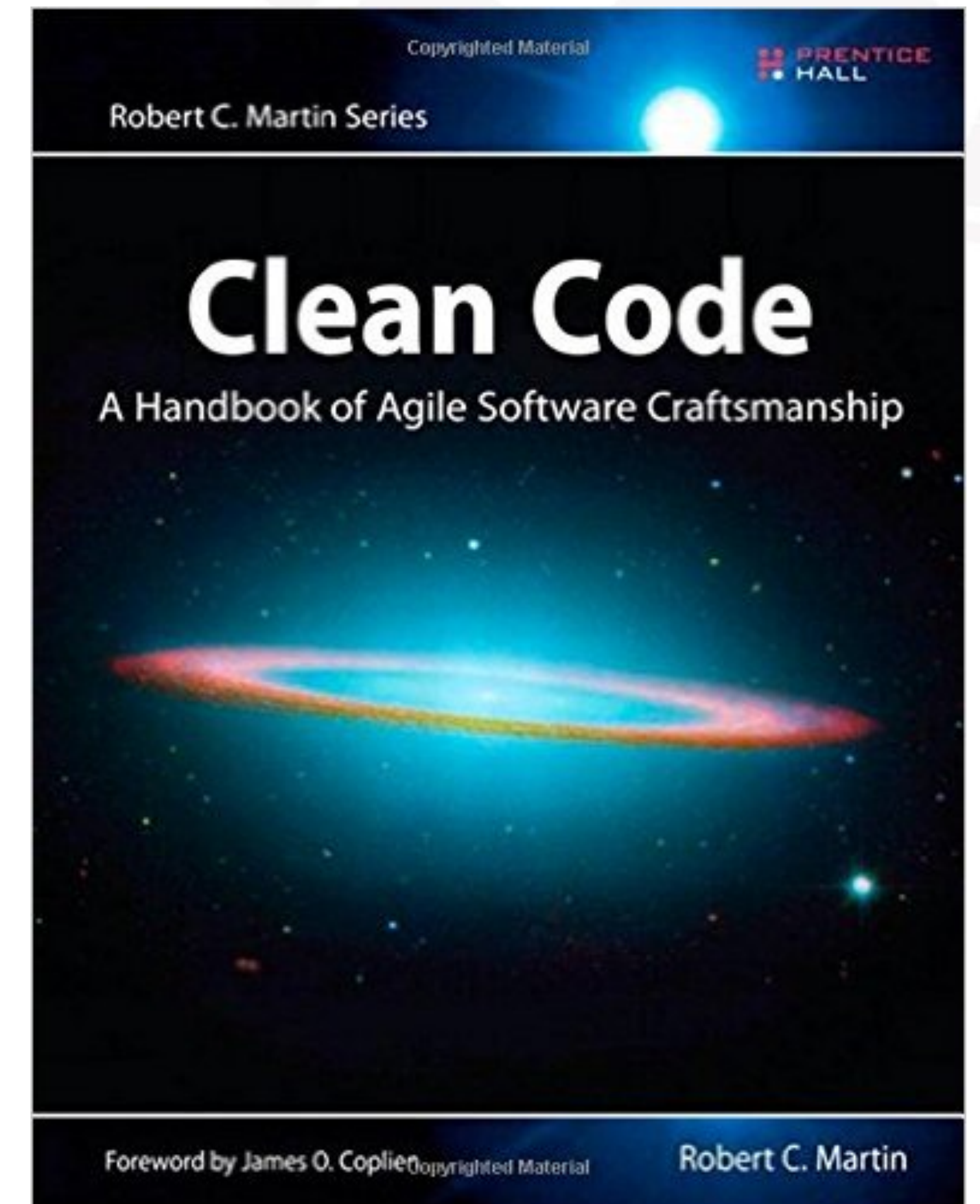


@paulmgower

Single Responsibility

- Responsibility == Axis of change

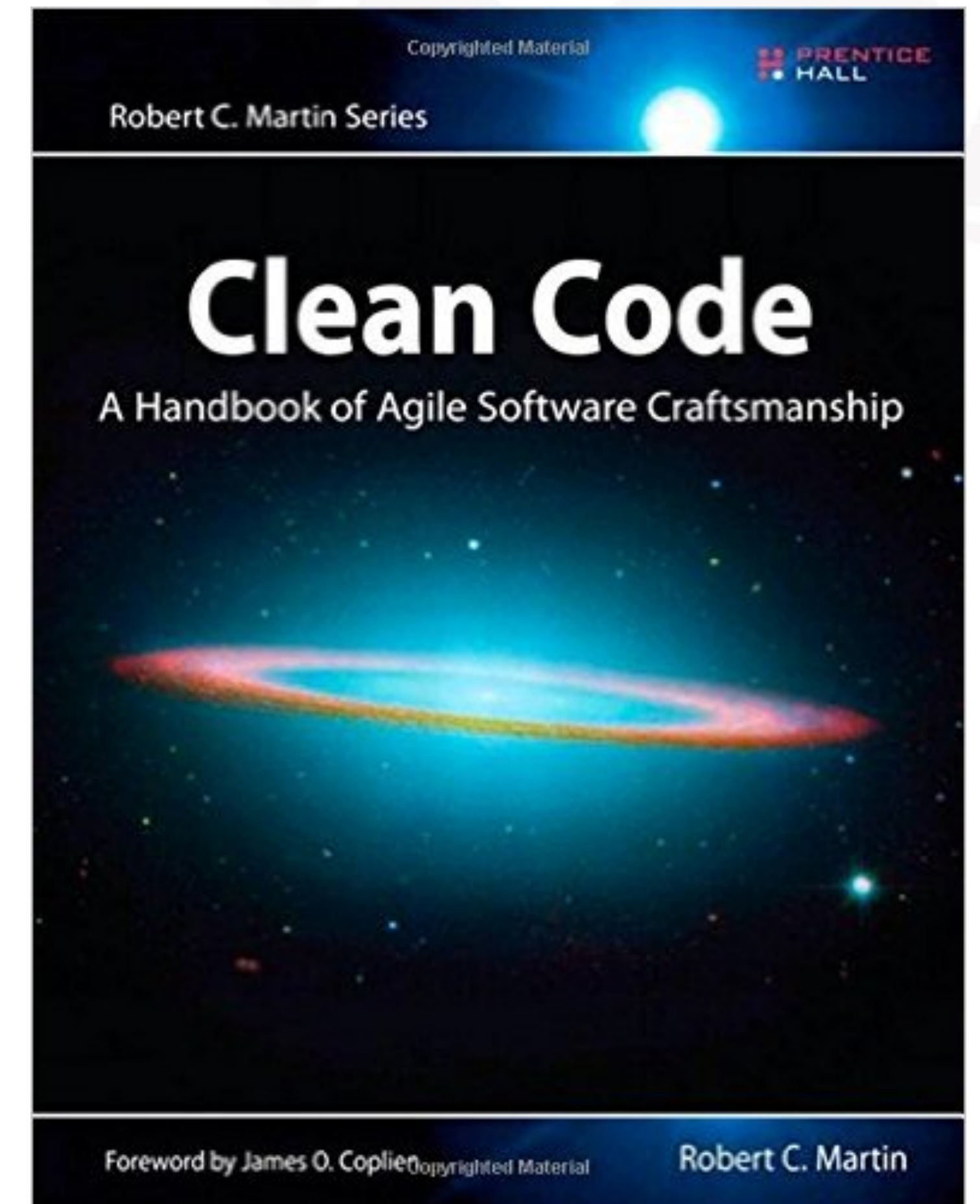
@paulmgower



Single Responsibility

- Responsibility == Axis of change
- Change will happen

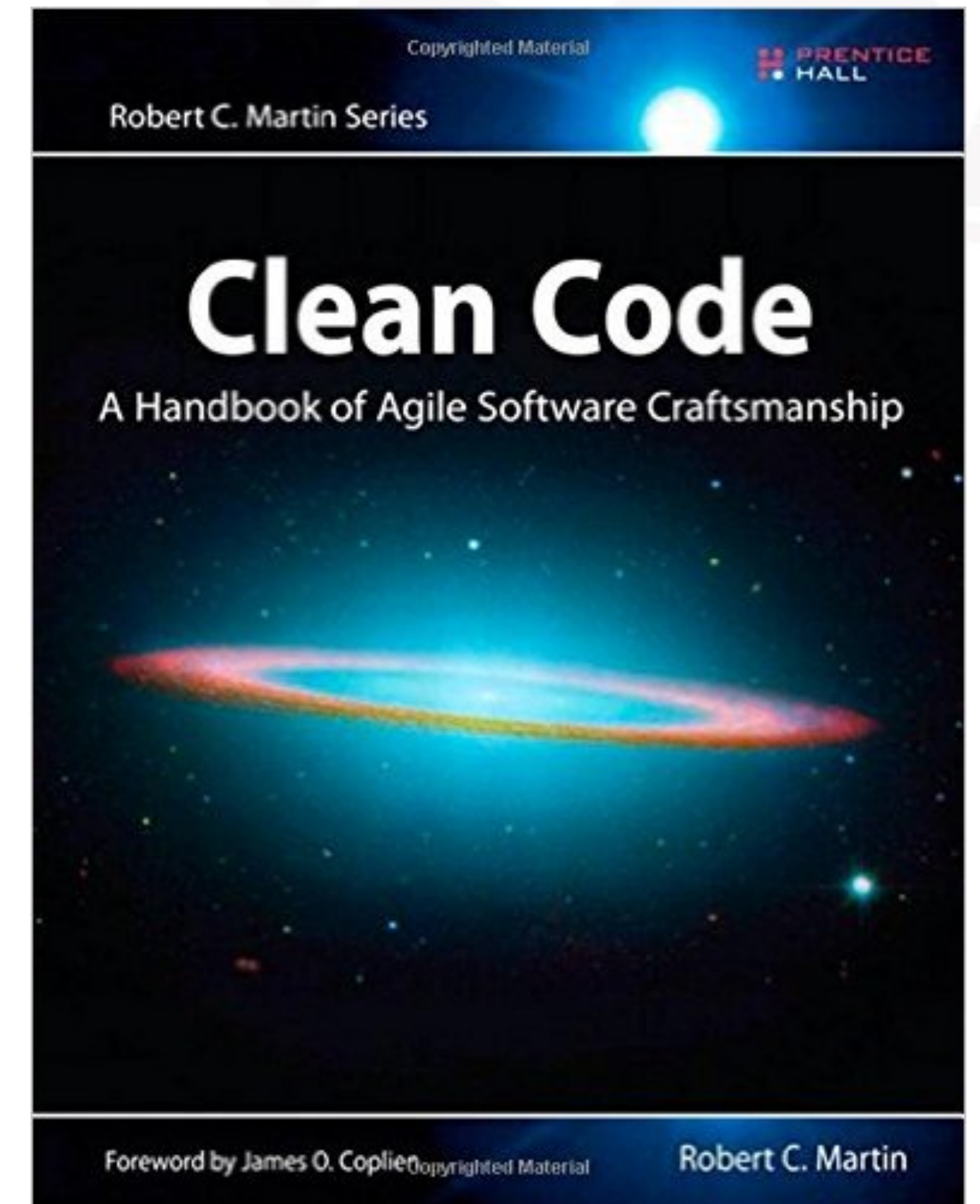
@paulmgower



Single Responsibility

- Responsibility == Axis of change
- Change will happen
- Loose Coupling

@paulmgower



More than 1 Responsibility

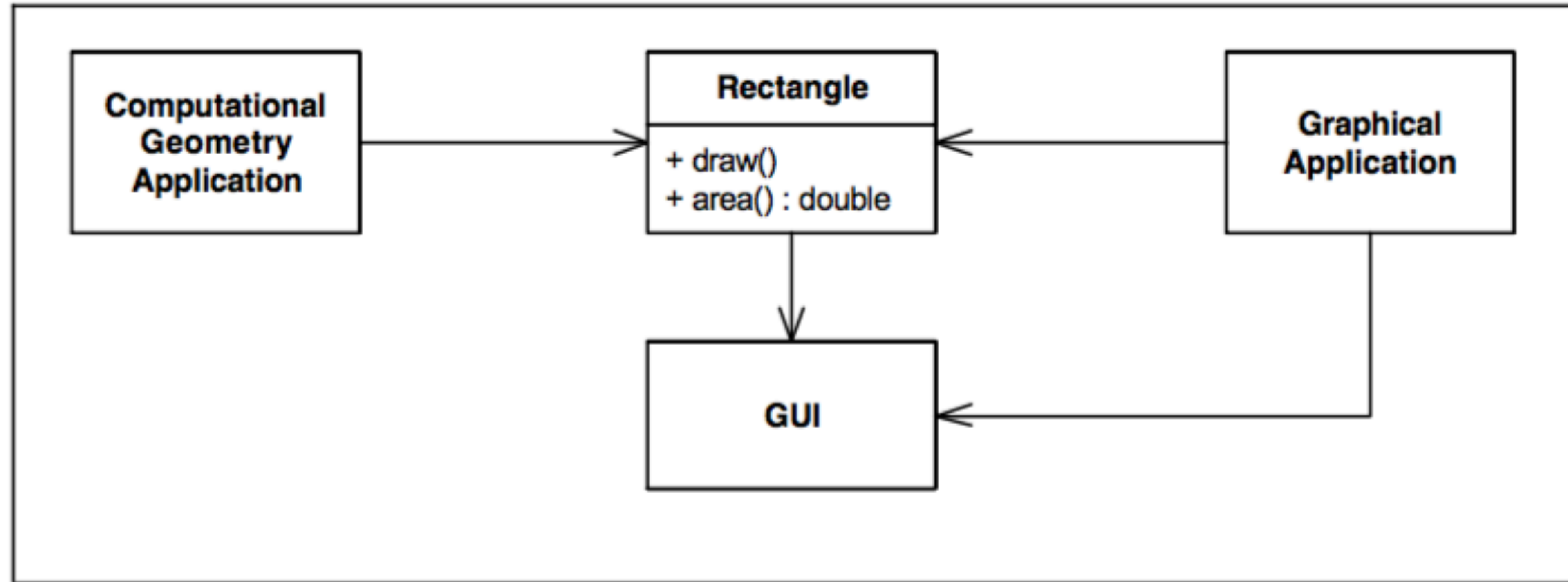


Figure 8-1
More than one responsibility

@paulmgower

Single Responsibility

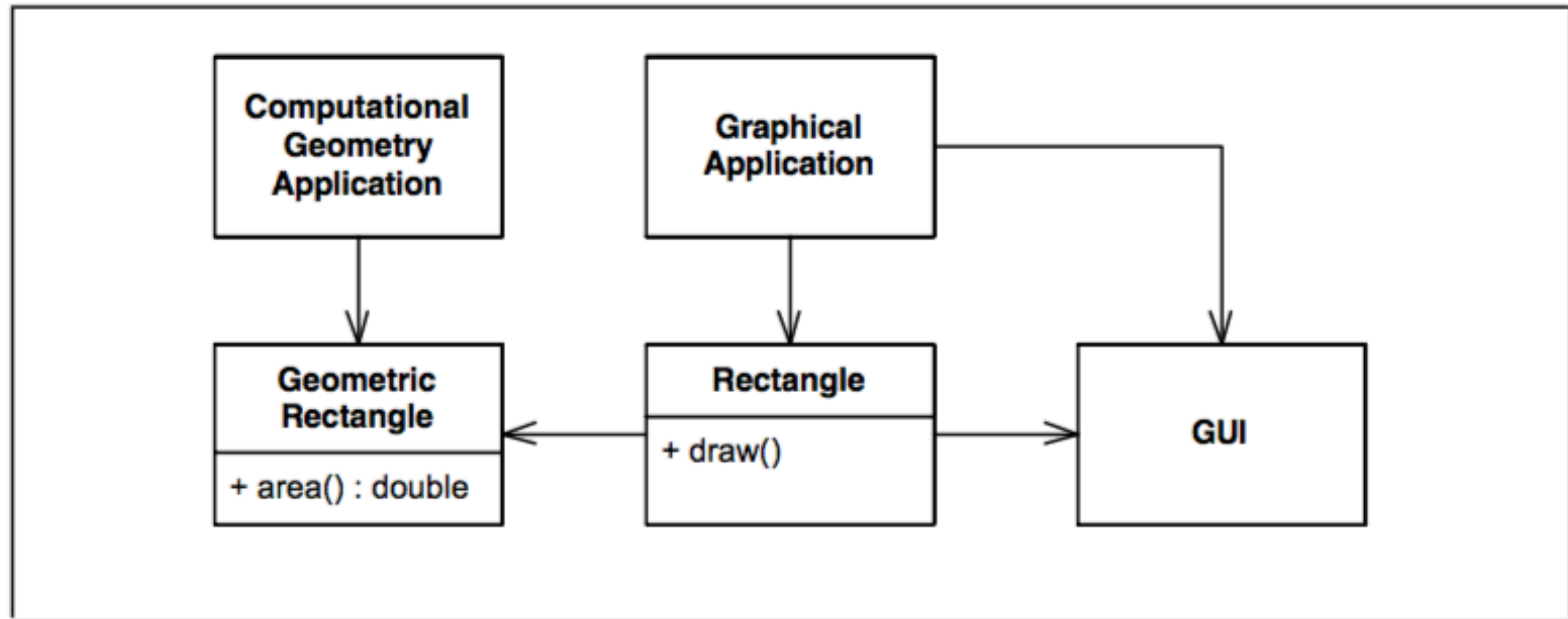


Figure 8-2
Separated Responsibilities

@paulmgower

Stop Writing Legacy Code

@paulmgower

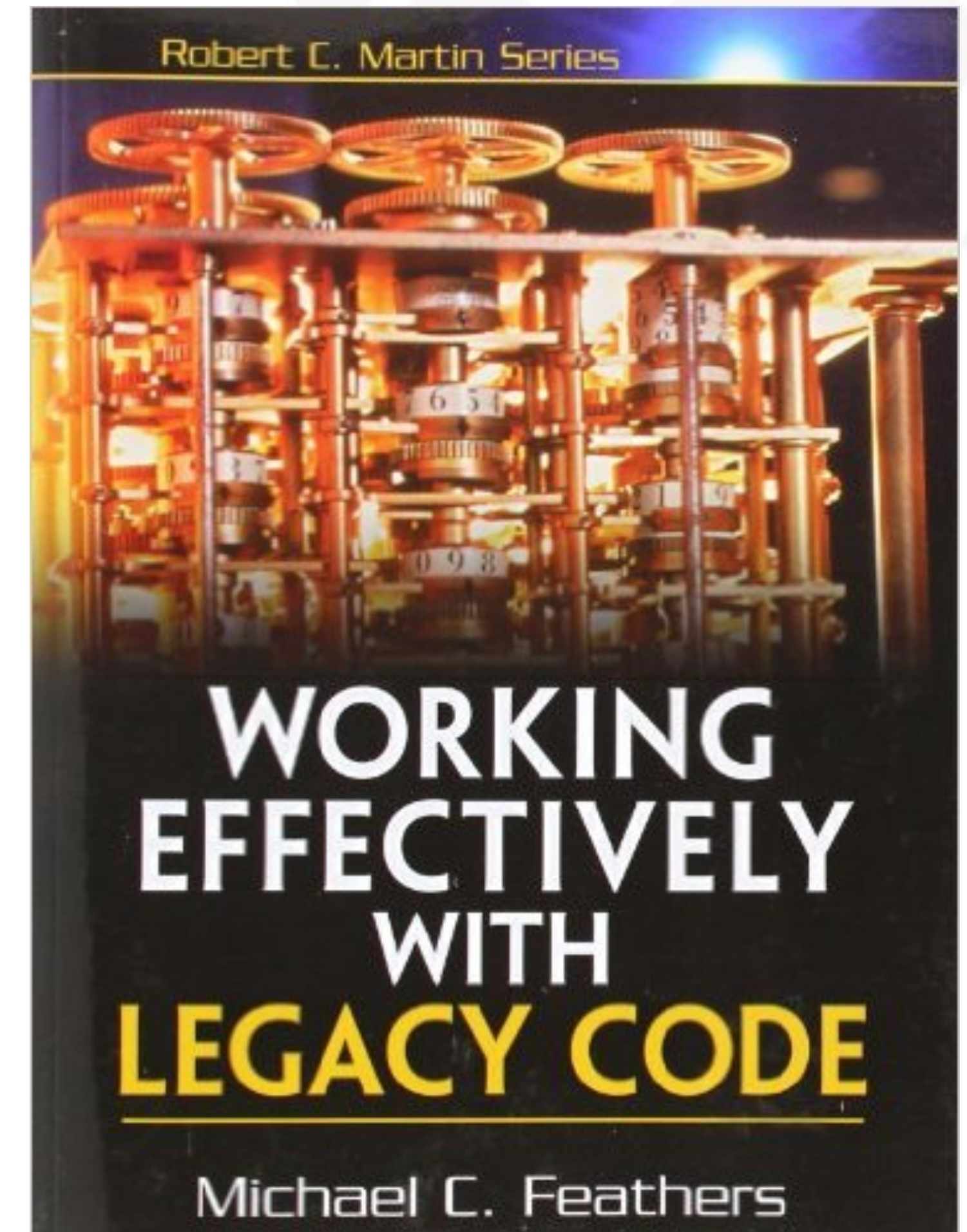
Stop Writing Legacy Code

- Legacy Code == Code w/out Tests

@paulmgower

Stop Writing Legacy Code

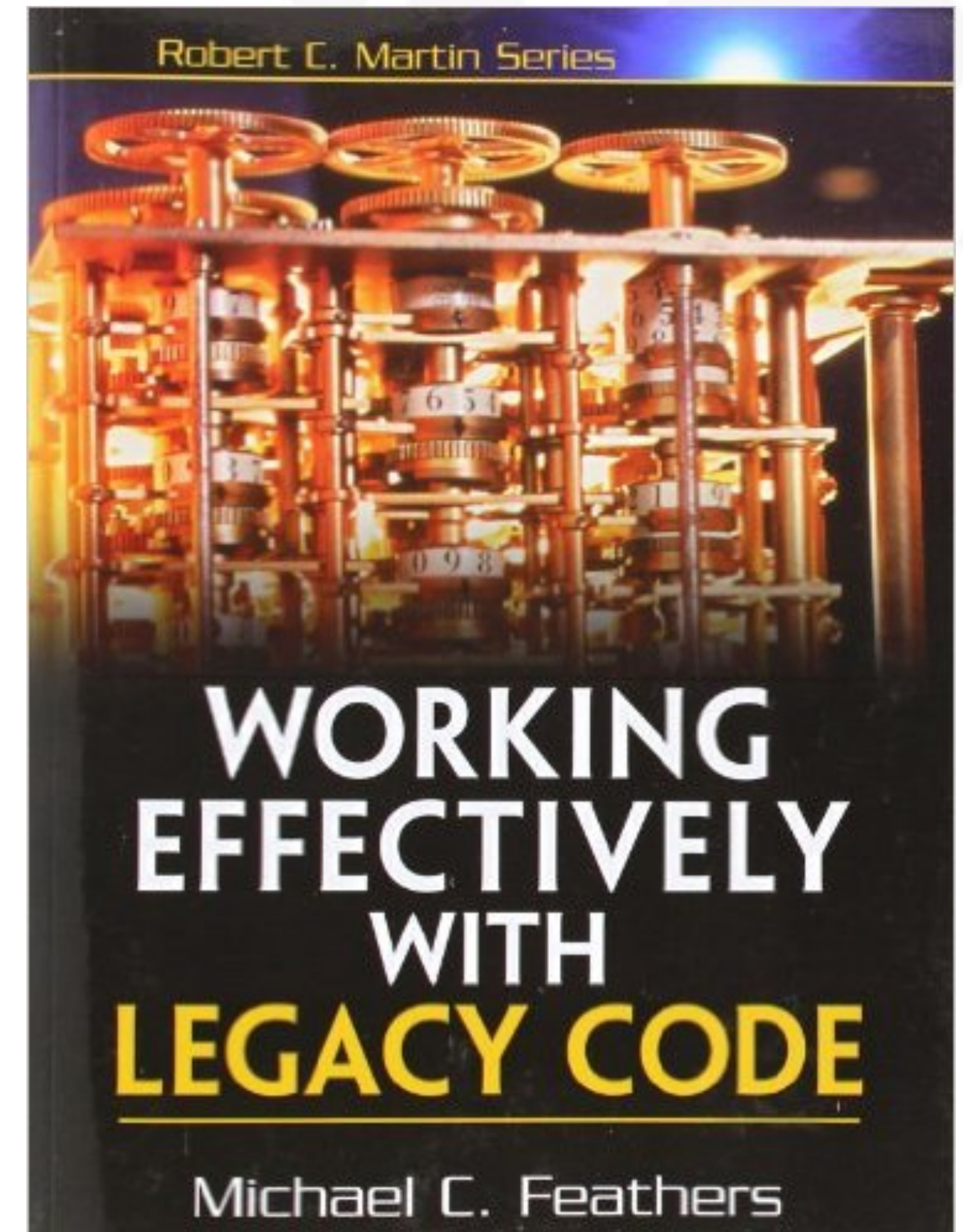
- Legacy Code == Code w/out Tests
- Great Step-by-step guide



@paulmgower

Stop Writing Legacy Code

- Legacy Code == Code w/out Tests
- Great Step-by-step guide
- Write unit tests as you use legacy code



@paulmgower

Code Reviews



@paulmgower

Code Reviews

- Author + 2 Reviewers

@paulmgower



Code Reviews

- Author + 2 Reviewers
- Using “peer pressure”

@paulmgower



Code Reviews

- Author + 2 Reviewers
- Using “peer pressure”
- Encourage collaboration

@paulmgower



Code Review Tips



@paulmgower

Code Review Tips

- Keep a positive attitude

@paulmgower



Code Review Tips

- Keep a positive attitude
- Check your ego

@paulmgower



Code Review Tips

- Keep a positive attitude
- Check your ego
- Review all code

@paulmgower



Code Review Tips

- Keep a positive attitude
- Check your ego
- Review all code
- Code Review early and often

@paulmgower



"Ask programmers to review
10 lines of code, they'll find
10 issues.

Ask them to do **500 lines**
and they'll say **it looks**
good".

The harsh truth about code reviews.

@paulmgower

Final Review

- Readable variable names
- No Magic Numbers
- DRY
- Do one thing
- Unit tests
- Code Reviews

@paulmgower



Questions?

@paulmgower



paul gower

CTO of Aristotle Labs

@paulmgower | aristotlelabs.com

Resources

- The Mythical Man Month by Frederick P. Brooks Jr (<http://www.amazon.com/The-Mythical-Man-Month-Engineering-Anniversary/dp/0201835959>)
- Code Complete by Steve McConnell (<http://www.amazon.com/Code-Complete-Practical-Handbook-Construction/dp/0735619670>)
- Clean Code by Robert C. Martin (<http://www.amazon.com/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882>)
- Working Effectively with Legacy Code by Michael Feathers (<http://www.amazon.com/Working-Effectively-Legacy-Michael-Feathers/dp/0131177052>)
- FogCreek Software Blog
 - Effective Code Reviews - 9 Tips from a Converted Skeptic (<http://blog.fogcreek.com/effective-code-reviews-9-tips-from-a-converted-skeptic/>)
 - Stop More Bugs with our Code Review Checklist (<http://blog.fogcreek.com/increase-defect-detection-with-our-code-review-checklist-example/>)

@paulmgower