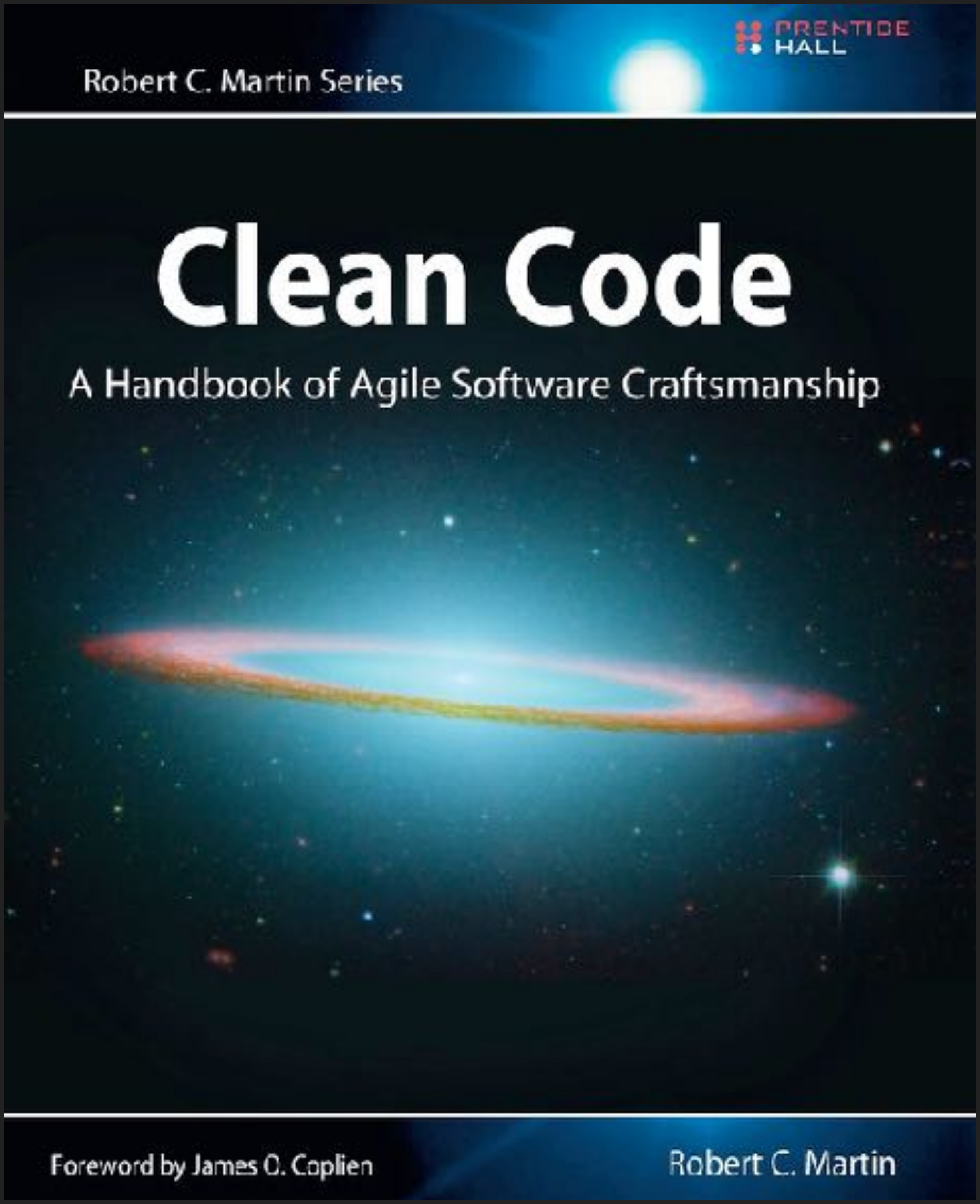
INTRODUCTION TO

REFACTORING

WHAT IS REFACTORING?

- “is the process of restructuring existing computer code without changing its external behaviour.”
- To improve:
  - **readability**
  - **maintainability**
  - **extensibility**
  - reduce **complexity**



Robert C. Martin Series

PRENTICE HALL

Clean Code

A Handbook of Agile Software Craftsmanship

Foreword by James O. Coplien

Robert C. Martin

PYTHON STYLE GUIDE PEP8

- https://www.python.org/dev/peps/pep-0008/
- “code is read much more often than it is written”
- Intended to improve readability
- Worth reading at least once
- Some editors and IDEs understand PEP8

MOTIVATION: WHAT DOES THIS DO?

```
…
pattern = re.compile('.*vmem=(?P<value>[0-9\.]+)(?P<suffix>[GgMmKk]*).*')
match = pattern.match(string)
vmem = float(match.group('value')) * multiplier[match.group('suffix')]
…
```

IS THIS BETTER?

```
…
vmem = extract_metric_from_string(string, metric='vmem')
…
```

STRUCTURE

- English uses sentences, paragraphs, sections, chapters etc. to improve readability
- The key structural element in code is the **function**
- We have full control over the name of the function (somewhat less control over its content)
- Other tools to improve structure are **classes**, **variables**, **comments**

FUNCTIONS

FUNCTIONS

- A function should do **one** thing
- Functions should be **small**. Or even smaller
- Specify argument names when calling a function can improve clarity
- Python has docstrings
- Names: lowercase with underscores: `i_am_a_function()`

EXAMPLE: A FUNCTION

```
def convert_value(value, suffix=''):
    """Convert string value and (optional) SI suffix to float"""
    multiplier = {'':1,
                  'k':1000, 'K':1024,
                  'm':1000**2, 'M':1024**2,
                  'g':1000**3, ‘G’:1024**3}
    return float(value) * multiplier[suffix]
```

VARIABLES

VARIABLES

- Well named variables help readability
- Names: same as functions: `i_am_a_variable`
- Additional variables can also help
- Not much more to say…

EXAMPLE: ADDITIONAL VARIABLE

```
match = re.match(‘.*vmem=(?P<value>[0-9\.]+)(?P<suffix>[GgMmKk]*).*’, string)
```

EXAMPLE: ADDITIONAL VARIABLE

```
pattern = ‘.*vmem=(?P<value>[0-9\.]+)(?P<suffix>[GgMmKk]*).*’
match = re.match(pattern, string)
```

COMMENTS

COMMENTS

- “If you have to use a comment you have failed”
- But, sometimes they can help and docstrings are part of PEP8
- Remove commented-out code

EXAMPLE: USEFUL VS POINTLESS COMMENTS

```
x = x + 1    # Increment x

…

x = x + 1    # Compensate for border
```

ITERATIVE REFINEMENT

INTERATIVE REFINEMENT

- Extract some lines of code into a function
- Name the function well; add a docstring
- Repeat until all functions do one thing

EXAMPLE IN PYCHARM