Single responsibility principle

Good coding practice in real life

Vladimir Alekseichenko

Agenda

- Related concepts
- Single responsibility principle
- Practical examples

Related concepts

Curly's law

Curly Washburn in the 1991 comedy **City Slickers**.

http://www.youtube.com/watch?v=2k1uOqRb0HU

Separation of concerns

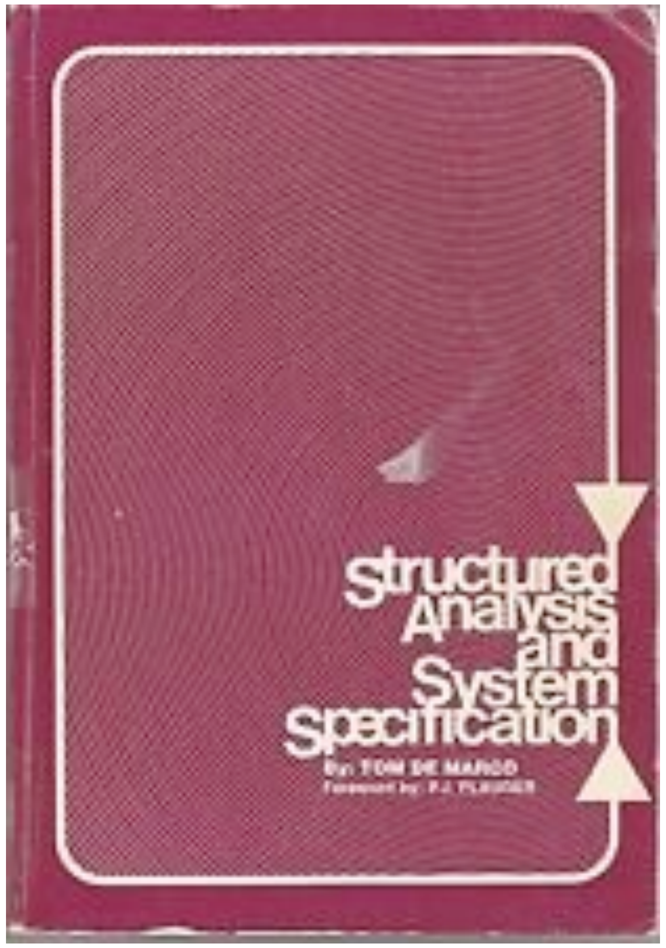
Create distance between dissimilar concepts in your code.

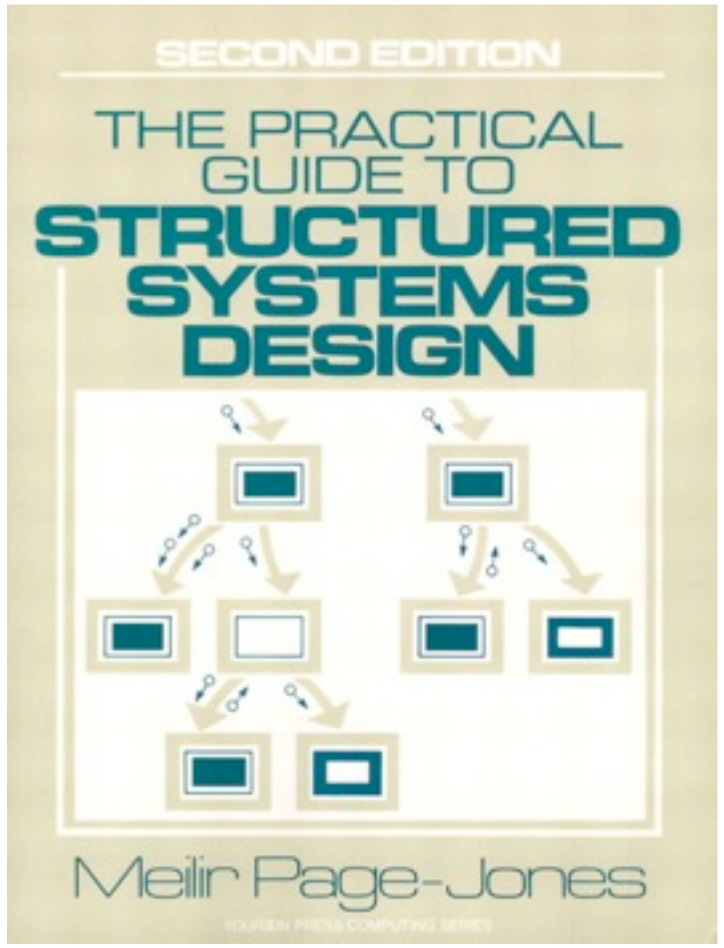
This allows you to change one without affecting the other.

Cohesion

This principle was described in the work of Tom DeMarco and Meilir Page-Jones.

They called it cohesion.


Structured Analysis and System Specification
By: TOM DE MARCO
Foreword by: P.J. PLAUGER


SECOND EDITION
THE PRACTICAL GUIDE TO STRUCTURED SYSTEMS DESIGN
Meilir Page-Jones
YOURDON PRESS COMPUTING SERIES

Low cohesion

http://persbaglio.ilcannocchiale.it/?TAG=disordine

High cohesion

http://www.3deko.info/inter/ofor/415-uyutnyj-balkon-mesto-otdyxa-chast-ii.html

And other related topics

- Don't Repeat Yourself
- Once and Only Once
- Single Point of Truth
- ...

http://www.butunclebob.com/CleanCoder.UncleBob

If you can think of more than one motive for changing a class, then that class has more than one responsibility.

What is a responsibility?

It's a reason for change.

Violate SRP

http://www.comodotnet.com/news/march-2014-meeting-announcement

Single responsibility principle

The class should have one **reason to change**.

Design your classes so they ideally only do one thing and do one thing well.

Unix philosophy

1. Small is beautiful.
2. Make each program do one thing well.
3. ...

http://en.wikipedia.org/wiki/Unix_philosophy

Quiz

It’s a dog or plane?

http://woman-project.com/post/23136

http://photo-fact.ru/index.php/humor/72-sobaka-samolet

Symptoms violation of SRP

- Class has too many comments
- Using too many instance variables
- Passing too many parameters
- Methods have many conditionals
- Hard to write unit test

Benefits of SRP

- Code complexity is reduced
- Readability is greatly improved
- Coupling is generally reduced

Examples

https://www.flickr.com/photos/doug88888

```ruby
class Radio
  def change_station
    #...
  end

  def volume_down
    #...
  end

  def volume_up
    #...
  end
end
```

```ruby
class Radio
  def change_station
    #...
  end

  def volume_down
    #...
  end

  def volume_up
    #...
  end
end
```

first →

second →

```ruby
class RadioStation
  def change_station
    #...
  end
end

class RadioVolume
  def volume_down
    #...
  end

  def volume_up
    #...
  end
end
```

https://www.flickr.com/photos/doug88888

```ruby
class User
  attr_accessor :name

  def valid?
    false == (name.nil? or name.empty?)
  end
end
```

```ruby
class User
  attr_accessor :name

  def initialize(validator)
    @validator = validator
  end

  def valid?
    @validator.valid?(name)
  end
end

class NonEmptyValidator
  def valid?(value)
    false == (value.nil? or value.empty?)
  end
end
```

```ruby
class User
  attr_accessor :name

  def initialize(validator)
    @validator = validator
  end

  def valid?
    @validator.valid?(name)
  end
end

class NonEmptyValidator
  def valid?(value)
    false == (value.nil? or value.empty?)
  end
end
```

```ruby
class LengthValidator
  def initialize(min, max)
    @min = min
    @max = max
  end

  def valid?(value)
    return false if value.length < @min
    return false if value.length > @max
    true
  end
end
```

```ruby
class Hasher
  def hash
    content = File.read("/tmp/my_file")
    Digest::MD5.hexdigest(content)
  end
end
```

Responsibilities

```ruby
class Hasher
  def hash
    content = File.read("/tmp/my_file")
    Digest::MD5.hexdigest(content)
  end
end
```

```ruby
class Hasher
  def initialize(algorithm = Digest::MD5)
    @algorithm = algorithm
  end

  def hash(value)
    @algorithm.hexdigest(value)
  end
end
```

```ruby
content = File.read("/tmp/my_file")
Hasher.new.hash(content)
Hasher.new(Digest::SHA256).hash(content)
```

Summarize

- Do one thing
- Do that thing only
- Do that thing well

Resources

- **The Single Responsibility Principle** by *Robert C. Martin.*
- **Clean Code: A Handbook of Agile Software Craftsmanship** by *Robert C. Martin.*
- **Head First Software Development** by *DanPilone, Russ Miles.*
- **Head First Object-Oriented Analysis and Design** by *Brett D. McLaughlin, Gary Pollice, David West.*