

# Working With Legacy Code

@MaciejOczko



# Plan

- What is legacy code?
- How to approach it?
- Example



# What is legacy code?

Code is legacy code as soon as it's written.

*Michael Feathers, Working Effectively with Legacy Code*



# What is legacy code?

- Code without tests (Michael Feathers)
- Inherited code
- Poorly designed code
- Too complicated code
- Illegible code

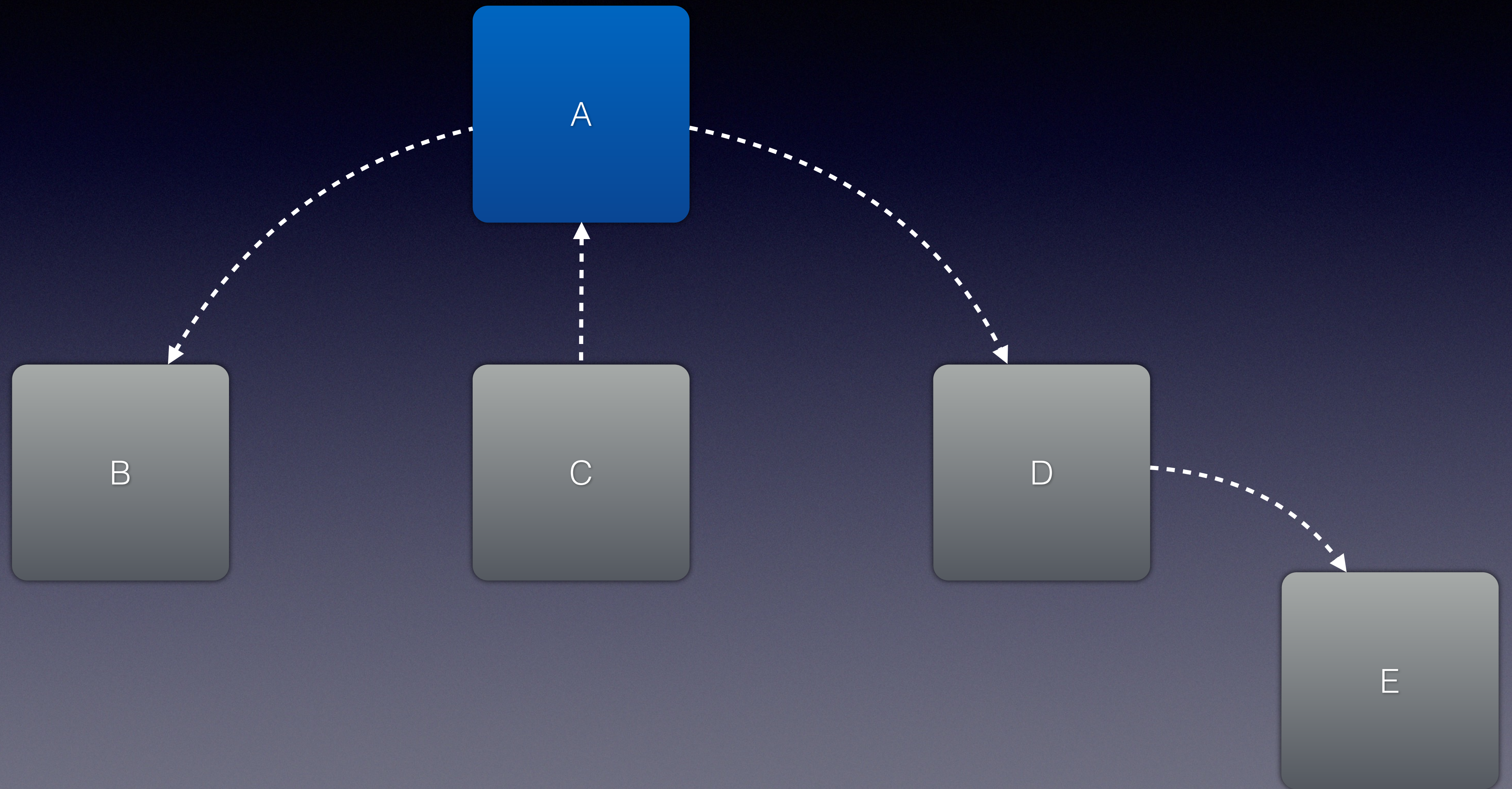


# How to approach it?

- Identify change points
- Find an inflection point
- Cover the inflection point
- Make changes
- Refactor the covered code



# Inflection point





# Inflection point

- Violation of Single Responsibility Principle
- Violation of Open-Closed Principle



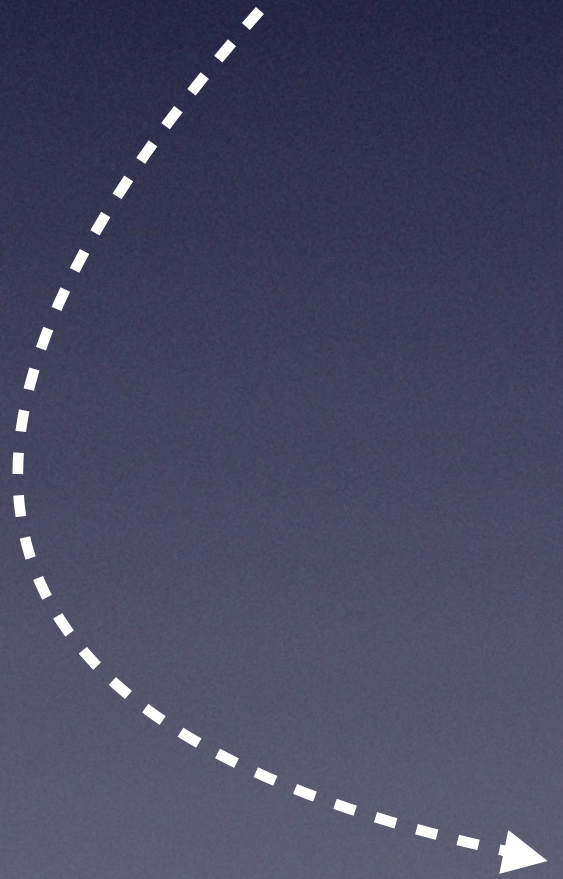
# Covering inflection point

- Break external dependencies
- Break internal dependencies
- Write tests



# Breaking **external** dependencies

```
- (NSUInteger)calculatePrice {  
    ...  
    StockAnalyzer *analyzer = [[StockAnalyzer alloc] init];  
    ...  
}
```



```
- (id)initWithAnalyzer:(id <AnalyzerProtocol>)analyzer {  
    self = [super init];  
    if (self) {  
        _analyzer = analyzer;  
    }  
    return self;  
}
```



# Breaking **internal** dependencies

```
- (CGSize)calculateSize {  
    ...  
    UIScreen *screen = [UIScreen mainScreen];  
    ...  
}
```



```
- (CGSize)calculateSize {  
    ...  
    UIScreen *screen = [self screen];  
    ...  
}  
  
- (UIScreen *)screen {  
    return [UIScreen mainScreen];  
}
```



Write tests



Make changes



# When to Refactor?

- Too wide class responsibility
- Duplicated code
- Not readable code
- Too many method arguments (more than 2)
- Method has more than 20 lines
- Class has more than 200 lines



# Keep in mind

- Composition over inheritance
- Avoid singletons
- Isolate dependencies
- Inject dependencies



Example



Thanks!

@maciejoczko