Compile time Guaranteed

Nikita Lutsenko

@nlutsenko

Facebook, Parse

Build & Run

- » Compile

Build & Run

- ✅ Compile
- Link

Build & Run

- ✅ Compile
- ✅ Link
- Run

Build & Run

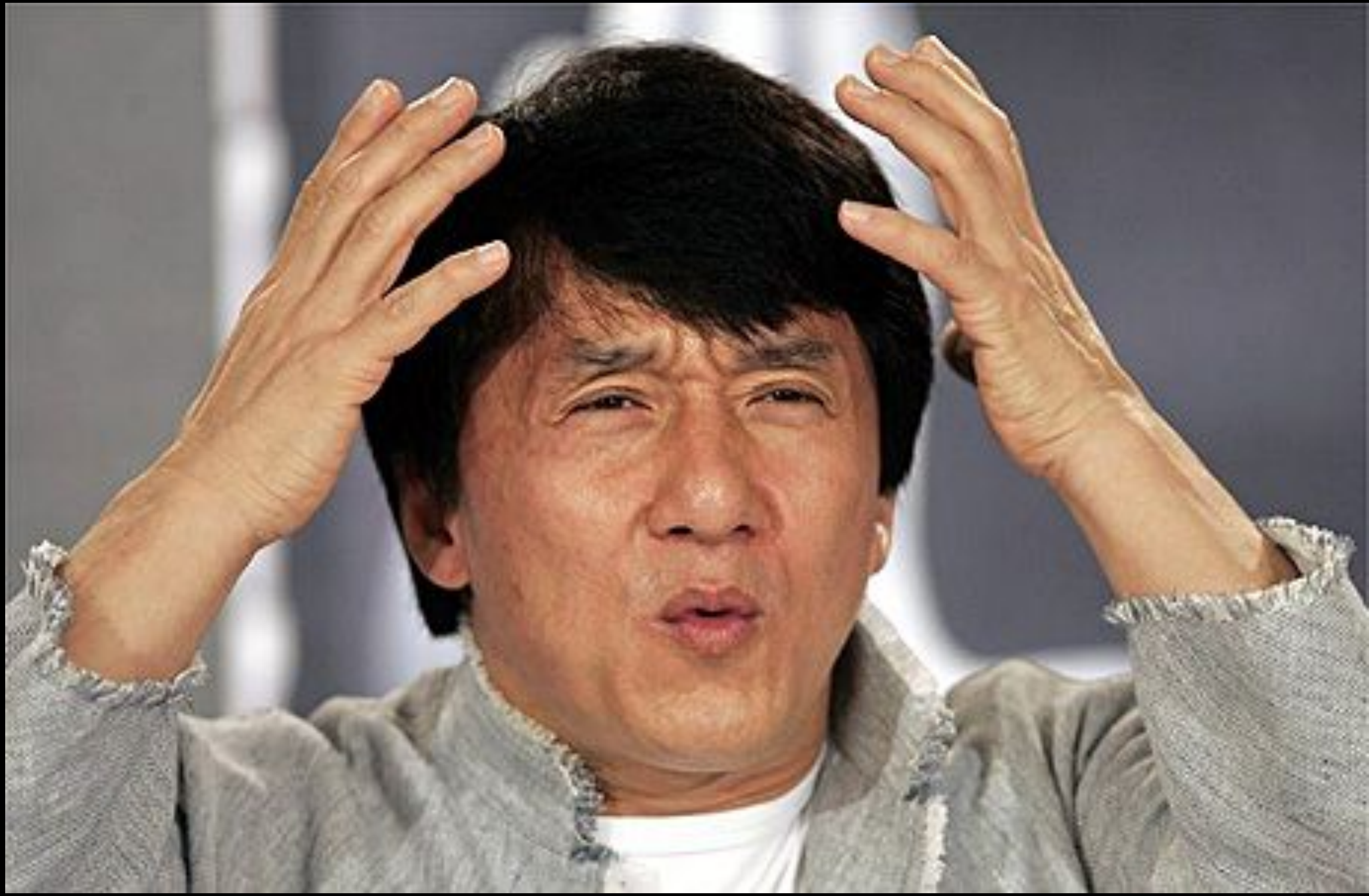
- ✅ Compile
- ✅ Link
- ✅ Run

Build & Run

- ✅ Compile
- ✅ Link
- ✅ Run
- ❌ Crash

Build & Run

- ✅ Compile
- ✅ Link
- ✅ Run
- ❌ Crash



We are Human!

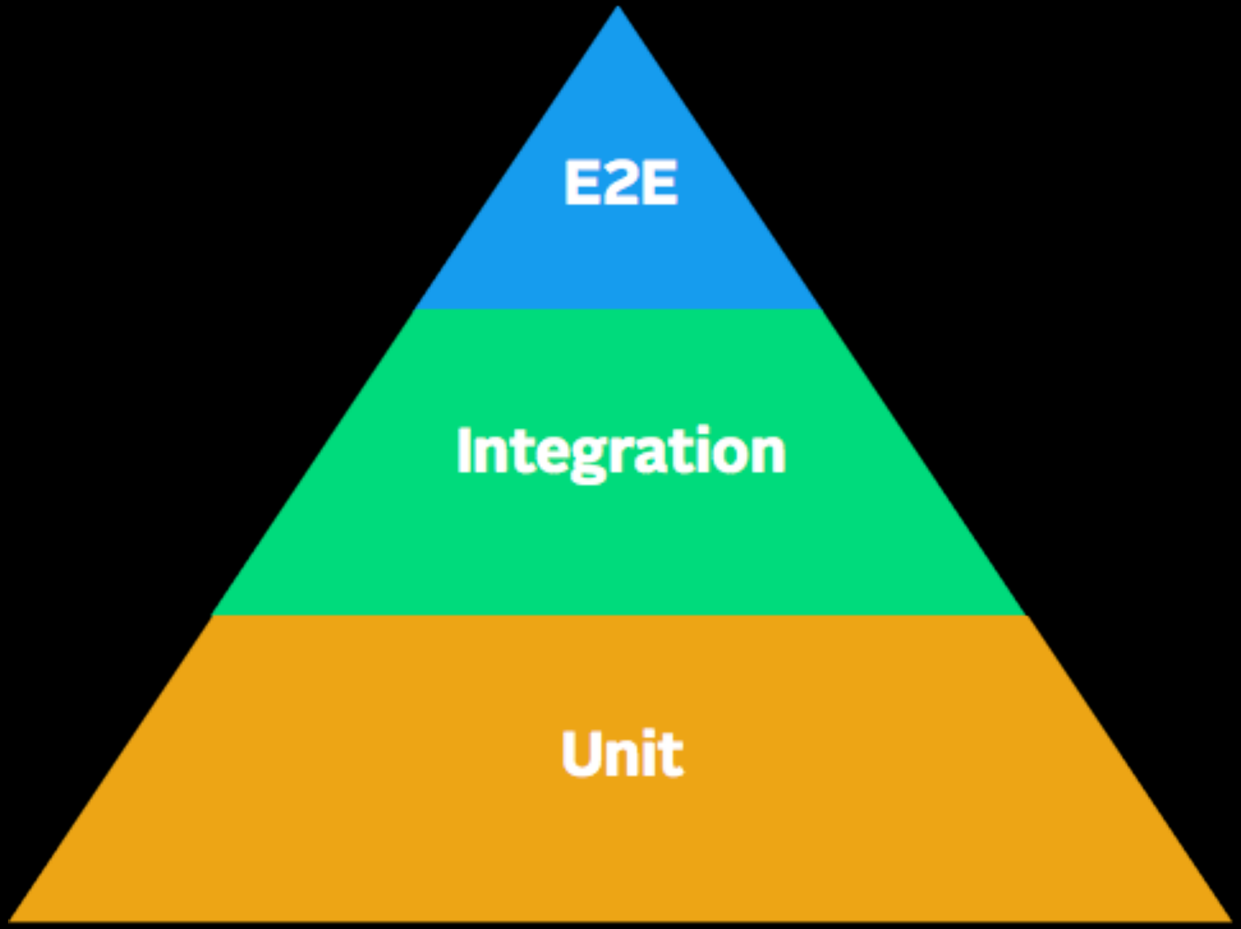
My brain tells me the truth, and it can't find any errors, therefore I have written perfect software.
— Zed A. Shaw

Solution?

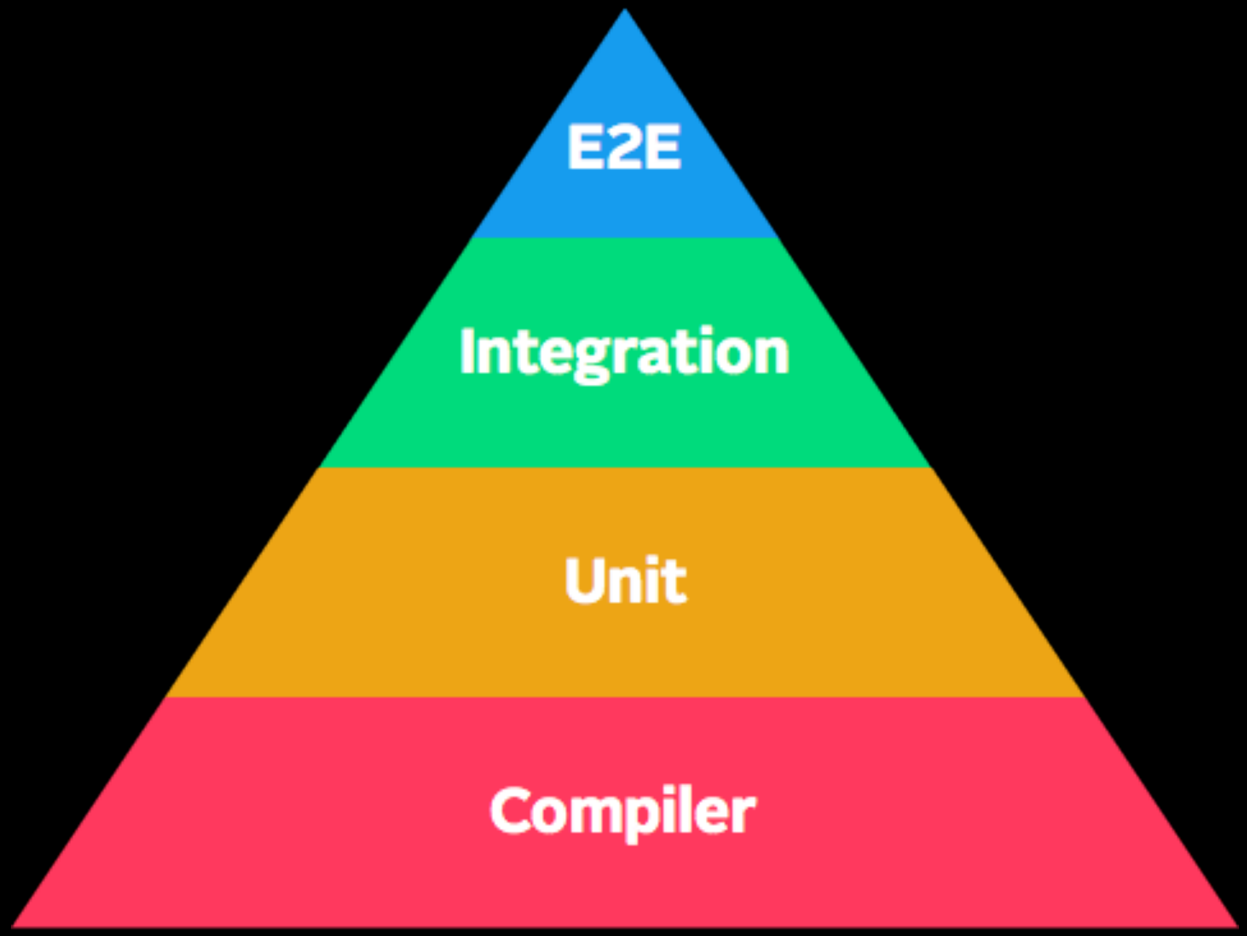
Test all the things!

- Unit
- Integration
- E2E
- Monkey
- Manual
- Dogfooding 🐶
- ...

Pyramid of Testing



Pyramid of Testing (Extended)



Compilers are not human!

- Syntax
- Types
- Logic
- Variables
- Access Control
- Semantic Analysis
- …

Compiler Guarantees

»

Compiler Guarantees

- Compiler Errors
  - Syntax
  - Intent

Compiler Guarantees

- Compiler Errors
  - Syntax
  - Intent
- Compiler Warnings
  - Potential Problems
  - Misuse or Wrong Use of API

Compiler Driven Development

»

Compiler Driven Development

- Fail Early, Fail Often

Compiler Driven Development

- Fail Early, Fail Often
    - **throws** and **Optional()** are your best friends!

Compiler Driven Development

- Fail Early, Fail Often
  - **throws** and **Optional()** are your best friends!
- Never Trust Input

Compiler Driven Development

- Fail Early, Fail Often
  - **throws** and **Optional()** are your best friends!
- Never Trust Input
  - ! vs ?

Compiler Driven Development

- Fail Early, Fail Often
  - **throws** and **Optional()** are your best friends!
- Never Trust Input
  - ! vs ?
- Protect Everything

Compiler Driven Development

- Fail Early, Fail Often
  - **throws** and **Optional()** are your best friends!
- Never Trust Input
  - `!` vs `?`
- Protect Everything
  - `guard`, `if`, `switch`, `where`

Compiler Driven Development

- Constants not Variables

Compiler Driven Development

- Constants not Variables
  - ~~var~~ **let**

Compiler Driven Development

- Constants not Variables
  - ~~var~~ **let**
- Be Explicit!

Compiler Driven Development

- Constants not Variables
  - ~~var~~ **let**
- Be Explicit!
  - `final`, `private`, `public`

Compiler Driven Development

- Constants not Variables
    - ~~var~~ **let**
- Be Explicit!
    - `final`, `private`, `public`
- Prevention Over Documentation
    - Prevent unsupported behaviour

```
class YarrRequest {


}
```

```swift
class YarrRequest {
  init(urlString: String) {
    let request = NSMutableURLRequest(URL: NSURL(string: urlString)!)
    request.addValue("captain", forHTTPHeaderField: "Authorization")
  }
}
```

```swift
class YarrRequest {
    init(urlString: String) {
        let request = NSMutableURLRequest(URL: NSURL(string: urlString)!)
        request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() {
        // Perform `request`...
    }
}
```

```swift
class YarrRequest {
    var request = NSMutableURLRequest()

    init(urlString: String) {
        request.URL = NSURL(string: urlString)!
        request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() {
        // Perform `request`...
    }
}
```

```swift
class YarrRequest {
    var request = NSMutableURLRequest()

    init(urlString: String) {
        request.URL = NSURL(string: urlString)!
        request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() { ... }
}

let request = YarrRequest(urlString: "http://yarr.com")
request.perform()
```

```swift
class YarrRequest {
    var request = NSMutableURLRequest()

    init(urlString: String) {
        request.URL = NSURL(string: urlString)!
        request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() { ... }
}

let request = YarrRequest(urlString: "Yolo™")
request.perform()
```

```swift
class YarrRequest {
    var request = NSMutableURLRequest()

    init(urlString: String) {
        request.URL = NSURL(string: urlString)!
        request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() { ... }
}

let request = YarrRequest(urlString: "Yolo™") // ❌ Crash!
request.perform()
```

```swift
class YarrRequest {
    var request = NSMutableURLRequest()

    init(urlString: String) {
        request.URL = NSURL(string: urlString)!
        request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() { ... }
}

let request = YarrRequest(urlString: "Yolo™") // ❌ Crash! (🐥?)
request.perform()
```

```
class YarrRequest {
    var request = NSMutableURLRequest()

    init(urlString: String) {
        request.URL = NSURL(string: urlString)!
        // Implicitly unwrapped optional!    ^^^
        request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() { ... }
}

let request = YarrRequest(urlString: "Yolo™") // ❌ Crash! (🐥!)
request.perform()
```

```swift
class YarrRequest {
    var request = NSMutableURLRequest()

    init(urlString: String) throws {
        guard let url = NSURL(string: urlString) else { throw NSError() }
        request.URL = url
        request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() { ... }
}

do {
  let request = try YarrRequest(urlString: "Yolo™")
  request.perform()
} catch { ... }
```

```
class YarrRequest {
    var request = NSMutableURLRequest()

    init(urlString: String) throws {
        guard let url = NSURL(string: urlString) else { throw NSError() }
        request.URL = url
        request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() { ... }
}

do {
  let request = try YarrRequest(urlString: "http://yarr.com")
  request.request = NSMutableURLRequest(URL: NSURL()) // Different request!
  request.perform()
} catch { ... }
```

```swift
class YarrRequest {
    var request = NSMutableURLRequest()

    init(urlString: String) throws {
        guard let url = NSURL(string: urlString) else { throw NSError() }
        request.URL = url
        request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() { ... }
}

do {
  let request = try YarrRequest(urlString: "http://yarr.com")
  request.request = NSMutableURLRequest(URL: NSURL()) // Different request!
  request.perform() // 🐥?
} catch { ... }
```

```swift
class YarrRequest {
    let request = NSMutableURLRequest()
//  ^^ `let` not `var`

    init(urlString: String) throws {
         guard let url = NSURL(string: urlString) else { throw NSError() }
         request.URL = url
         request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() { ... }
}

do {
  let request = try YarrRequest(urlString: "http://yarr.com")
  request.request = NSMutableURLRequest(URL: NSURL()) // ❌ Compiler Error
  request.perform() // 🐥!
} catch { ... }
```

```swift
class YarrRequest {
    let request = NSMutableURLRequest()

    init(urlString: String) throws {
         guard let url = NSURL(string: urlString) else { throw NSError() }
         request.URL = url
         request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() { ... }
}

do {
  let request = try YarrRequest(urlString: "http://yarr.com")
  request.request.URL = NSURL() // Different URL
  request.perform()
} catch { ... }
```

```swift
class YarrRequest {
    let request = NSMutableURLRequest()

    init(urlString: String) throws {
        guard let url = NSURL(string: urlString) else { throw NSError() }
        request.URL = url
        request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() { ... }
}

do {
  let request = try YarrRequest(urlString: "http://yarr.com")
  request.request.URL = NSURL() // Different URL!
  request.perform() // 🐥?
} catch { ... }
```

```swift
// YarrRequest.swift
class YarrRequest {
    private let request = NSMutableURLRequest()

    init(urlString: String) throws {
        guard let url = NSURL(string: urlString) else { throw NSError() }
        request.URL = url
        request.addValue("captain", forHTTPHeaderField: "Authorization")
    }

    func perform() { ... }
}

// Main.swift
do {
  let request = try YarrRequest(urlString: "http://yarr.com")
  request.request.URL = NSURL() // ❌ Compiler Error
  request.perform() // 🐥!
} catch { ... }
```

Compiler Driven Development

- Be Explicit!
- Never Trust Input
- Protect Everything
- Fail Early, Fail Often
- Constants not Variables
- Prevention Over Documentation

There is more!

- Analyzer Warnings
  - `xcodebuild test`
  - `xcodebuild analyze`

There is more!

- Analyzer Warnings
  - `xcodebuild test`
  - `xcodebuild analyze`
- Compiler Warnings
  - Do not ignore warnings!

There is more!

- Analyzer Warnings
  - `xcodebuild test`
  - `xcodebuild analyze`
- Compiler Warnings
  - Do not ignore warnings!
  - tr.im/goshdarnclangwarnings by @mattt

There is more!

- Analyzer Warnings
  - `xcodebuild test`
  - `xcodebuild analyze`
- Compiler Warnings
  - Do not ignore warnings!
  - tr.im/goshdarnclangwarnings by @mattt
  - 26 Lexer, 21 Parser, 234 Semantic Warnings

There is more in ObjC!

- Target -> Build Settings -> Other Warning Flags

There is more in ObjC!

- Target -> Build Settings -> Other Warning Flags
  - **-Weverything**

There is more in ObjC!

- Target -> Build Settings -> Other Warning Flags
  - **-Weverything -Wpedantic**

There is more in ObjC!

- Target -> Build Settings -> Other Warning Flags
  - **-Weverything -Wpedantic**
- Disable if necessary:

```
#pragma clang diagnostic push
#pragma clang diagnostic ignored "-W<FLAG>"
// ...
#pragma clang diagnostic pop
```

Questions?

github.com/nlutsenko/compile-time-guaranteed