



Telerik

TeamPulse



@Team_Pulse



gplus.to/teampulse



facebook.com/telerik

DEALING WITH CODE FROM THE PIT OF DESPAIR

Philip Japikse (@skimedic)

phil@telerik.com

www.skimedic.com/blog

MVP, MCSD.Net, MCDBA, CSM, CSP

Agile Practices Evangelist, Telerik

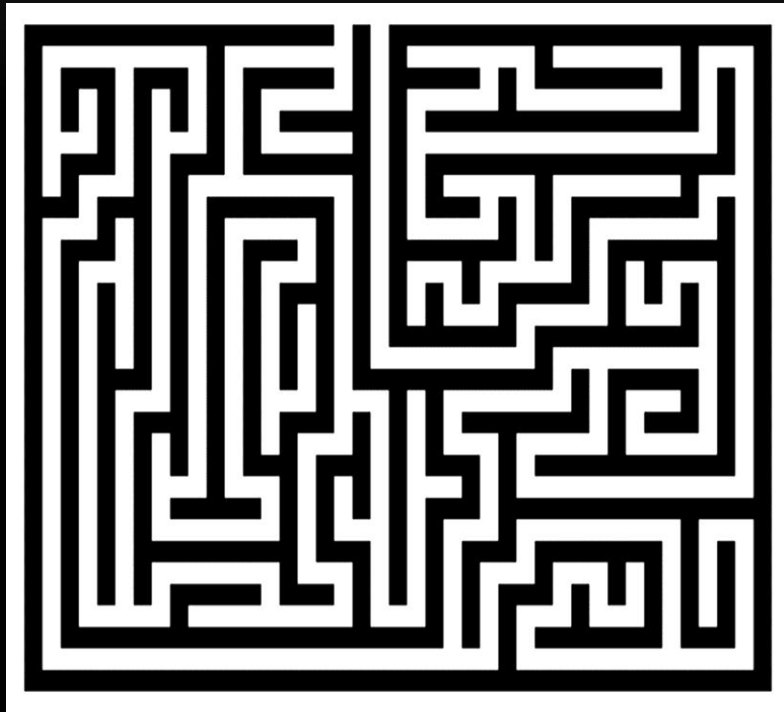


PHIL.TOSTRING()

- Agile Practices Evangelist, Telerik, Inc.
- Microsoft MVP, MCSD, MCDBA, CSM, CSP
- Lead Director, Cincinnati .NET User's Group
- Founder, Agile Conferences, Inc.
 - www.dayofagile.org
- Host, Zero To Agile podcast
 - www.telerik.com/zerotoagile



LEGACY CODE



- Any Code Not Under Test – Martin Fowler
- High Technical Debt
- Icky



WHEN TO CHANGE SOFTWARE



- Killing Bugs
- Adding Features
- Optimizing Resources
- Removing FUD



MAKE ONE CHANGE AT A TIME



- Fix or Add
- Test after every change
- Commit after every test



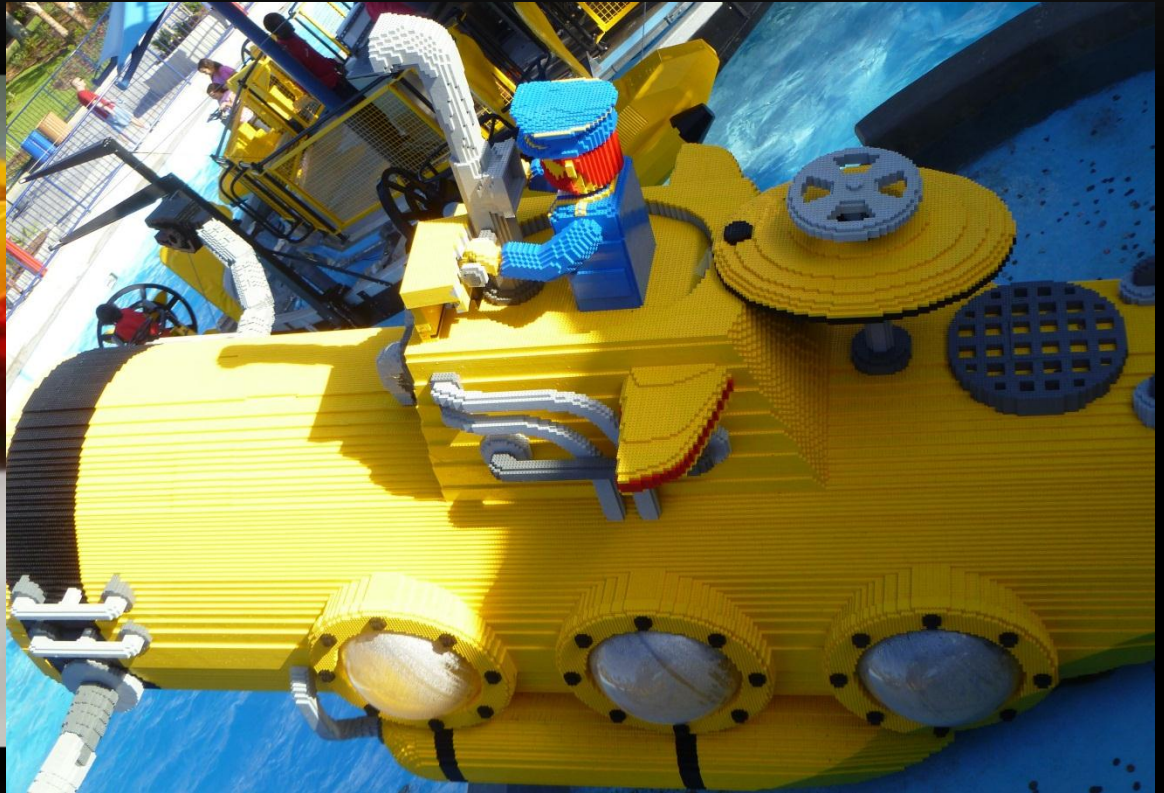
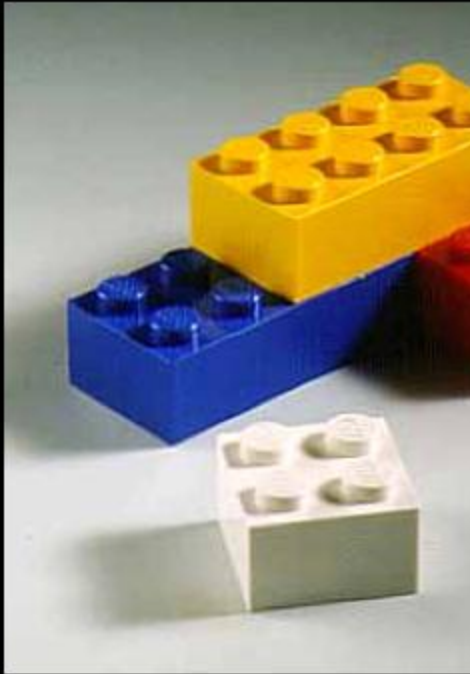
TOOLS OF THE TRADE



- TDD
- BDD
- Mocking
- Dependency Inversion
- UI Testing



TEST DRIVEN DEVELOPMENT



BEHAVIOR DRIVEN DEVELOPMENT



<http://www.1000steine.com/brickset/images/10179-1.jpg>



www.telerik.com

Plan. Build. Test.

FAKES, STUBS, AND MOCKS



- Fakes have working implementations
 - Hardcoded, not suitable for production
- Stubs provide canned answers
- Mocks pre-programmed with expectations
 - Create a specification
 - Record behavior



DEPENDENCY INVERSION



USER INTERFACE TESTING



- Typically Manual
- Often tested by customers
- Tools becoming better





DEMO

NUnit – TDD Framework

MSpec – BDD Framework

JustMock – Mocking Framework

Test Studio – User Interface Testing Framework



TESTING AND REFACTORING

SAFETY FIRST!



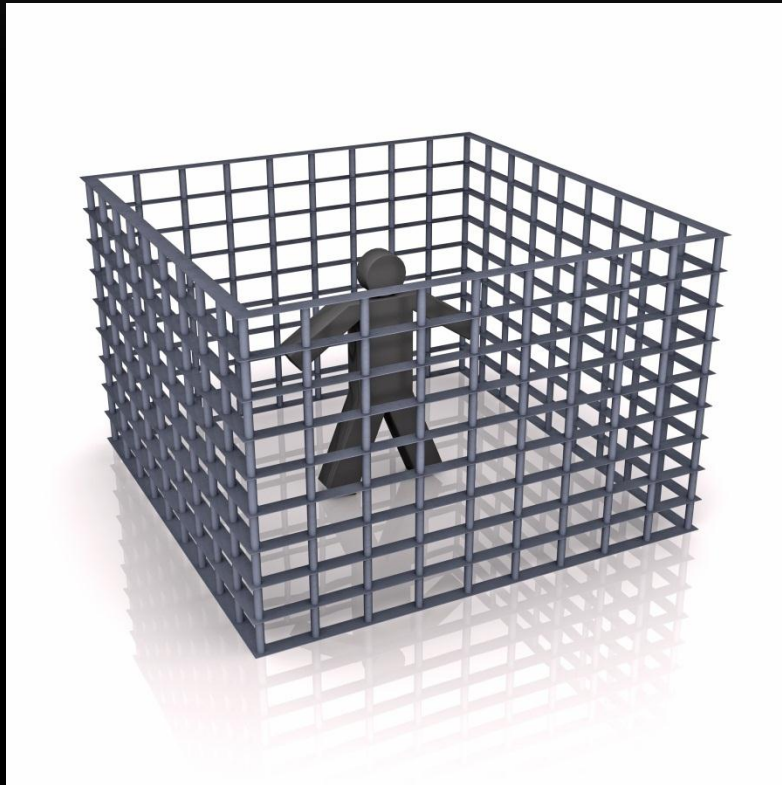
- Remember:
Only you can prevent bugs!



“INTEGRATION” TEST



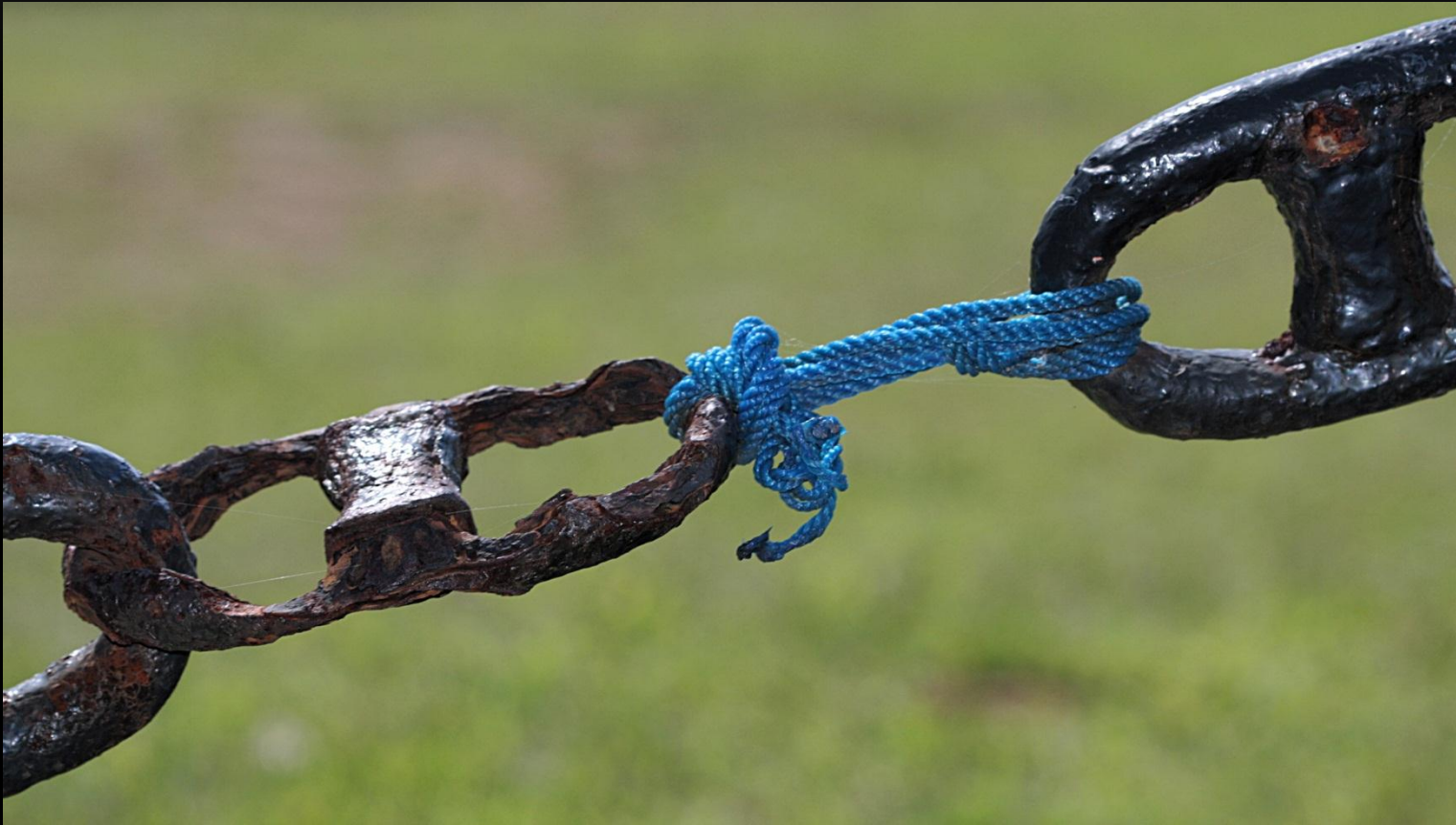
CODE ISOLATION AND REFACTORING



```
10: FindChangePoint()  
20: CreateASeam()  
30: AddNewTest()  
40: Refactor()  
50: If !Isolated GOTO 20  
60: FixTargetIssue()  
70: HappyHour()
```



FIND A CHANGE POINT



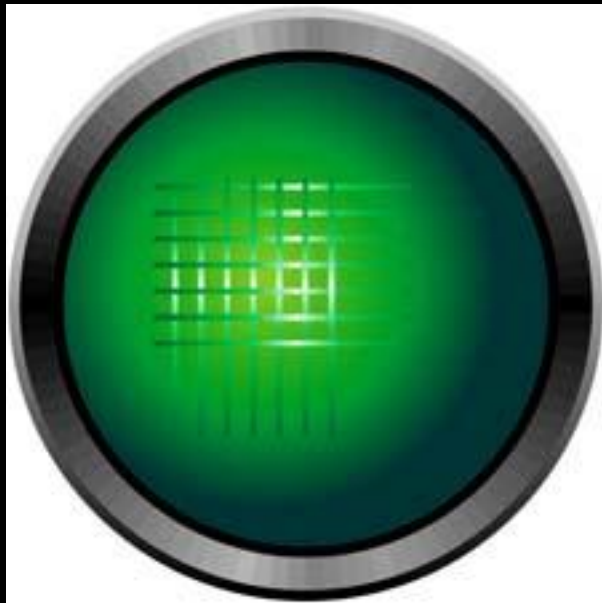
CREATE A SEAM

- Dependency Injection
- Commercial Mocking



ADD A TEST

- TED Style



REFACTOR



- Extract Methods
- Reduce Complexity
- DRY
- SOLID



LATHER, RINSE, REPEAT



- Continue to drill down



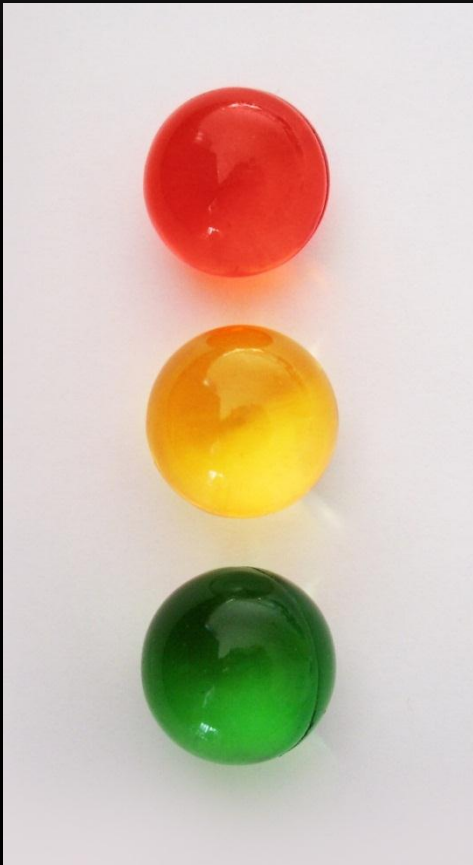
ENTER THE DRAGON



- Code is isolated
- Tests are in place
- Switch to T/BDD



TEST/BEHAVIOR DRIVEN DEVELOPMENT



- Write a failing test
- Make the test pass
- Refactor
 - Remove magic code
 - Add use cases



HAPPY HOUR



- Wave to your coworkers as they struggle with their legacy code...



RESOURCES – UNIT TESTING

- Unit Testing Frameworks
 - NUnit – <http://www.nunit.org>
 - NUnit.Should – <http://www.tinyurl.com/nunit.should>
- Behavior Driven Development Frameworks
 - MSpec – <http://tinyurl.com/mspec>
 - <http://github.com/machine/machine.specifications>
- Mocking Frameworks
 - JustMock Free Edition – <http://telerik.com/mocking>
 - <http://www.telerik.com/products/mocking.aspx>



RESOURCES – TESTING USER INTERFACES

- Unit Testing Frameworks
 - MbUnit/Gallio
 - <http://www.gallio.org>
- Test Studio Framework
 - <http://www.telerik.com/community/download-free-products.aspx>



CONTACT ME

- phil@telerik.com
- www.skimedic.com/blog
- www.twitter.com/skimedic
- www.telerik.com/zerotoagile





Active Refactoring

