



codemanship

# Continuous Inspection

Jason Gorman

@jasongorman



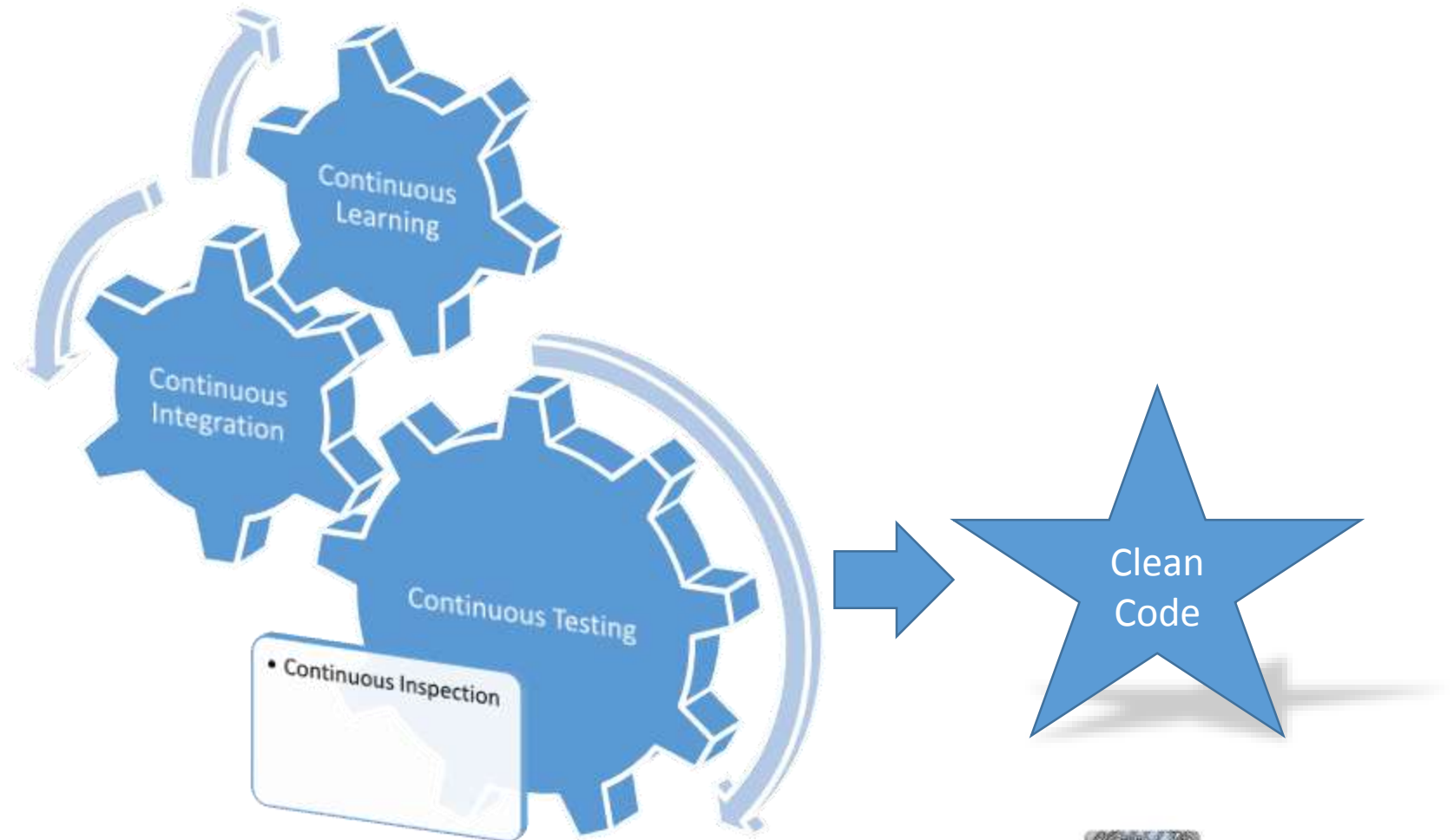
codemanship

# About Me

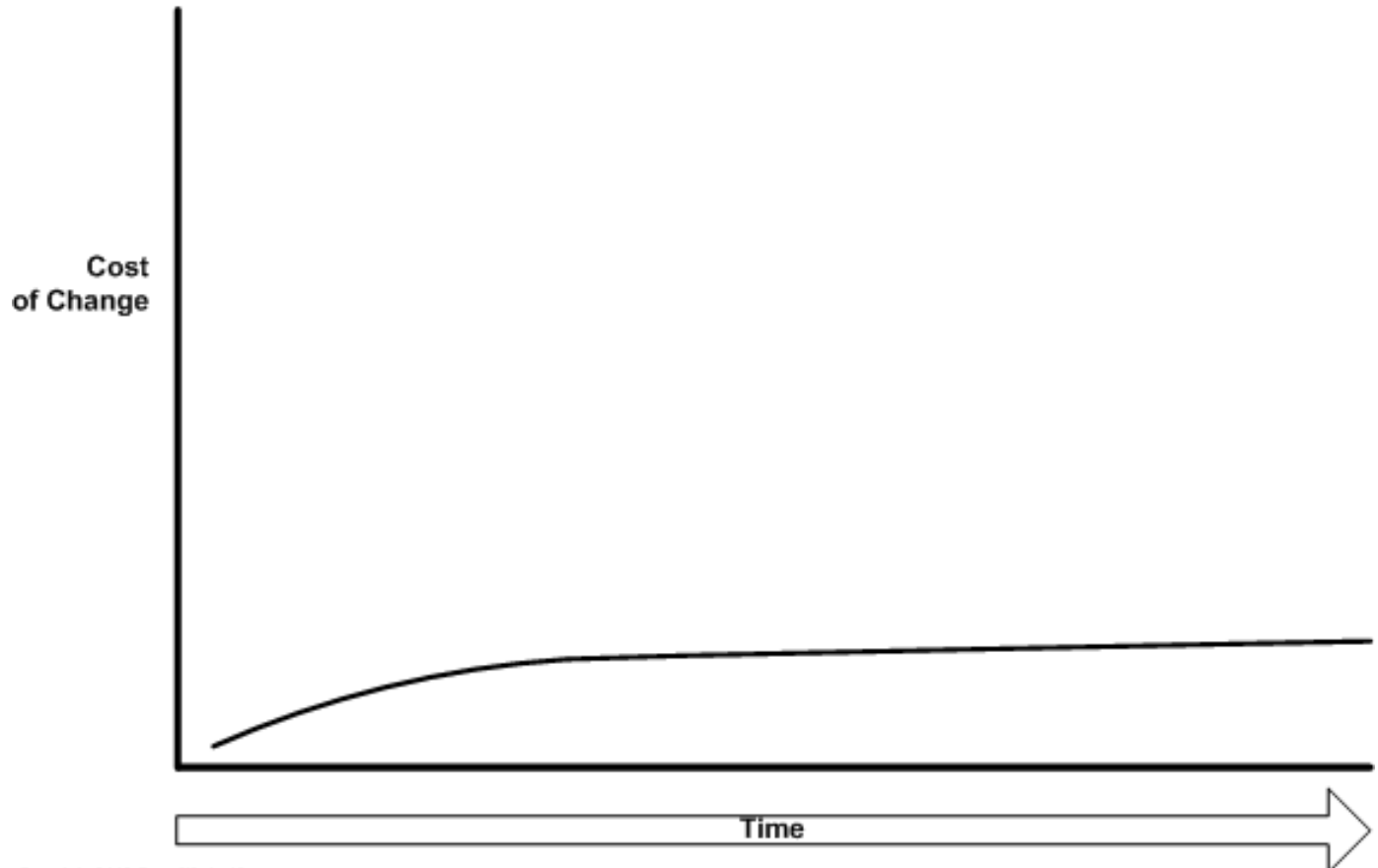
- Training & coaching in Test-driven Development, Refactoring, OO Design Principles & Advanced Techniques for developers
- Programming since 1982
- Founder of the original Software Craftsmanship conference
- Apprentice Will Price studying Comp Sci at Uni of Bristol
- Patron of Bletchley Park Trust
- Available for Weddings and Bar Mitzvahs



# The 5 C's of Continuous Delivery



# Sustainable Pace



Copyright 2002 Scott W. Ambler

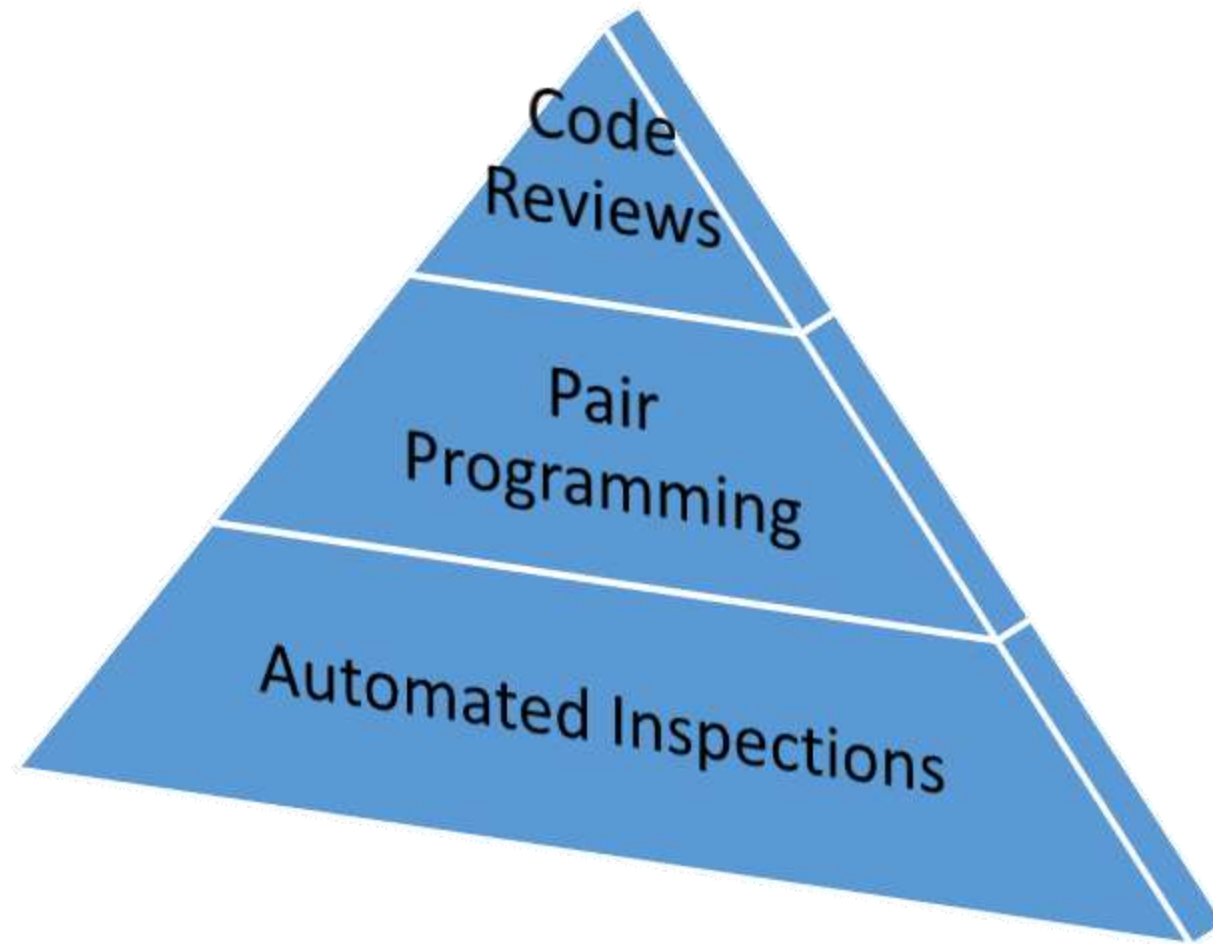
# What is Continuous Inspection?



Very frequent inspection of software to provide *early warning* of maintainability and other code quality issues



# 3 Approaches to Code Inspections



# Code Reviews

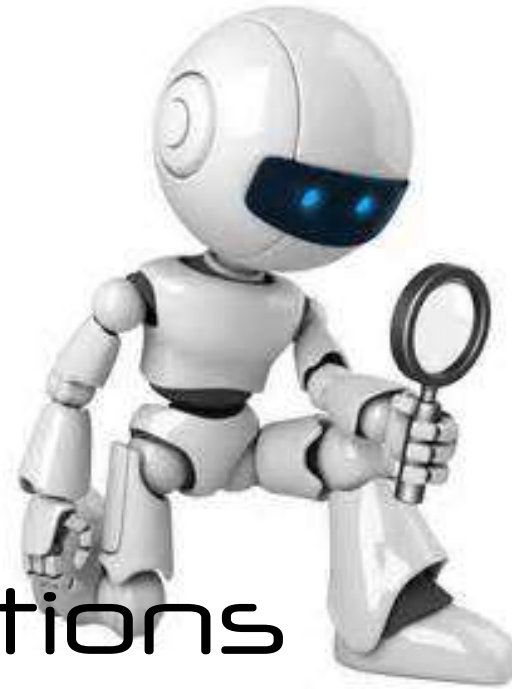




# Pair Programming







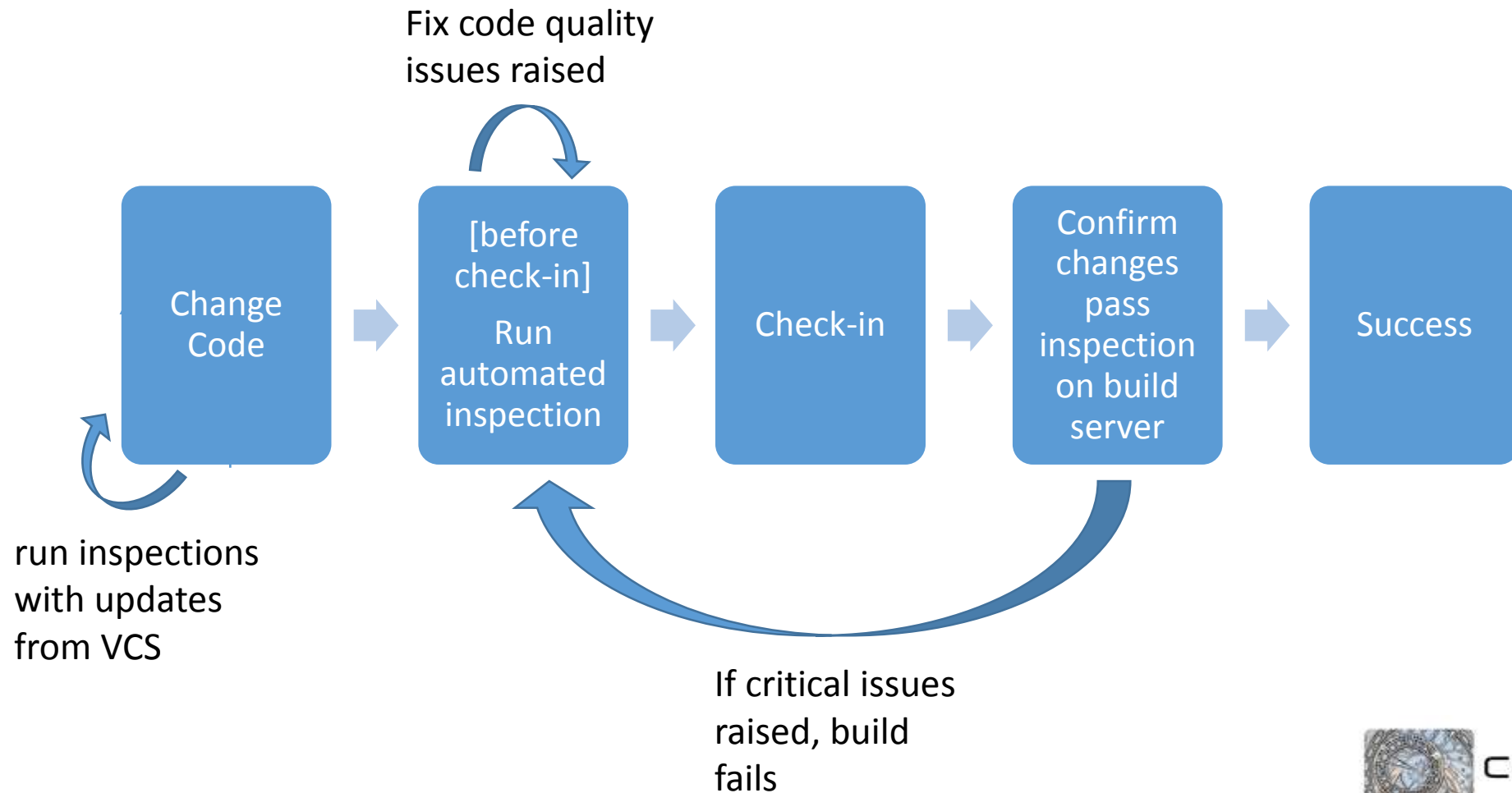
# Automated Inspections

...because an extensive code review every 10-15 minutes is robot's work

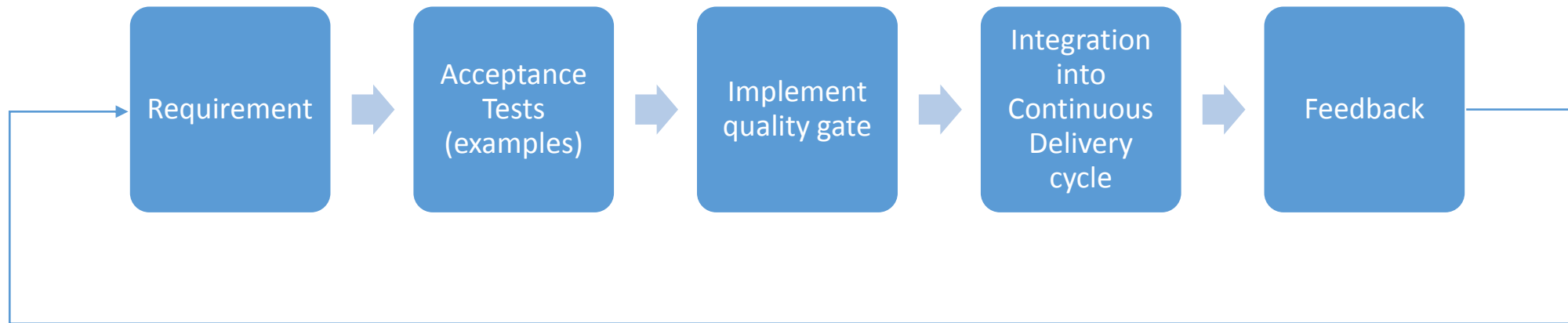


codemanship

# Example Automated Continuous Inspection Process



# The Continuous Inspection Automation Process



# Quality Gates



## Soft Gate

- Quality issues raised do not break the build, but may prompt further review by peers



## Hard Gate

- Breaks the build
- Can be overridden after team review



# Example Code Quality Requirement

## Feature Envy

As a developer, I need to know about any methods that have more than one dependency on features of another class so I can move them to where they belong and reduce coupling



# Example Examples

```
public string Summary {  
    get  
    {  
        return name + ", " + address.House + " " + address.Street + ", "  
            + address.City + ", " + address.Postcode;  
    }  
}
```

This is feature envy

```
public void RentFor(Customer customer) {  
    if(isUnderAge(customer))  
        throw new CustomerUnderageException();  
    customer.AddRental(this);  
}
```

This is NOT feature envy

```
public string Summary  
{  
    get  
    {  
        return this.House + " " + this.Street + ", " +  
            this.City + ", " + this.Postcode;  
    }  
}
```

This is NOT feature envy



# Choose Your Weapon

CodeIt.Right

ndepend

FxCop



- Cecil
- Gendarme

DevExpress™

sonarqube

.NET Reflector

StyleCop

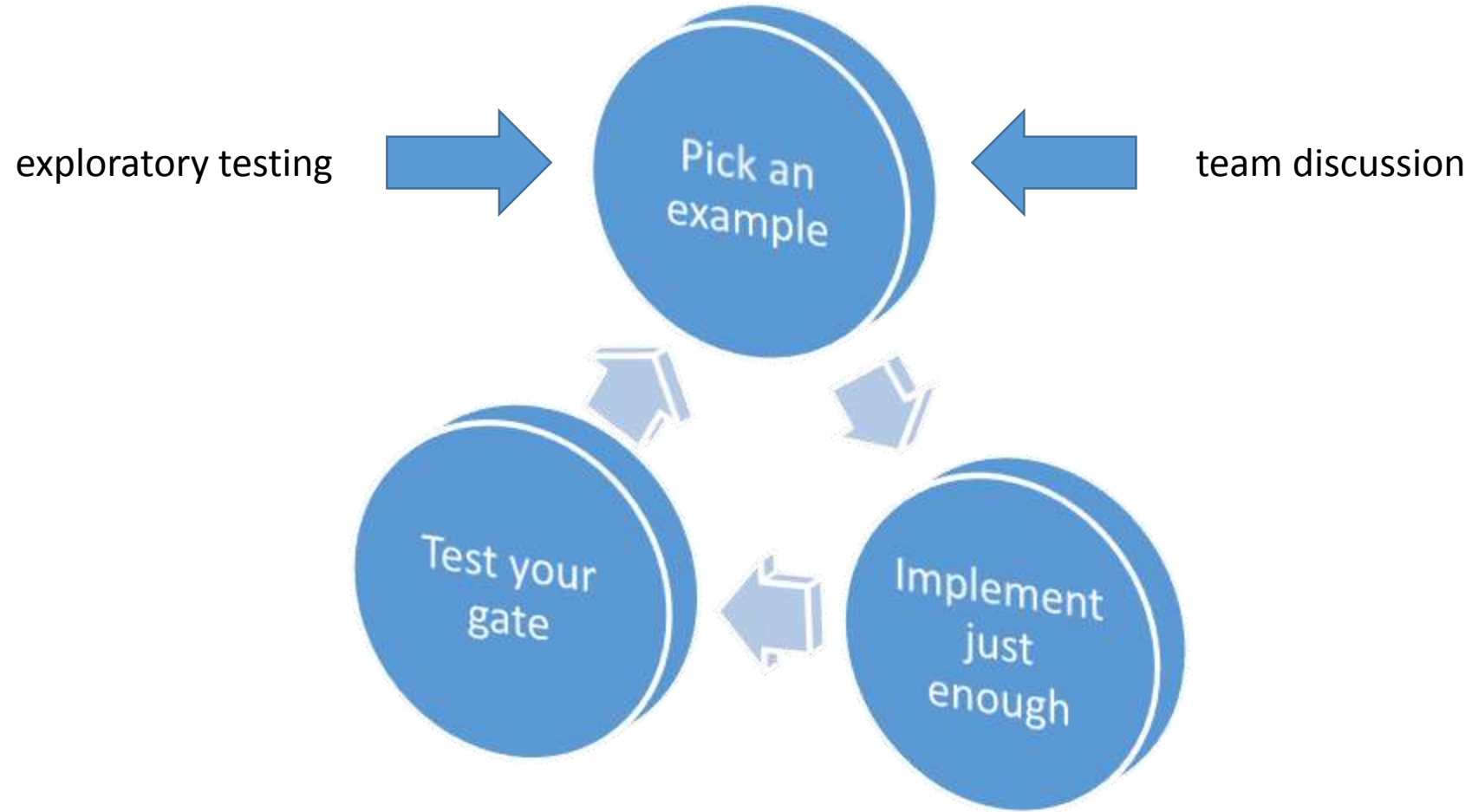
R#

ANTLR

codemanship



# Implementing Your Quality Gate



# Examples We Didn't Think Of

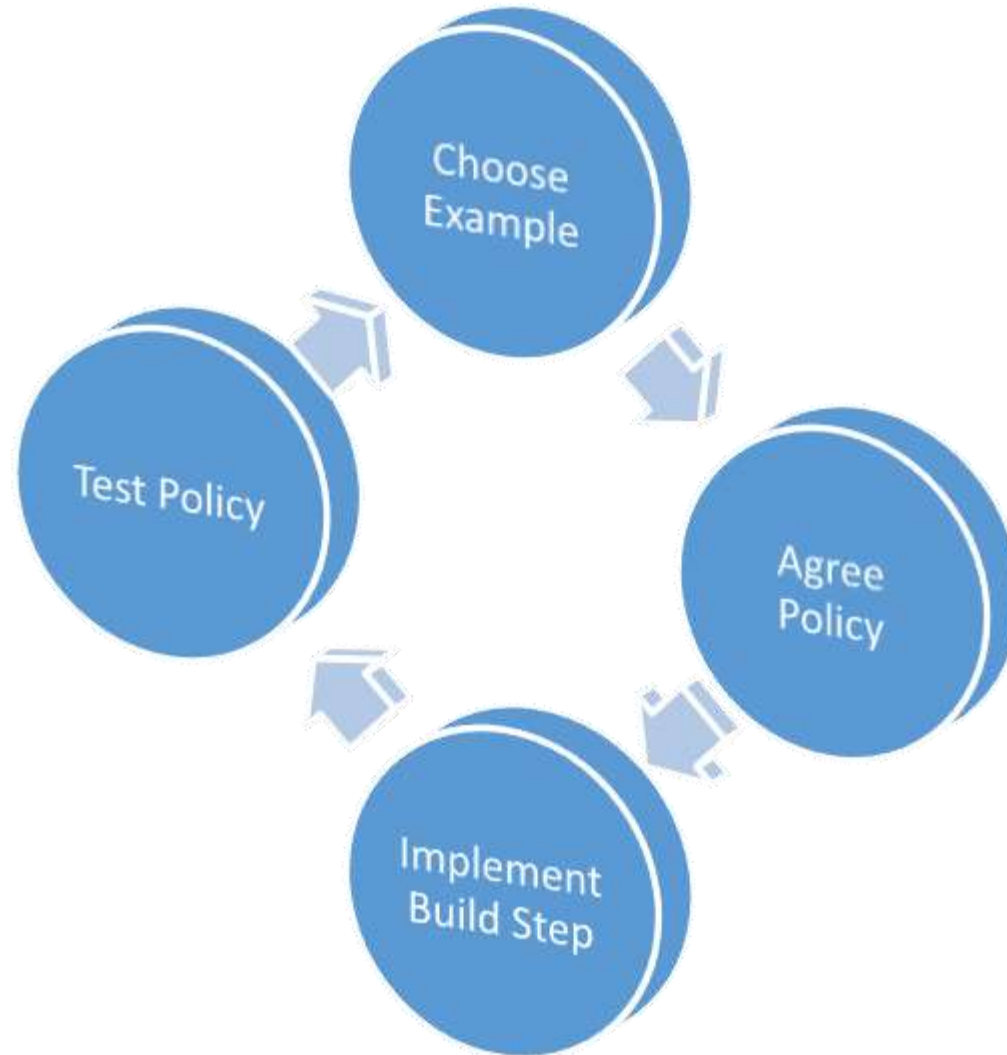
```
public string IsThisFeatureEnvy()  
{  
    Video video = new Video("X Men", Rating.TWELVE);  
    return video.Title;  
}
```



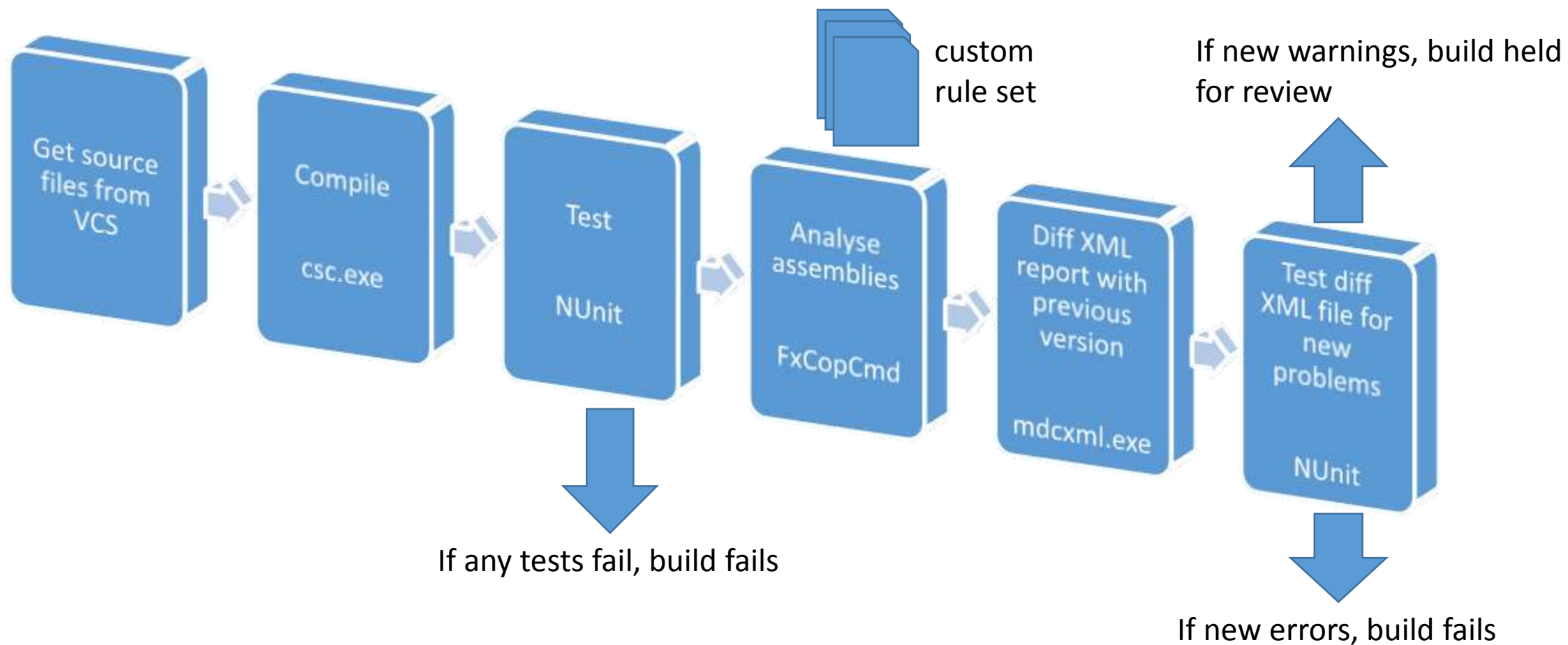
# When Are We Done?



# Integrating Your Quality Gate



# Example FxCop Build Pipeline



# Continuous Inspection Patterns

# Our House, Our Rules



Beware of reusing someone else's ruleset

Involve all developers



codemanship



# Code Analysis Code Is Code

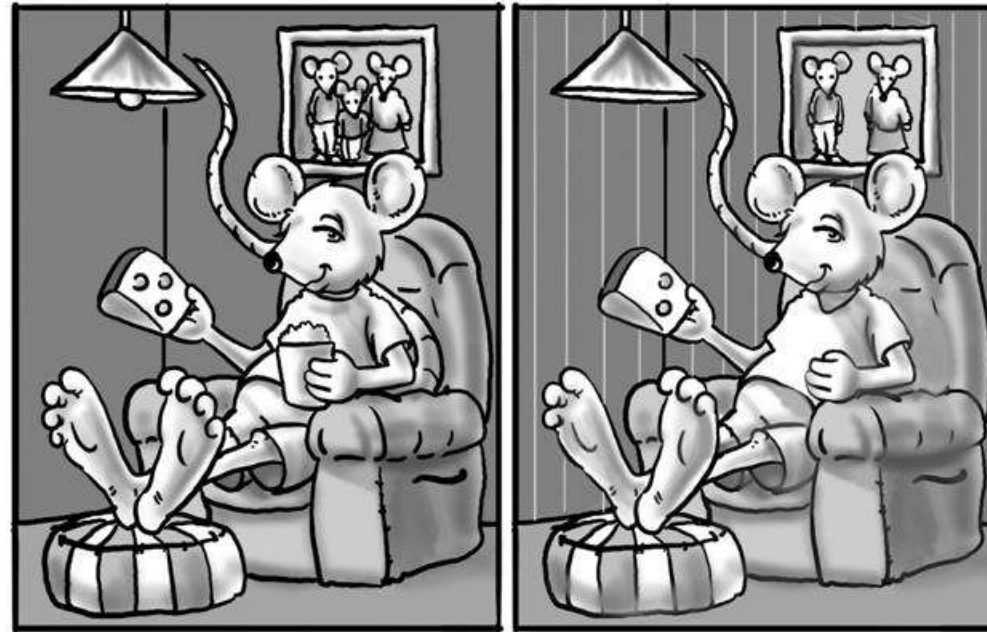


Apply the same care to the analysis code and Continuous Inspection “plumbing” that you apply to production code



codemanship

# Differential Inspections



Apply Continuous Inspection  
to new/changed code



codemanship

# Clean Code Check-in



Run code rules against changes before committing them



# Call To Action



Report problems that  
developers know how to fix



codemanship

# Rising Tide



On legacy code, fail builds if quality gets worse. Raise the bar if quality improves.



codemanship



# Adopt By Stealth



Be ready to run before making it visible



codemanship

[www.codemanship.com](http://www.codemanship.com)