Unit Testing – the first step to Continuous Delivery

promapp™

SoT Master Class

Why unit testing is important

Shaun Field

Ruskin Dantra

WIFI Password:
PromappStepsItUpANotch2018!

promapp™

Before we start…

- Install Visual Studio Code 1.5
- Install .Net Core 2.1.3 SDK
- Visual Studio Code Extensions
    - C# for Visual Studio Code
    - .NET Core Test Explorer
    - Rust Test Lens

Agenda

1. A bit about Promapp
2. Why testing is important
3. How to be kick arse at writing unit tests
4. Exercises
5. Wrap

A BIT ABOUT PROMAPP

Who is Nintex Promapp

- Promapp is the process platform teams love to use.
- Build, improve and share process knowledge from a central online repository.
- Recently acquired by Nintex, the world's leader in intelligent process automation.
- Promapp is the process platform of choice for hundreds of organizations around the world.

nintex

CCA
COCA-COLA AMATIL

TOYOTA
Let's Go Places

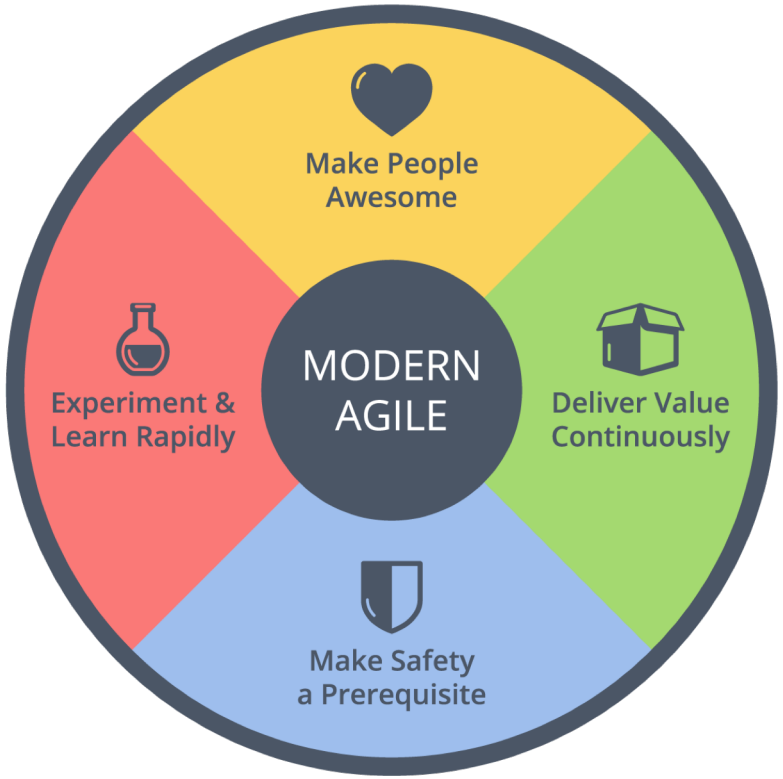
Fisher&Paykel

AIR NEW ZEALAND

WHY TESTING IS IMPORTANT

So why continuous delivery?

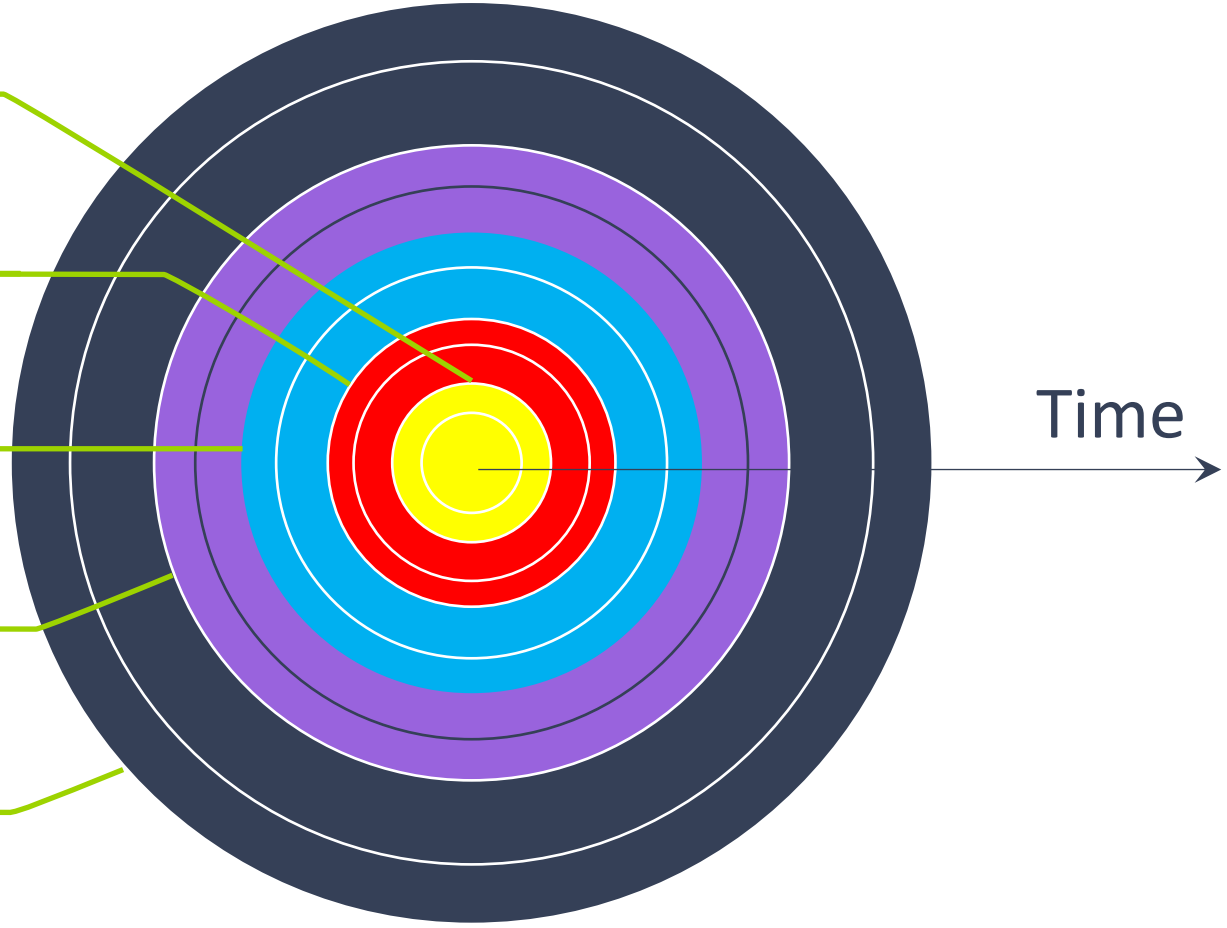
- We want to continually provide value to our customers.
- Routine removes risks.
- Deploying/releasing should become a non event.
- It is a practise that supports agile software development.



WHY TESTING IS IMPORTANT

Feedback builds confidence

- Code
- Continuous Integration
- Peer Review
- Quality Assurance
- Customer

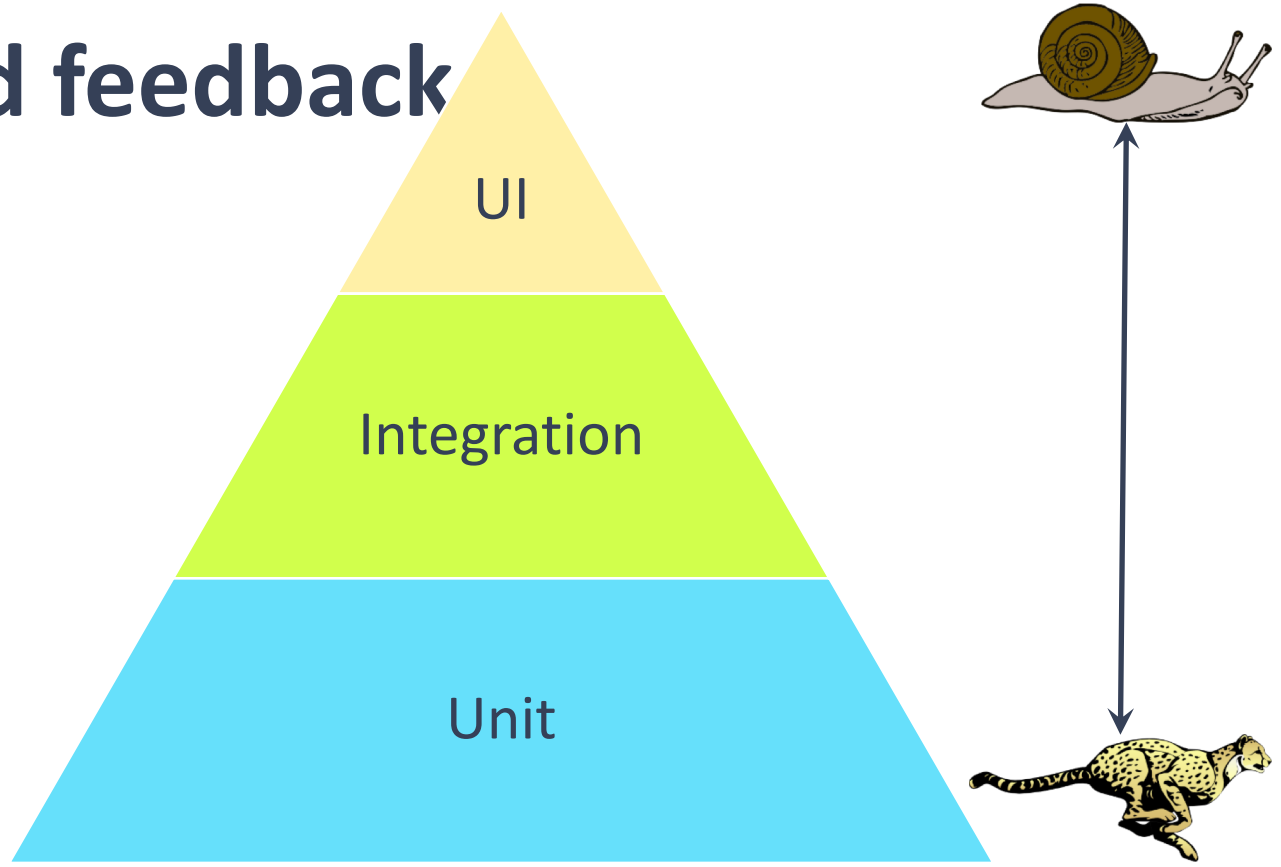


WHY TESTING IS IMPORTANT

So what makes a good feedback loop…

- It’s fast
- It’s reliable
- It isolates failures

- So think smaller, not larger
- Unit tests are awesome!



How to write kick arse unit tests

Ideally code should…

- Exhibit the SOLID principles
- Be easily refactored
- Be modifiable without having widespread effects

WRITING UNIT TEST

SOLID Principles - S

- **Single responsibility**
- Do one thing, do it right
- Applies to functions/methods, classes and entire applications
- E.g. Instagram, Uber, SnapChat

```
5 references | Ruskin Dantra, 2 days ago | 1 author, 1 change
public void Log(string message)
{
    Console.WriteLine(message);
}
```

WRITING UNIT TEST

SOLID Principles - O

- **Open closed principle**
- Artefacts you create should be open for extension…but closed for modification
- Input/output parameters
- Internals marked as private
- Returning an immutable collection

```
private readonly StringBuilder _stringBuilder;

1 reference | Ruskin Dantra, 2 days ago | 1 author, 1 change
public InMemoryPersister()
{
    _stringBuilder = new StringBuilder();
}
```

```
2 references | 0 changes | 0 authors, 0 changes
IEnumerable<string> AllOperations { get; }
```

SOLID Principles - L

- **Liskov's substitution principle**
- Objects in your code can be substituted by other subtypes of that object.
- Code correctness should not be affected.

```
0 references | 0 changes | 0 authors, 0 changes
public void PrintSomething()
{
    Print(new[] {1, 2, 3, 4, 5});
    Print(new List<int> {1, 2, 3, 4, 5});
}

2 references | 0 changes | 0 authors, 0 changes
private void Print(IList<int> items)...
```

WRITING UNIT TEST

SOLID Principles - I

- **Interface segregation**
- Better to have purpose specific interfaces rather than one general purpose interface

```
0 references | 0 changes | 0 authors, 0 changes
public interface IFax...

0 references | 0 changes | 0 authors, 0 changes
public interface IPhotoCopy...
```

SOLID Principles - D

- **Dependency inversion**
- Abstractions should not depend on details, details should depend on abstractions, aka interfaces should depend on other interfaces

```
0 references | 0 changes | 0 authors, 0 changes
public interface ICar
{
    0 references | 0 changes | 0 authors, 0 changes
    IEngine Engine { get; }
    0 references | 0 changes | 0 authors, 0 changes
    IGearBox GearBox { get; }
}
```

Exercises

promapp™

You'll need the worksheet

- https://s3-ap-southeast-2.amazonaws.com/sot2018-collateral/Workshop_v1.pdf

promapp™

Thanks!