What Happens If You Don't Dispose?

Elton Stoneman
geekswithblogs.net/eltonstoneman
@EltonStoneman

pluralsight
hardcore dev and IT training

What Happens If You Don't Dispose?

Finding IDisposable issues
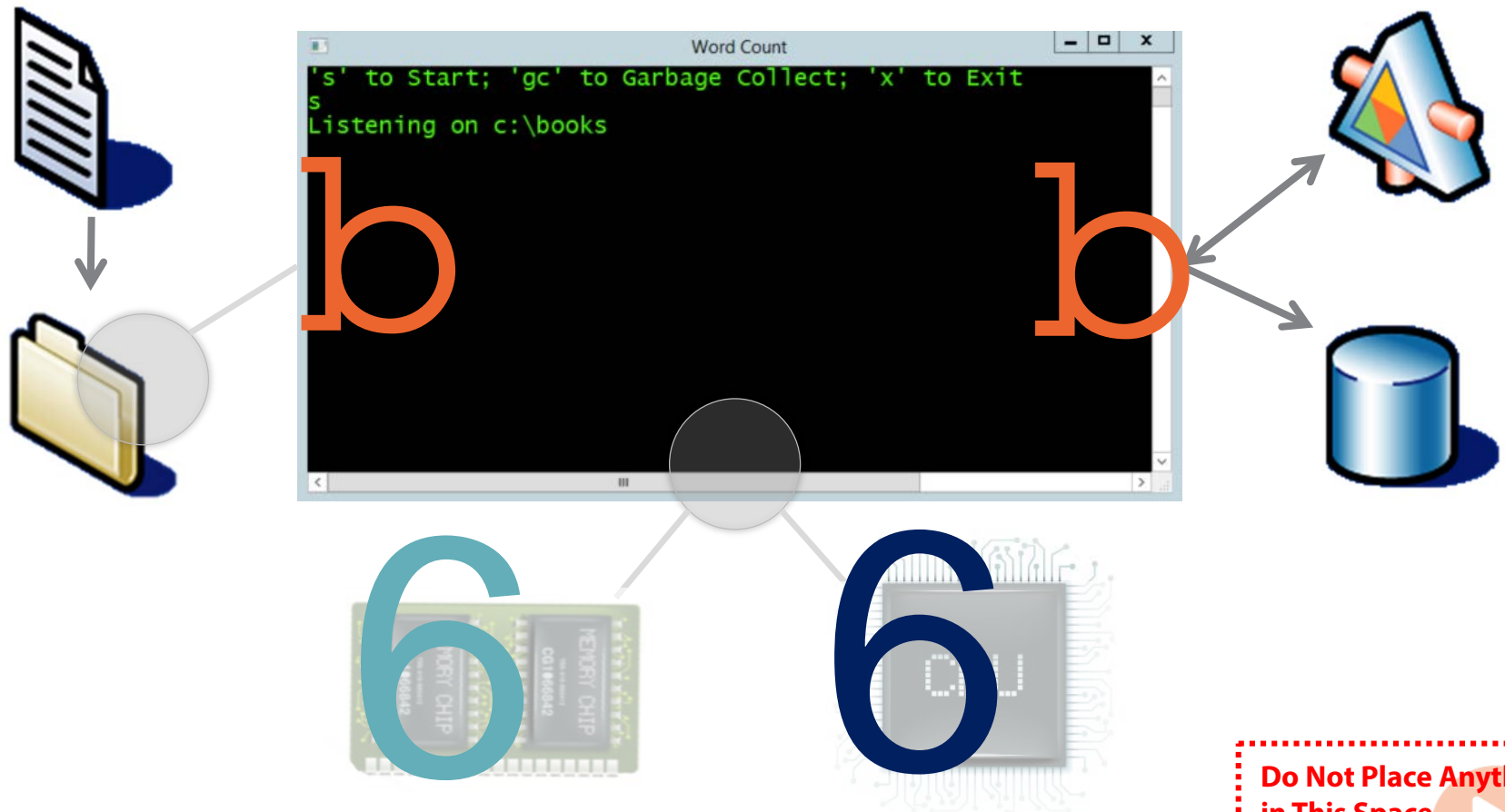
IDisposable and **domain-driven design**

Fixing IDisposable issues

What Happens If You Don't Dispose


```
Word Count                                    _  □  x
's' to Start; 'gc' to Garbage Collect; 'x' to Exit
s
Listening on c:\books
```

Diagnosing IDisposable Issues
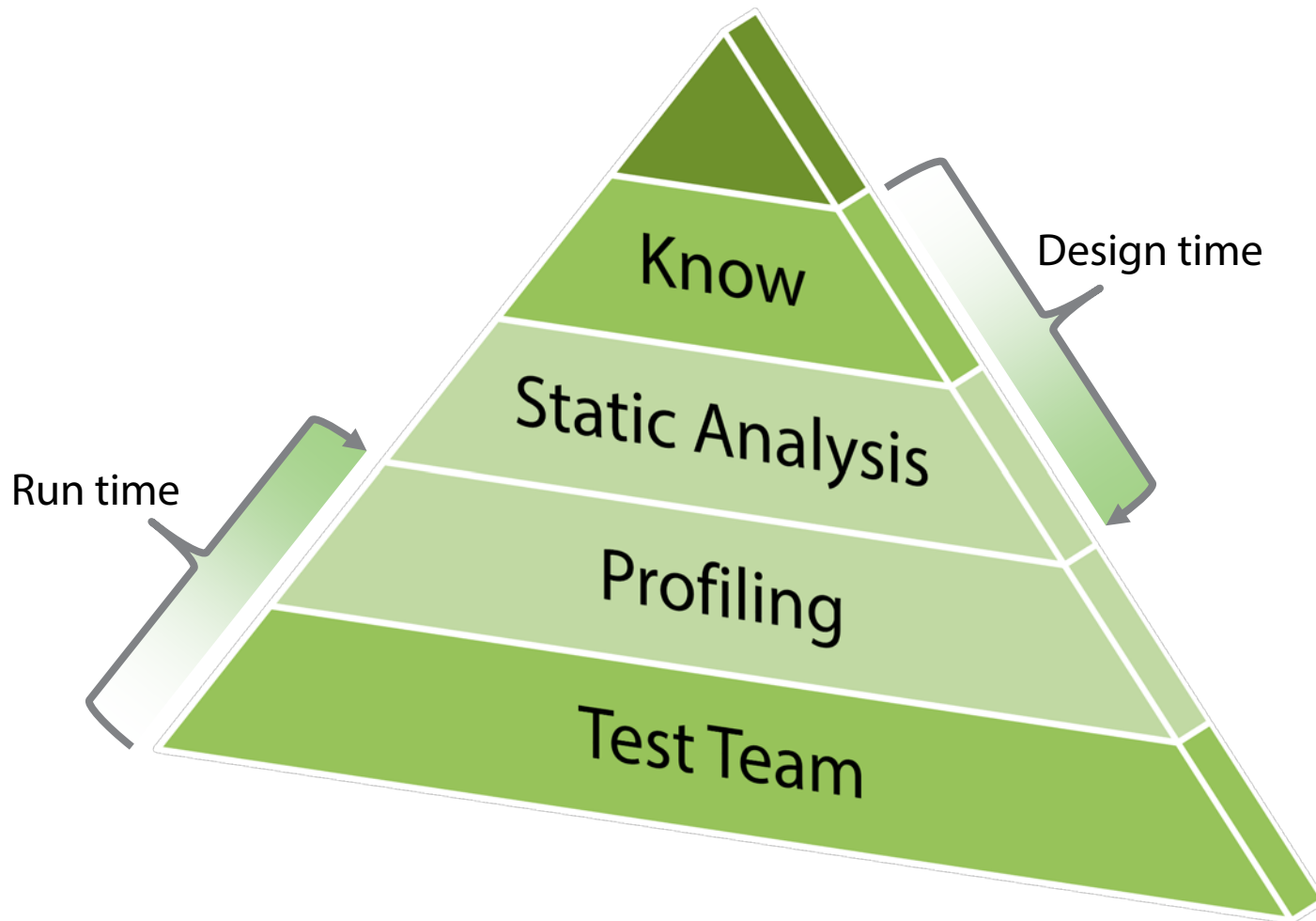
Know

Static Analysis

Profiling

Test Team

Design time

Run time

Fixing the Word Counting App
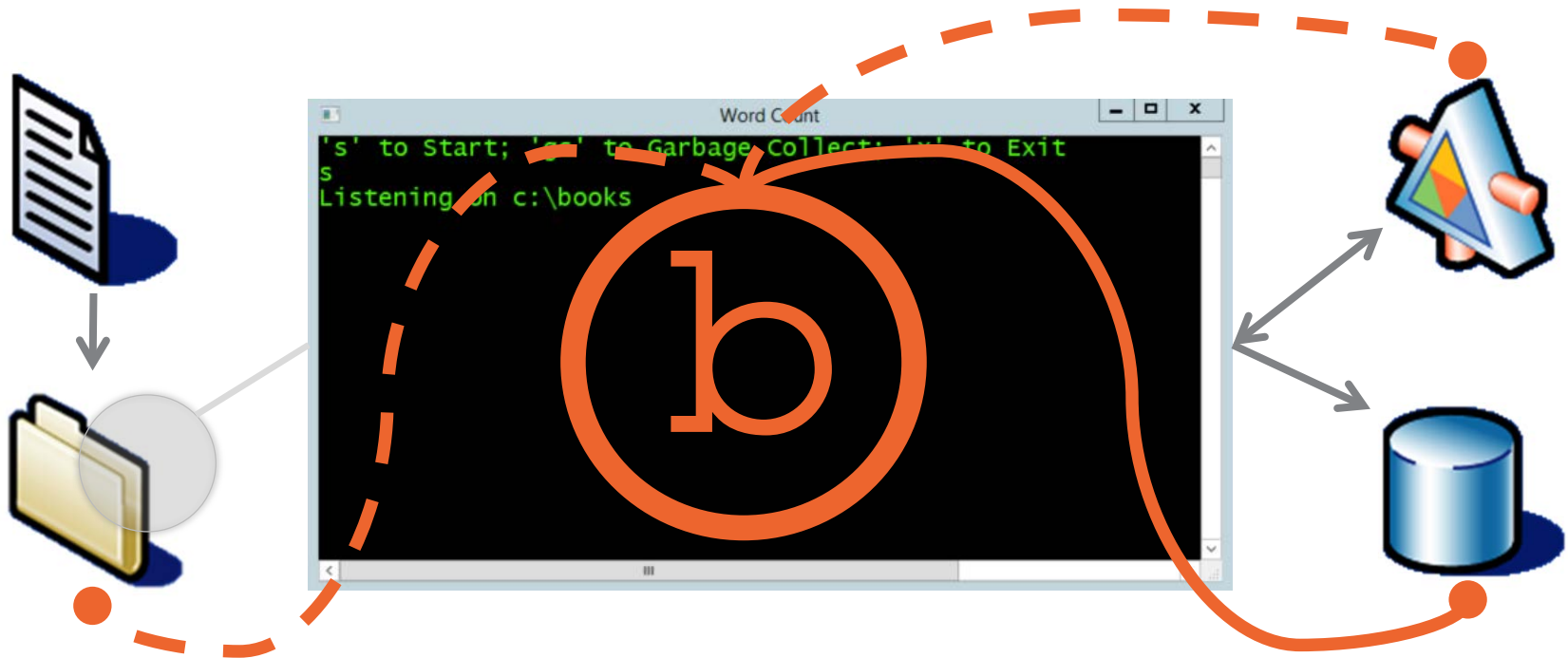
Word Count

```
's' to Start; 'gc' to Garbage Collect; 'x' to Exit
s
Listening on c:\books
```

Demo 1: Finding Problems at Run-Time

Goal

Identify potential run-time issues from not disposing resources

Walkthrough

Check for defects running the app

Walkthrough

Check for memory issues profiling the app

Demo 1: Finding Problems at Run-Time

Demo 1: Finding Problems at Run-Time

Word Count

's' to Start; 'gc' to Garbage Collect; 'x' to Exit
s
Listening on c:\books

Demo 1: Finding Problems at Run-Time

■ - original app

Demo 1: Finding Problems at Run-Time

Chart showing memory usage across Startup, First Run, Multiple Runs, and GC.Collect.

Y-axis: 0, 10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000

X-axis: Startup, First Run, Multiple Runs, GC.Collect

Original app (orange): 392, 15000, 63000, 63000

Domain-driven app (green): 6000, 71000, 74000, 74000

■ - original app   ■ - domain-driven app

Demo 2: Finding Problems at Design-Time

Goal

Identify potential design-time issues from not disposing

Walkthrough

Check for issues using static code analysis & manual checks

Walkthrough

Fix issues and check performance with profiler

Demo 2: Finding Problems at Design-Time

Demo 2: Finding Problems at Design-Time

CA2000 Dispose objects before...

CA2000 Dispose objects before...

CA2202 Do not dispose objects...
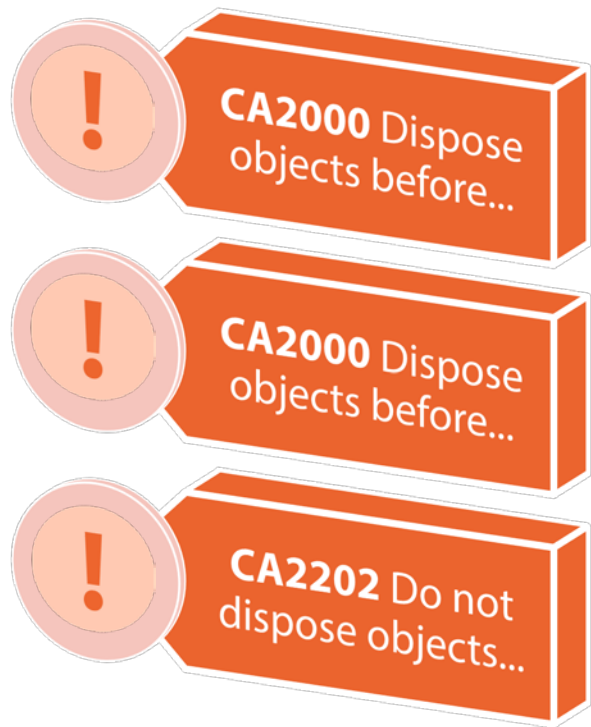
Static code analysis
Not enabled by default

"Recommended" rules
Do not include CA2000

Correctly set up
Find and fix issues quickly
But not all issues

Demo 2: Finding Problems at Design-Time

```
public void CreateFeedResults(...)
{
    var sqlConnection = new SqlConnection(...);
    sqlConnection.Open();
    var command = sqlConnection.CreateCommand();
    command.CommandType = CommandType.Text;
    command.CommandText = "...",
    command.ExecuteNonQuery();
}
```

CA2000 warning

Demo 2: Finding Problems at Design-Time

- original app

Demo 2: Finding Problems at Design-Time

Legend:
- **orange** – original app
- **light pink** – best practice version

Chart data:

	Startup	First Run	Multiple Runs	GC.Collect
original app	392	15000	63000	63000
best practice version	394	11000	12600	12600
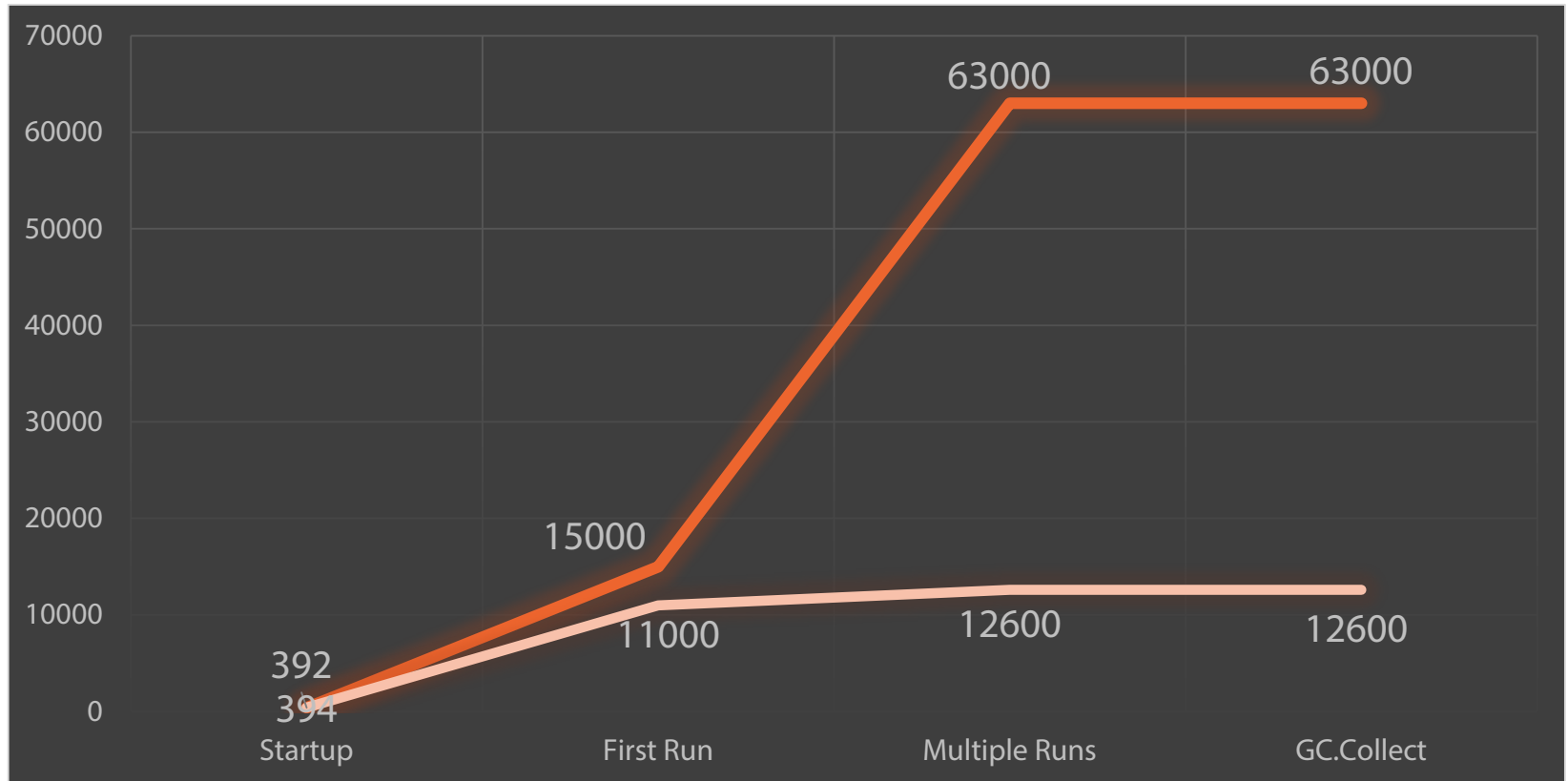
Best Practice #6

Enable Code Analysis with CA2000 enabled – but don't rely on it

Best Practice #6

Discovering IDisposables

```
using System;
using System.Linq;
using System.Text;
using System.Web;
using System.Xml.Linq;

namespace Solved.Scratchpad.Identity.Web
{
    public partial class ClaimInspector : System.Web
    {
        protected void Page_Load(object sender, Even
        {
            wrappedToken = XDocument.Parse(HttpC
            binaryToken = wrappedToken.Root.Desc
            tokenBytes = Convert.FromBase64Strin
            token = Encoding.UTF8.GetString(toke
                     wrappedToken.Root.Descen

            if (tokenType == TokenType.Swt)
            {
                ParseSWT(token);
            }
            else if (tokenType == TokenType.Jwt)
            {
                ParseJWT(token);
            }
```

Check class hierarchy

Go to definition

Code map

Trial-and-error

Try *using{}* block

Intellisense

Look for *.Dispose()*

Demo 3: Making IDisposable Discoverable

Goal

Implement IDisposable to aid discovery

Walkthrough

Fix issues in domain-driven version of app

Walkthrough

Implementing IDisposable and other interfaces

Demo 3: Making IDisposable Discoverable
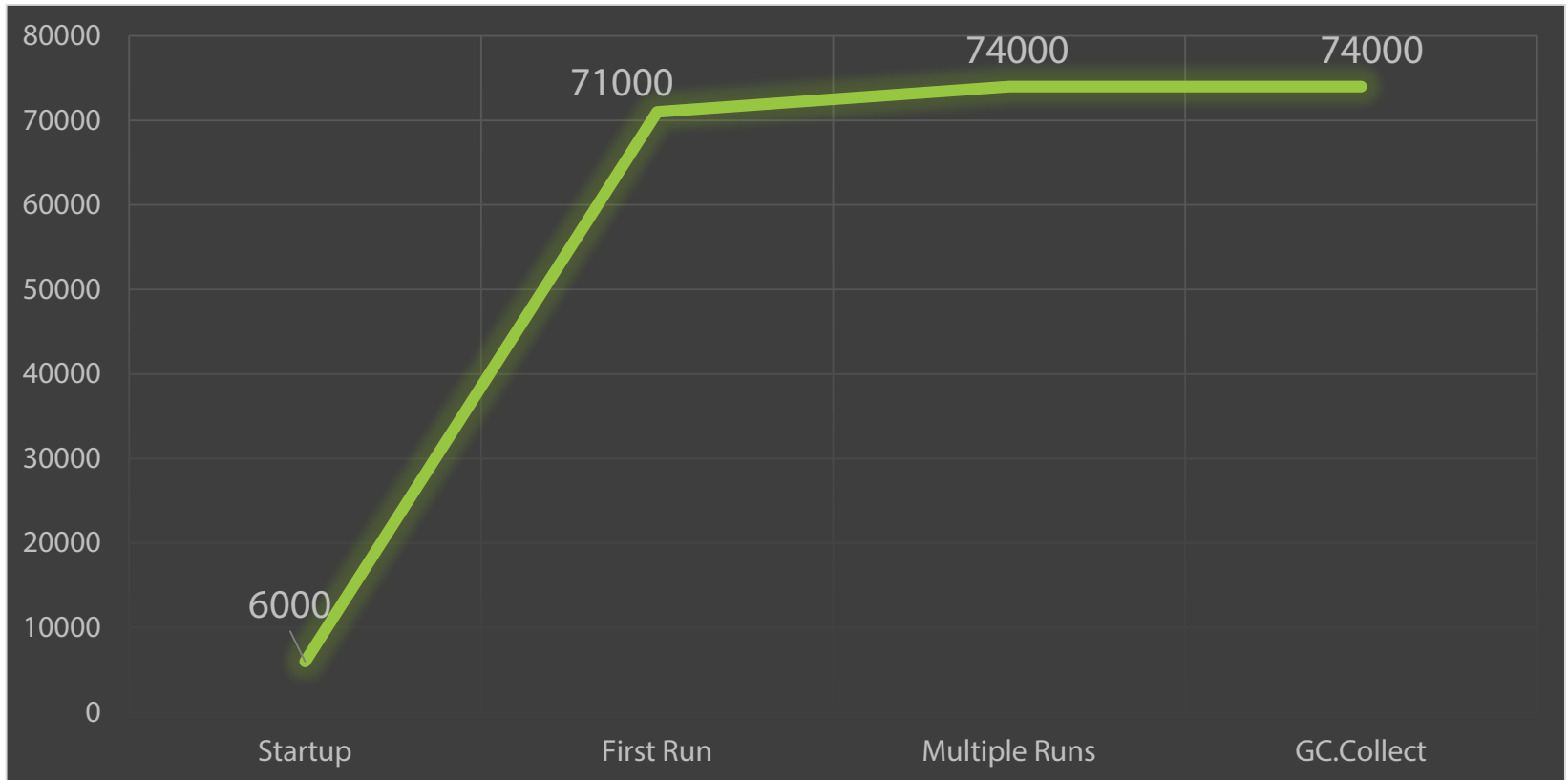
Demo 3: Making IDisposable Discoverable

■ - domain-driven app

Demo 3: Making IDisposable Discoverable

80000

71000

74000   74000

70000

68000   69000   69000

60000

50000

40000

30000

20000

10000

6000
6000

0

Startup   First Run   Multiple Runs   GC.Collect
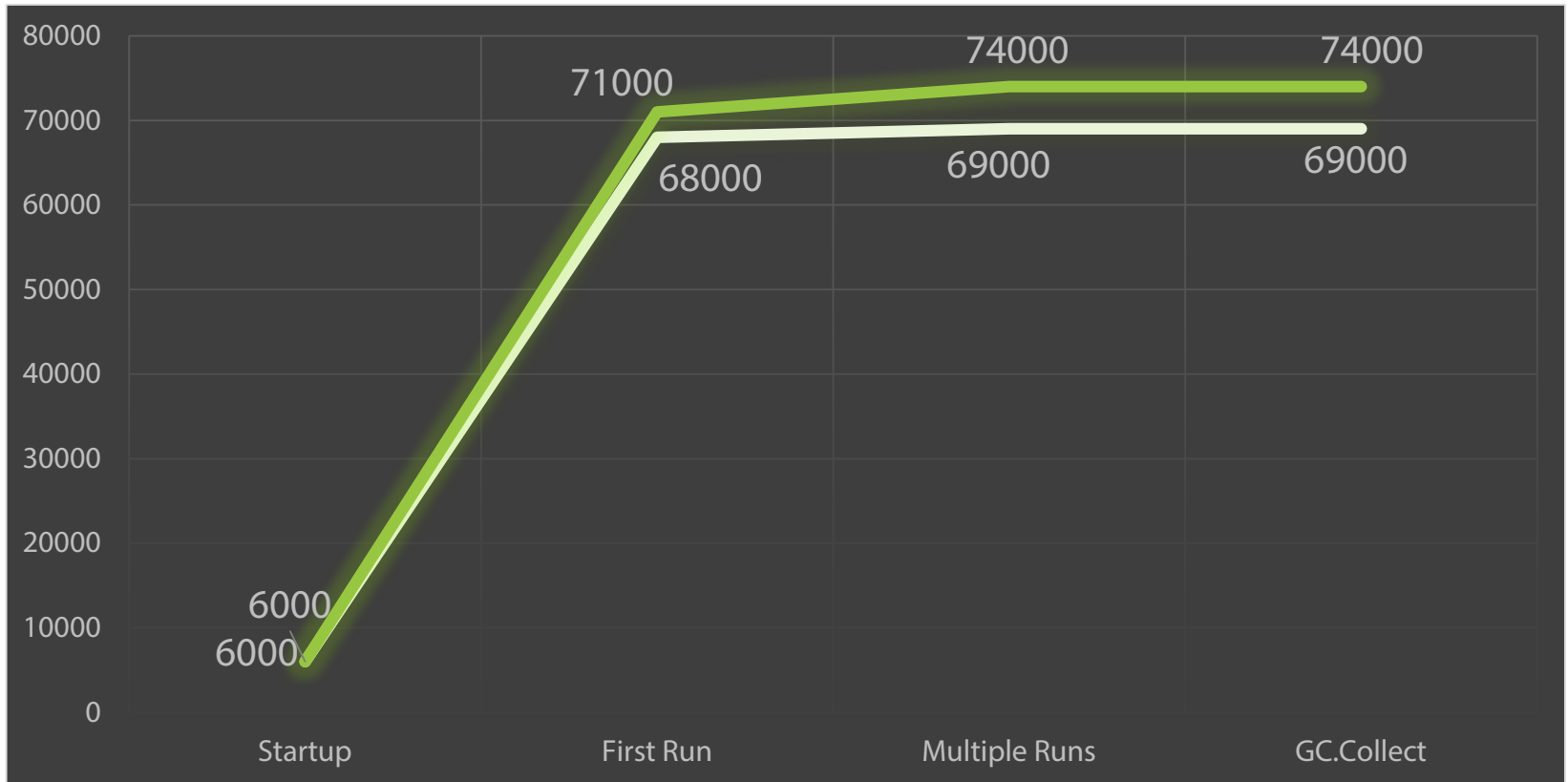
■ - domain-driven app   □ - best-practice version

Best Practice #7

If you implement an interface and use IDisposable fields, extend your interface from IDisposable

Best Practice #7

```csharp
public class BookFeedRepository : IBookFeedRepository
{
    private BookFeedContext _context;

    //...
}
```

```csharp
public interface IBookFeedRepository : IDisposable
{
    //...
}
```

Best Practice #8

If you implement IDisposable, don't implement it explicitly

Best Practice #8

```csharp
public class DifficultToDiscover : IDisposable
{
    void IDisposable.Dispose()
    {
        //...
    }
}
```

Best Practice #8

```
public class DifficultToDiscover : IDisposable
{
    void IDisposable.Dispose()
    {
        //...
    }
}
```

```
public class EasyToDiscover : IDisposable
{
    public void Dispose()
    {
        //...
    }
}
```
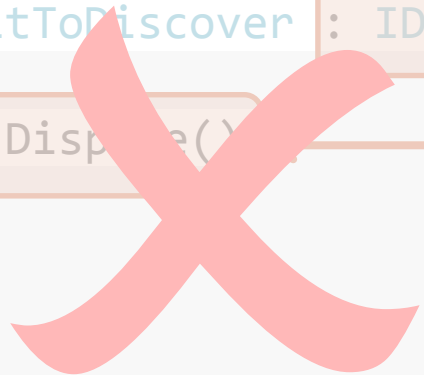
Summary

- **What happens if you don't dispose?** ☑
    - File locks & event handlers
    - Starving SQL connection pool & object leaks

- **Finding problems from not disposing** ☑
    - Run-time
        - Test team & profiling
    - Design-time
        - Static analysis & class knowledge

- **Fixing the demo solution** ☑
    - Applying the 8 best practices

Summary