ACCU 2023

GILDING THE ROSE

KEVLIN HENNEY

Gilding the Rose

Refactoring-Oriented Software Engineering

@KevlinHenney

@KevlinHenney
@kevlin@mastodon.social
about.me/kevlin
linkedin.com/in/kevlin
kevlinhenney.medium.com
kevlin@curbralan.com

Terry Hughes
@TerryHughes

it blows my mind how popular the Gilded Rose kata is, thanks @NotMyself for exposing it to the world. also, laugh everytime i read the setup

8:51 PM · Jul 16, 2013

KE

When did Terry Hughes create the Gilded Rose kata?

KE

When did Terry Hughes create the Gilded Rose kata?

Terry Hughes did not create the Gilded Rose Kata. The Gilded Rose Kata is a coding exercise that was originally created by Emily Bache in 2011. It is a refactoring exercise designed to help developers practice their skills in working with legacy code. Since its creation, the kata has gained popularity and has been adapted and modified by many developers and organizations.

Kata: Gilded Rose[33]

[33]With thanks to Terry Hughes, who originally designed this Kata, for permission to include it here.

Hi and welcome to team Gilded Rose. As you know, we are a small inn with a prime location in a prominent city ran by a friendly innkeeper named Allison.

We also buy and sell only the finest goods. Unfortunately, our goods are constantly degrading in quality as they approach their sell by date.

We have a system in place that updates our inventory for us.
It was developed by a no-nonsense type named Leeroy, who has moved on to new adventures.

legacy

✓ PRONUNCIATION ✓ SPELLINGS ✓ ETYMOLOGY ✓ QUOTATIONS

Find

Children of alumni were over 2½ times as likely to be admitted as those without such a connection. **2002** *Philadelphia* Oct. 4/2 'Being a legacy and having dad donate a million bucks isn't even a guarantee.' So, how *do* you get into Penn?

Draft partial entry September 2007

▸ *attrib.* Chiefly *Computing*. Designating software or hardware which, although outdated or limiting, is an integral part of a computer system and difficult to replace. Also in extended use.

1989 *Industry Week* 20 Nov. 46 A migration mechanism enabling the automated conversion of legacy databases and application systems to the integration platform. **1993** *Computer Weekly* 14 Oct. 34/6 Too many IT people ossify with the IT they are comfortable with—they become legacy people, and that's dangerous. **1998** *Byte* (U.K. ed.) June 89/1 The project consists of about 25 legacy FORTRAN codes glued together by Perl. **2005** *Computer Weekly* 29 Nov. 26/1 IT organisations should not dismiss these applications as useless legacy artifacts, destined for rip-and-replace.

legacy

✓ PRONUNCIATION ✓ SPELLINGS ✓ ETYMOLOGY ✓ QUOTATIONS

Find

Children of alumni were over 2½ times as likely to be admitted as those without such a connection. **2002** *Philadelphia* Oct. 4/2 'Being a legacy and having dad donate a million bucks isn't even a guarantee.' So, how *do* you get into Penn?

Draft partial entry September 2007

▸ *attrib.* Chiefly *Computing*. Designating software or hardware which, although outdated or limiting, is an integral part of a computer system and difficult to replace. Also in extended use.

1989 *Industry Week* 20 Nov. 46 A migration mechanism enabling the automated conversion of legacy databases and application systems to the integration platform. **1993** *Computer Weekly* 14 Oct. 34/6 Too many IT people ossify with the IT they are comfortable with—they become legacy people, and that's dangerous. **1998** *Byte* (U.K. ed.) June 89/1 The project consists of about 25 legacy FORTRAN codes glued together by Perl. **2005** *Computer Weekly* 29 Nov. 26/1 IT organisations should not dismiss these applications as useless legacy artifacts, destined for rip-and-replace.

Your task is to add the new feature to our system so that we can begin selling a new category of items.

First an introduction to our system:

- All items have a SellIn value which denotes the number of days we have to sell the item
- All items have a Quality value which denotes how valuable the item is
- At the end of each day our system lowers both values for every item

Pretty simple, right?

Well this is where it gets interesting:

- Once the sell by date has passed, Quality degrades twice as fast
- The Quality of an item is never negative
- “Aged Brie” actually increases in Quality the older it gets
- The Quality of an item is never more than 50

…

- “Sulfuras”, being a legendary item, never has to be sold or decreases in Quality
- “Backstage passes”, like aged brie, increases in Quality as it’s SellIn value approaches; Quality increases by 2 when there are 10 days or less and by 3 when there are 5 days or less but Quality drops to 0 after the concert

We have recently signed a supplier of conjured items. This requires an update to our system:

- “Conjured” items degrade in Quality twice as fast as normal items

this is where it gets interesting...

```
public class GildedRose
{
    public IList<Item> Items;

    public void UpdateQuality()
    {
        for (var i = 0; i < Items.Count; i++)
        {
            if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;

                    if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].SellIn < 11)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }

                        if (Items[i].SellIn < 6)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }
                    }
                }
            }

            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].SellIn = Items[i].SellIn - 1;
            }

            if (Items[i].SellIn < 0)
            {
                if (Items[i].Name != "Aged Brie")
                {
                    if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].Quality > 0)
                        {
                            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                            {
                                Items[i].Quality = Items[i].Quality - 1;
                            }
                        }
                    }
                    else
                    {
                        Items[i].Quality = Items[i].Quality - Items[i].Quality;
                    }
                }
                else
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    public class Item
    {
        public string Name { get; set; }

        public int SellIn { get; set; }

        public int Quality { get; set; }
    }
}
```


technical
debt


technical
neglect

```
public class GildedRose
{
    public IList<Item> Items;

    public void UpdateQuality()
    {
        for (var i = 0; i < Items.Count; i++)
        {
            if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;

                    if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].SellIn < 11)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }

                        if (Items[i].SellIn < 6)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }
                    }
                }
            }

            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].SellIn = Items[i].SellIn - 1;
            }

            if (Items[i].SellIn < 0)
            {
                if (Items[i].Name != "Aged Brie")
                {
                    if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].Quality > 0)
                        {
                            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                            {
                                Items[i].Quality = Items[i].Quality - 1;
                            }
                        }
                    }
                    else
                    {
                        Items[i].Quality = Items[i].Quality - Items[i].Quality;
                    }
                }
                else
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    public class Item
    {
        public string Name { get; set; }

        public int SellIn { get; set; }

        public int Quality { get; set; }
    }
}
```

```
public class GildedRose
{
    public IList<Item> Items;

    public void UpdateQuality()
    {
        for (var i = 0; i < Items.Count; i++)
        {
            if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;

                    if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].SellIn < 11)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }

                        if (Items[i].SellIn < 6)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }
                    }
                }
            }

            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].SellIn = Items[i].SellIn - 1;
            }

            if (Items[i].SellIn < 0)
            {
                if (Items[i].Name != "Aged Brie")
                {
                    if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].Quality > 0)
                        {
                            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                            {
                                Items[i].Quality = Items[i].Quality - 1;
                            }
                        }
                    }
                    else
                    {
                        Items[i].Quality = Items[i].Quality - Items[i].Quality;
                    }
                }
                else
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    public class Item
    {
        public string Name { get; set; }

        public int SellIn { get; set; }

        public int Quality { get; set; }
    }
}
```

```
public class GildedRose
{
    public IList<Item> Items;

    public void UpdateQuality()
    {
        ...
    }

    public class Item
    {
        public string Name { get; set; }

        public int SellIn { get; set; }

        public int Quality { get; set; }
    }
}
```

Feel free to make any changes to the UpdateQuality method and add any new code as long as everything still works correctly.

However, do not alter the Item class or Items property as those belong to the goblin in the corner who will insta-rage and one-shot you as he doesn't believe in shared code ownership (you can make the UpdateQuality method and Items property static if you like, we'll cover for you).

Feel free to make any changes to the UpdateQuality method and add any new code as long as everything still works correctly.

However, do not alter the Item class or Items property as those belong to the goblin in the corner who will insta-rage and one-shot you as he doesn't believe in shared code ownership ~~(you can make the UpdateQuality method and Items property static if you like, we'll cover for you)~~.

Feel free to make any changes to the UpdateQuality method and add any new code as long as everything still works correctly.

However, do not alter the Item class or Items property as those belong to the goblin in the corner who will insta-rage and one-shot you as he doesn't believe in shared code ownership.

works correctly?

```
public class GildedRoseTests
{
    [Test]
    public void Example_of_existing_functionality()
    {
        var gildedRose = new GildedRose()
        {
            Items = new List<Item>
            {
                new Item {Name = "+5 Dexterity Vest", SellIn = 10, Quality = 20},
                new Item {Name = "Aged Brie", SellIn = 2, Quality = 0},
                new Item {Name = "Elixir of the Mongoose", SellIn = 5, Quality = 7},
                new Item {Name = "Sulfuras, Hand of Ragnaros", SellIn = 0, Quality = 80},
                new Item {Name = "Backstage passes to a TAFKAL80ETC concert",  SellIn = 15, Quality = 20},
            }
        };

        gildedRose.UpdateQuality();

        var items = gildedRose.Items;
        Assert.AreEqual(9, items[0].SellIn);
        Assert.AreEqual(19, items[0].Quality);
        Assert.AreEqual(1, items[1].SellIn);
        Assert.AreEqual(1, items[1].Quality);
        Assert.AreEqual(4, items[2].SellIn);
        Assert.AreEqual(6, items[2].Quality);
        Assert.AreEqual(0, items[3].SellIn);
        Assert.AreEqual(80, items[3].Quality);
        Assert.AreEqual(14, items[4].SellIn);
        Assert.AreEqual(21, items[4].Quality);
    }
    ...
}
```

```
public class GildedRoseTests
{
    [Test]
    public void Example_of_existing_functionality()
    {
        var gildedRose = new GildedRose()
        {
            Items = new List<Item>
            {
                new Item {Name = "+5 Dexterity Vest", SellIn = 10, Quality = 20},
                new Item {Name = "Aged Brie", SellIn = 2, Quality = 0},
                new Item {Name = "Elixir of the Mongoose", SellIn = 5, Quality = 7},
                new Item {Name = "Sulfuras, Hand of Ragnaros", SellIn = 0, Quality = 80},
                new Item {Name = "Backstage passes to a TAFKAL80ETC concert",  SellIn = 15, Quality = 20},
            }
        };

        gildedRose.UpdateQuality();

        var items = gildedRose.Items;
        Assert.AreEqual(9, items[0].SellIn);
        Assert.AreEqual(19, items[0].Quality);
        Assert.AreEqual(1, items[1].SellIn);
        Assert.AreEqual(1, items[1].Quality);
        Assert.AreEqual(4, items[2].SellIn);
        Assert.AreEqual(6, items[2].Quality);
        Assert.AreEqual(0, items[3].SellIn);
        Assert.AreEqual(80, items[3].Quality);
        Assert.AreEqual(14, items[4].SellIn);
        Assert.AreEqual(21, items[4].Quality);
    }
    ...
}
```

```
public class GildedRoseTests
{
    ...
    [Test]
    [Ignore("Not yet implemented")]
    public void Example_of_proposed_functionality()
    {
        var gildedRose = new GildedRose()
        {
            Items = new List<Item>
            {
                new Item {Name = "+5 Dexterity Vest", SellIn = 10, Quality = 20},
                new Item {Name = "Aged Brie", SellIn = 2, Quality = 0},
                new Item {Name = "Elixir of the Mongoose", SellIn = 5, Quality = 7},
                new Item {Name = "Sulfuras, Hand of Ragnaros", SellIn = 0, Quality = 80},
                new Item {Name = "Backstage passes to a TAFKAL80ETC concert",  SellIn = 15, Quality = 20},
                new Item {Name = "Conjured Mana Cake", SellIn = 3, Quality = 6}
            }
        };

        gildedRose.UpdateQuality();

        var items = gildedRose.Items;
        Assert.AreEqual(2, items[5].SellIn);
        Assert.AreEqual(4, items[5].Quality);
    }
}
```

```
public class GildedRoseTests
{
    ...
    [Test]
    [Ignore("Not yet implemented")]
    public void Example_of_proposed_functionality()
    {
        var gildedRose = new GildedRose()
        {
            Items = new List<Item>
            {
                new Item {Name = "+5 Dexterity Vest", SellIn = 10, Quality = 20},
                new Item {Name = "Aged Brie", SellIn = 2, Quality = 0},
                new Item {Name = "Elixir of the Mongoose", SellIn = 5, Quality = 7},
                new Item {Name = "Sulfuras, Hand of Ragnaros", SellIn = 0, Quality = 80},
                new Item {Name = "Backstage passes to a TAFKAL80ETC concert",  SellIn = 15, Quality = 20},
                new Item {Name = "Conjured Mana Cake", SellIn = 3, Quality = 6}
            }
        };

        gildedRose.UpdateQuality();

        var items = gildedRose.Items;
        Assert.AreEqual(2, items[5].SellIn);
        Assert.AreEqual(4, items[5].Quality);
    }
}
```

Compatibility Rule

Make any changes to the UpdateQuality method and add any new code as long as everything still works correctly.

Goblin Rule

Do not alter the Item class or Items property as those belong to the goblin in the corner who will insta-rage and one-shot you as he doesn't believe in shared code ownership.

```
public class GildedRose
{
    public IList<Item> Items;

    public void UpdateQuality()
    {
        for (var i = 0; i < Items.Count; i++)
        {
            if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;

                    if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].SellIn < 11)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }

                        if (Items[i].SellIn < 6)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }
                    }
                }
            }

            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].SellIn = Items[i].SellIn - 1;
            }

            if (Items[i].SellIn < 0)
            {
                if (Items[i].Name != "Aged Brie")
                {
                    if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].Quality > 0)
                        {
                            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                            {
                                Items[i].Quality = Items[i].Quality - 1;
                            }
                        }
                    }
                    else
                    {
                        Items[i].Quality = Items[i].Quality - Items[i].Quality;
                    }
                }
                else
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    public class Item
    {
        public string Name { get; set; }

        public int SellIn { get; set; }

        public int Quality { get; set; }
    }
}
```

```
public class GildedRose
{
    public IList<Item> Items;

    public void UpdateQuality()
    {
        for (var i = 0; i < Items.Count; i++)
        {
            if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;

                    if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].SellIn < 11)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }

                        if (Items[i].SellIn < 6)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }
                    }
                }
            }

            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].SellIn = Items[i].SellIn - 1;
            }

            if (Items[i].SellIn < 0)
            {
                if (Items[i].Name != "Aged Brie")
                {
                    if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].Quality > 0)
                        {
                            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                            {
                                Items[i].Quality = Items[i].Quality - 1;
                            }
                        }
                    }
                    else
                    {
                        Items[i].Quality = Items[i].Quality - Items[i].Quality;
                    }
                }
                else
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    public class Item
    {
        public string Name { get; set; }

        public int SellIn { get; set; }

        public int Quality { get; set; }
    }
}
```

```
public class GildedRose
{
    public IList<Item> Items;

    public void UpdateQuality()
    {
        for (var i = 0; i < Items.Count; i++)
        {
            if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;

                    if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].SellIn < 11)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }

                        if (Items[i].SellIn < 6)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }
                    }
                }
            }

            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].SellIn = Items[i].SellIn - 1;
            }

            if (Items[i].SellIn < 0)
            {
                if (Items[i].Name != "Aged Brie")
                {
                    if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].Quality > 0)
                        {
                            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                            {
                                Items[i].Quality = Items[i].Quality - 1;
                            }
                        }
                    }
                    else
                    {
                        Items[i].Quality = Items[i].Quality - Items[i].Quality;
                    }
                }
                else
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    public class Item
    {
        public string Name { get; set; }

        public int SellIn { get; set; }

        public int Quality { get; set; }
    }
}
```

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
                else
                {
                    Items[i].Quality = Items[i].Quality - Items[i].Quality;
                }
            }
            else
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;
                }
            }
        }
    }
}
```


Just one more thing...
© Universal

Just for clarification, an item can never have its Quality increase above 50, however “Sulfuras” is a legendary item and as such its Quality is 80 and it never alters.

```
public class GildedRose
{
    public IList<Item> Items;

    public void UpdateQuality()
    {
        for (var i = 0; i < Items.Count; i++)
        {
            if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;

                    if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].SellIn < 11)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }

                        if (Items[i].SellIn < 6)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }
                    }
                }
            }

            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].SellIn = Items[i].SellIn - 1;
            }

            if (Items[i].SellIn < 0)
            {
                if (Items[i].Name != "Aged Brie")
                {
                    if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].Quality > 0)
                        {
                            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                            {
                                Items[i].Quality = Items[i].Quality - 1;
                            }
                        }
                    }
                    else
                    {
                        Items[i].Quality = Items[i].Quality - Items[i].Quality;
                    }
                }
                else
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    public class Item
    {
        public string Name { get; set; }

        public int SellIn { get; set; }

        public int Quality { get; set; }
    }
}
```

;{;{;;{"""{{""{;}}}{{;""{{{;}}{{;}}}}""{;}{""{""{{""{;}}}{;}}{{;}}}}}{{;;}{;;}{;;}}}

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

await	0
break	0
case	0
catch	0
continue	0
default	0
do	0
else	3
finally	0
for	1
foreach	0
goto	0
if	16
lock	0
return	0
switch	0
throw	0
try	0
while	0
yield	0

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

await	0
break	0
case	0
catch	0
continue	0
default	0
do	0
else	3
finally	0
for	1
foreach	0
goto	0
if	16
lock	0
return	0
switch	0
throw	0
try	0
while	0
yield	0

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

```
if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes t
{
    if (Items[i].Quality > 0)
    {
        if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
        {
            Items[i].Quality = Items[i].Quality - 1;
        }
    }
}
else
{
    if (Items[i].Quality < 50)
    {
        Items[i].Quality = Items[i].Quality + 1;

        if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert"
        {
            if (Items[i].SellIn < 11)
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;
                }
            }

            if (Items[i].SellIn < 6)
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;
                }
            }
        }
    }
}
```

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

```
if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
{
    Items[i].SellIn = Items[i].SellIn - 1;
}
```

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

```
if (Items[i].SellIn < 0)
{
    if (Items[i].Name != "Aged Brie")
    {
        if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert"
        {
            if (Items[i].Quality > 0)
            {
                if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                {
                    Items[i].Quality = Items[i].Quality - 1;
                }
            }
        }
        else
        {
            Items[i].Quality = Items[i].Quality - Items[i].Quality;
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;
        }
    }
}
```

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

today

date line

tomorrow


Kevlin Henney 编
李军 译 吕骏 审校
電子工業出版社
O'REILLY
オライリー・ジャパン
97
Collective Wisdom
from the Experts
97 Things Every
Programmer
Should Know
O'REILLY
Edited by Kevlin Henney

Thinking in States

In most real-world situations, people's relaxed attitude to state is not an issue. Unfortunately, however, many programmers are quite vague about state too — and that is a problem.

Niclas Nilsson

97-things-every-x-should-know.gitbooks.io/97-things-every-programmer-should-know/content/en/thing_84

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

item[i].Quality

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

item[i].Quality = …

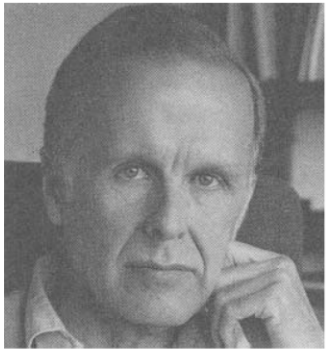
```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

~32%

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    else
    {
        if (Items[i].Quality < 50)
        {
            Items[i].Quality = Items[i].Quality + 1;

            if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].SellIn < 11)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }

                if (Items[i].SellIn < 6)
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
    {
        Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0)
    {
        if (Items[i].Name != "Aged Brie")
        {
            if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                Items[i].Quality = Items[i].Quality - Items[i].Quality;
            }
        }
        else
        {
            if (Items[i].Quality < 50)
            {
                Items[i].Quality = Items[i].Quality + 1;
            }
        }
    }
}
```

Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs

John Backus
IBM Research Laboratory, San Jose



Author's address: 91 Saint Germain Ave., San Francisco, CA 94114.

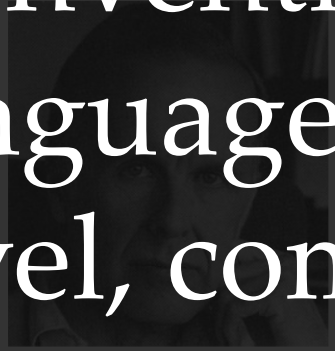
Conventional programming languages are growing ever more enormous, but not stronger. Inherent defects at the most basic level cause them to be both fat and weak: their primitive word-at-a-time style of programming inherited from their common ancestor—the von Neumann computer, their close coupling of semantics to state transitions, their division of programming into a world of expressions and a world of statements, their inability to effectively use powerful combining forms for building new programs from existing ones, and their lack of useful mathematical properties for reasoning about programs.

An alternative functional style of programming is founded on the use of combining forms for creating programs. Functional programs deal with structured data, are often nonrepetitive and nonrecursive, are hierarchically constructed, do not name their arguments, and do not require the complex machinery of procedure declarations to become generally applicable. Combining forms can use high level programs to build still higher level ones in a style not possible in conventional languages.

Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs

John Backus
IBM Research Laboratory, San Jose

Conventional programming languages are basically high level, complex versions of the von Neumann computer.



Author's address: 91 Saint Germain Ave., San Francisco, CA 94114.

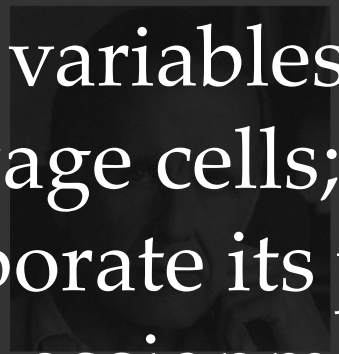
[illegible] ever more enormous, but not stronger. Inherent defects at the most basic level cause them to be both fat and weak: their primitive word-at-a-time style of programming [illegible] their common bottleneck [illegible] their [illegible] state transitions, their division of programming into a world of expressions and a world of statements, their inability to effectively use powerful combining forms for [illegible] and the lack of useful [illegible] programs.

An alternative functional style of programming is founded on the use of combining forms for creating programs. Functional programs deal with [illegible] archically constructed, do not name their arguments, and do not require the complex machinery of procedure declarations to become generally applicable. Combining forms can use high level programs to build still higher level ones in a style not possible in conventional languages.

Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs

John Backus
IBM Research Laboratory, San Jose

Von Neumann programming languages use variables to imitate the computer's storage cells; control statements elaborate its jump and test instructions; and assignment statements imitate its fetching, storing, and arithmetic.



Author's address: 91 Saint Germain Ave., San Francisco, CA 94114.

Conventional programming languages are growing ever more enormous, but not stronger. Inherent defects [illegible] weak: their primitive word-at-a-time style of programming inherited from their common ancestor—the von [illegible] world of expressions and a world of statements, their inability to effectively use powerful combining forms for [illegible] of useful mathematical properties for reasoning about programs.

[illegible] programs. Functional programs deal with structured data, are often nonrepetitive and nonrecursive, are hierarchically constructed [illegible] their arguments, and do not require the complex machinery of procedure declarations to become generally applicable. Combining forms can use high level programs to build still higher level ones in a style not possible in conventional languages.

commitstrip.com/en/2021/06/22/it-haunts-us

What is refactoring-oriented software development?
A variant of test-driven development that places a special emphasis on refactoring.

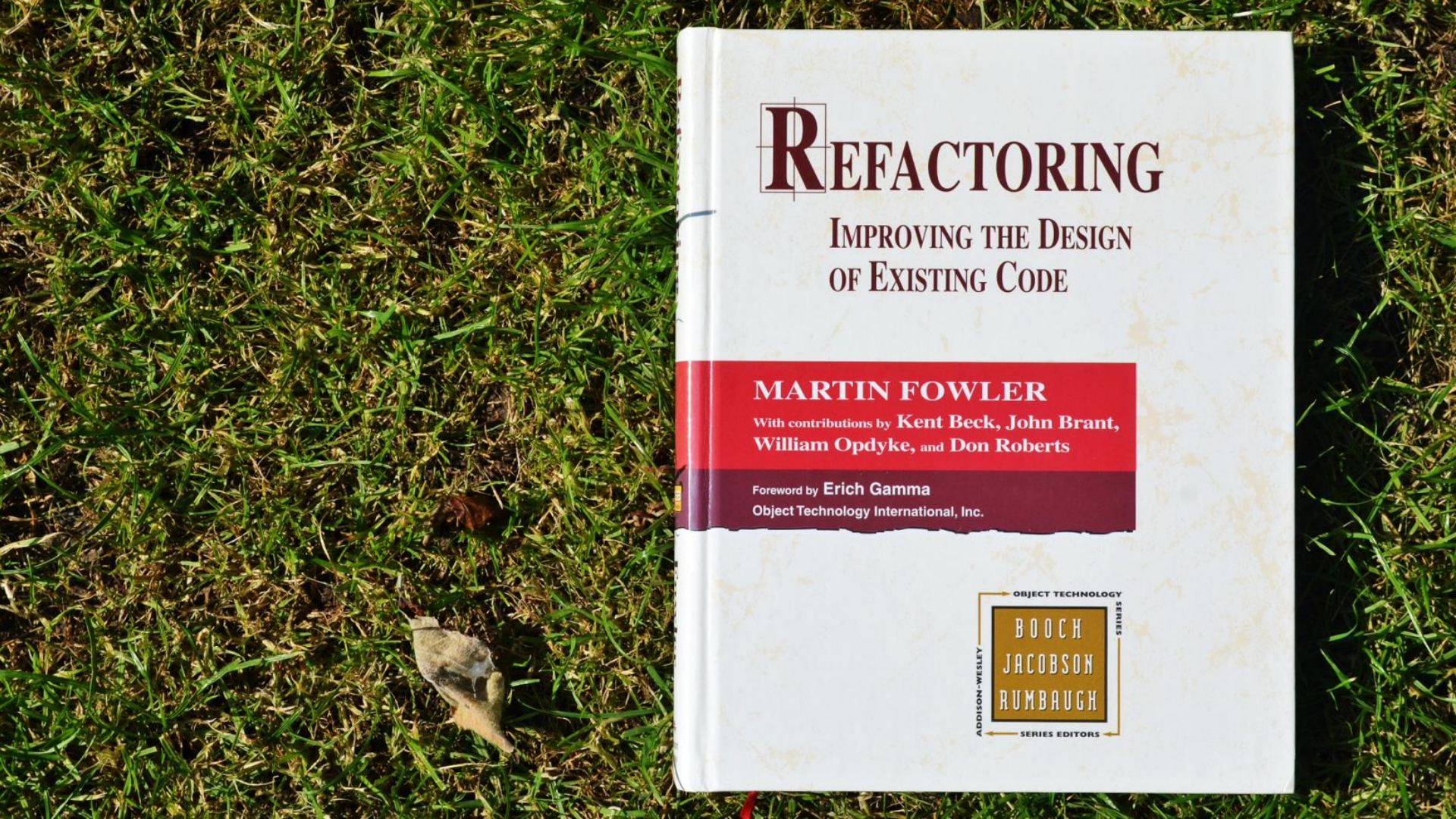
Roly Perera

"Refactoring-driven software engineering in Python"

A methodology for software development that relies heavily on refactoring and tests — to check for behavioural invariance not behavioural correctness.

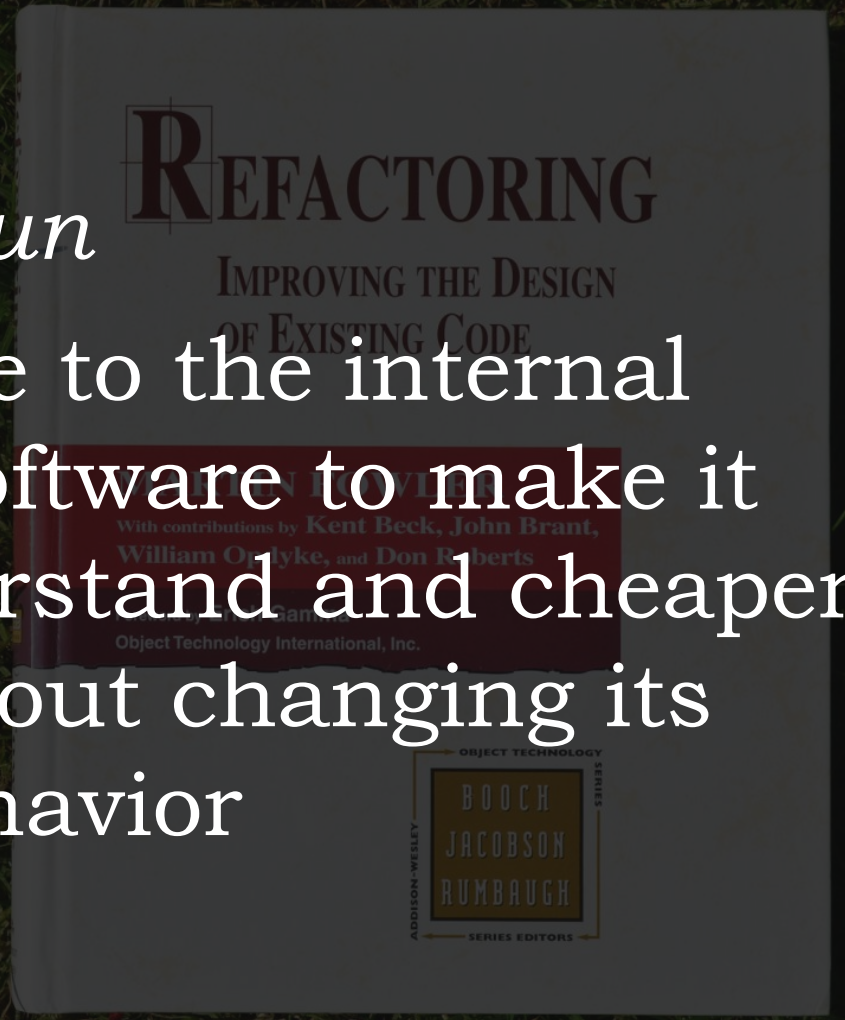
Roly Perera

"Refactoring-driven software engineering in Python"


REFACTORING
IMPROVING THE DESIGN OF EXISTING CODE
MARTIN FOWLER
With contributions by Kent Beck, John Brant, William Opdyke, and Don Roberts
Foreword by Erich Gamma
Object Technology International, Inc.
OBJECT TECHNOLOGY SERIES
BOOCH
JACOBSON
RUMBAUGH
ADDISON-WESLEY
SERIES EDITORS

Refactoring, *noun*

a change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior



without changing

The Addison-Wesley Signature Series

"Any fool can write code that a computer can understand.
Good programmers write code that humans can understand."
—M. Fowler (1999)

A Martin Fowler Signature Book

Refactoring

Improving the Design of Existing Code

Martin Fowler
with contributions by
Kent Beck

Second Edition

REFACTORING OBJECT-ORIENTED FRAMEWORKS

William F. Opdyke, Ph.D.
Department of Computer Science
University of Illinois at Urbana-Champaign, 1992
Ralph E. Johnson, Advisor

This thesis defines a set of program restructuring operations (refactorings) that support the design, evolution and reuse of object-oriented application frameworks.

The focus of the thesis is on automating the refactorings in a way that preserves the behavior of a program. The refactorings are defined to be behavior preserving, provided that their preconditions are met. Most of the refactorings are simple to implement and it is almost trivial to show that they are behavior preserving. However, for a few refactorings, one or more of their preconditions are in general undecidable. Fortunately, for some cases it can be determined

REFACTORING OBJECT-ORIENTED FRAMEWORKS

William F. Opdyke, Ph.D.
Department of Computer Science
University of Illinois at Urbana-Champaign, 1992
Ralph E. Johnson, Advisor

The refactorings are defined to be behavior preserving

This thesis defines a set of program restructuring operations (refactorings) that support the design, evolution and reuse of object-oriented application frameworks.

The focus of the thesis is on automating the refactorings in a way that preserves the behavior of a program. The refactorings are defined to be behavior preserving, provided that their preconditions are met. Most of the refactorings are simple to implement and it is almost trivial to show that they are behavior preserving. However, for a few refactorings, one or more of their preconditions are in general undecidable. Fortunately, for some cases it can be determined

Write Tests for People

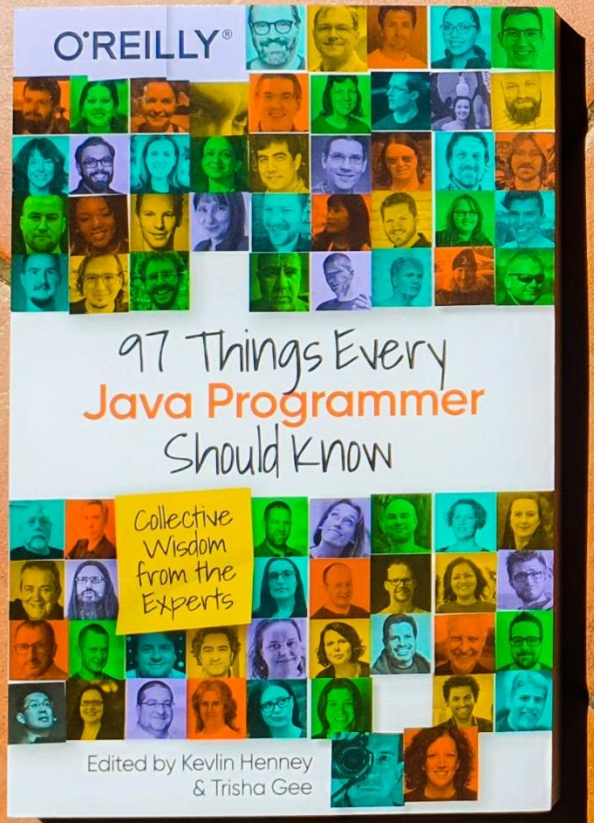
So who should you be writing the tests for?
For the person trying to understand your code.
Good tests act as documentation for the code they are testing.

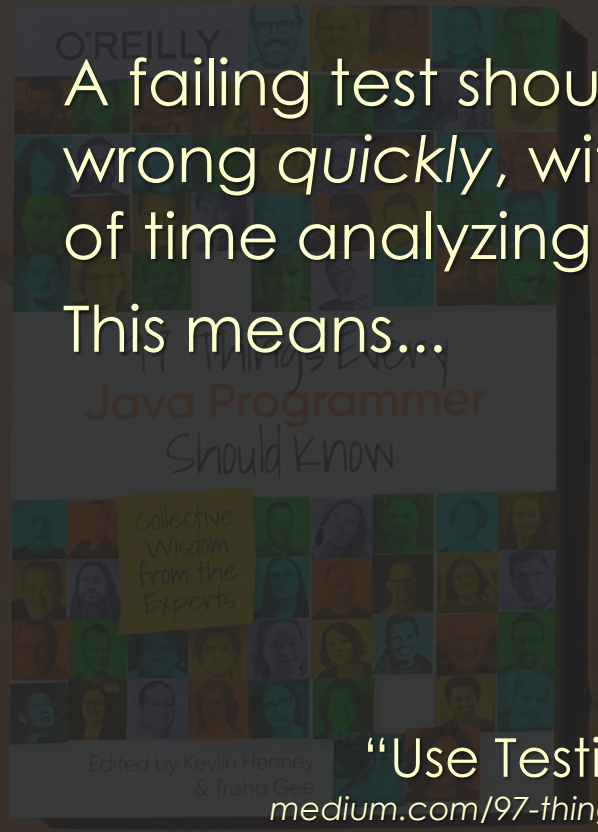
Gerard Meszaros

97-things-every-x-should-know.gitbooks.io/97-things-every-programmer-should-know/content/en/thing_95

```
namespace GildedRose_spec ...
    public class The_quality_of_a_normal_item ...
        public void decreases_by_1_every_day_before_its_sell_by_date(...)
        public void decreases_by_2_every_day_past_its_sell_by_date(...)
        public void never_decreases_below_0(...)
    public class The_quality_of_aged_brie ...
        public void increases_by_1_every_day_before_its_sell_by_date(...)
        public void increases_by_2_past_its_sell_by_date(...)
        public void never_increases_beyond_50(...)
    public class The_quality_of_backstage_passes ...
        public void increases_by_2_every_day_between_10_and_5_days_before_its_sell_by_date(...)
        public void increases_by_3_every_day_within_5_days_of_its_sell_by_date(...)
        public void drops_to_0_after_its_sell_by_date(...)
        public void never_increases_beyond_50(...)
    public class The_quality_of_sulfuras ...
        public void never_changes_and_never_has_to_be_sold(...)
    public class The_quality_of_any_item ...
        public void changes_independently_of_other_items()
```

```
namespace GildedRose_spec ...
    public class The_quality_of_a_normal_item ...
        public void decreases_by_1_every_day_before_its_sell_by_date(...)
        public void decreases_by_2_every_day_past_its_sell_by_date(...)
        public void never_decreases_below_0(...)
    public class The_quality_of_aged_brie ...
        public void increases_by_1_every_day_before_its_sell_by_date(...)
        public void increases_by_2_past_its_sell_by_date(...)
        public void never_increases_beyond_50(...)
    public class The_quality_of_backstage_passes ...
        public void increases_by_2_every_day_between_10_and_5_days_before_its_sell_by_date(...)
        public void increases_by_3_every_day_within_5_days_of_its_sell_by_date(...)
        public void drops_to_0_after_its_sell_by_date(...)
        public void never_increases_beyond_50(...)
    public class The_quality_of_sulfuras ...
        public void never_changes_and_never_has_to_be_sold(...)
    public class The_quality_of_any_item ...
        public void changes_independently_of_other_items()
```


O'REILLY®
97 Things Every
Java Programmer
Should Know
Collective Wisdom from the Experts
Edited by Kevlin Henney
& Trisha Gee



A failing test should tell you *exactly* what is wrong *quickly*, without you having to spend a lot of time analyzing the failure.

This means...

Marit van Dijk

"Use Testing to Develop Better Software Faster"

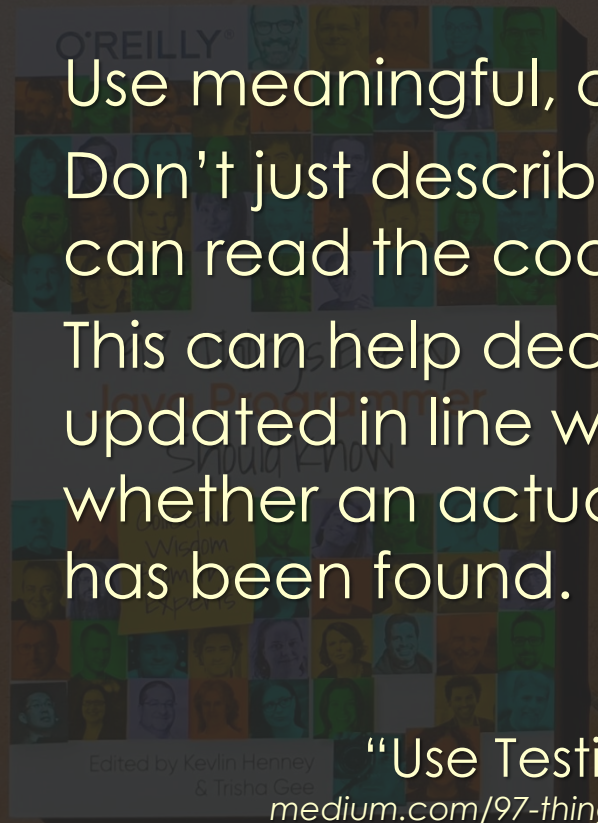
medium.com/97-things/use-testing-to-develop-better-software-faster-9dd2616543d3

Each test should test one thing.

Marit van Dijk

"Use Testing to Develop Better Software Faster"

medium.com/97-things/use-testing-to-develop-better-software-faster-9dd2616543d3



Use meaningful, descriptive names.

Don't just describe what the test does either (we can read the code), tell us why it does this.

This can help decide whether a test should be updated in line with changed functionality or whether an actual failure that should be fixed has been found.

Marit van Dijk

"Use Testing to Develop Better Software Faster"

medium.com/97-things/use-testing-to-develop-better-software-faster-9dd2616543d3

Never trust a test you haven't seen fail.

Marit van Dijk

"Use Testing to Develop Better Software Faster"

medium.com/97-things/use-testing-to-develop-better-software-faster-9dd2616543d3

```
public class The_quality_of_a_normal_item
{
    [TestCase("+5 Dexterity Vest", 10, 20)]
    [TestCase("Elixir of the Mongoose", 5, 7)]
    public void decreases_by_1_every_day_before_its_sell_by_date(string name, int sellIn, int quality)
    ...
    [TestCase("+5 Dexterity Vest", 0, 10)]
    [TestCase("Elixir of the Mongoose", -1, 2)]
    public void decreases_by_2_every_day_past_its_sell_by_date(string name, int sellIn, int quality)
    ...
    [TestCase("+5 Dexterity Vest", 1, 0)]
    [TestCase("Elixir of the Mongoose", -5, 0)]
    public void never_decreases_below_0(string name, int sellIn, int quality)
    ...
}
```

```
public class The_quality_of_a_normal_item
{
    [TestCase("+5 Dexterity Vest", 10, 20)]
    [TestCase("Elixir of the Mongoose", 5, 7)]
    public void decreases_by_1_every_day_before_its_sell_by_date(string name, int sellIn, int quality)
    ...
    [TestCase("+5 Dexterity Vest", 0, 10)]
    [TestCase("Elixir of the Mongoose", -1, 2)]
    public void decreases_by_2_every_day_past_its_sell_by_date(string name, int sellIn, int quality)
    ...
    [TestCase("+5 Dexterity Vest", 1, 0)]
    [TestCase("Elixir of the Mongoose", -5, 0)]
    public void never_decreases_below_0(string name, int sellIn, int quality)
    ...
}
```

```
public class The_quality_of_a_normal_item
{
    [TestCase("+5 Dexterity Vest", 10, 20)]
    [TestCase("Elixir of the Mongoose", 5, 7)]
    [TestCase("Morning Potion", 1, 1)]
    [TestCase("Sour Bread with Rye Humour and Cold Vengeance", 3, 5)]
    public void decreases_by_1_every_day_before_its_sell_by_date(string name, int sellIn, int quality)
    ...
    [TestCase("+5 Dexterity Vest", 0, 10)]
    [TestCase("Elixir of the Mongoose", -1, 2)]
    [TestCase("Morning Potion", 0, 2)]
    [TestCase("Sour Bread with Rye Humour and Cold Vengeance", -2, 2)]
    public void decreases_by_2_every_day_past_its_sell_by_date(string name, int sellIn, int quality)
    ...
    [TestCase("+5 Dexterity Vest", 1, 0)]
    [TestCase("Elixir of the Mongoose", -5, 0)]
    [TestCase("Morning Potion", 0, 1)]
    [TestCase("Sour Bread with Rye Humour and Cold Vengeance", -2, 1)]
    public void never_decreases_below_0(string name, int sellIn, int quality)
    ...
}
```

```
public class The_quality_of_a_normal_item
{
    [TestCase("+5 Dexterity Vest", 10, 20)]
    [TestCase("Elixir of the Mongoose", 5, 7)]
    [TestCase("Morning Potion", 1, 1)]
    [TestCase("Sour Bread with Rye Humour and Cold Vengeance", 3, 5)]
    public void decreases_by_1_every_day_before_its_sell_by_date(string name, int sellIn, int quality)
    {
        var typicalItem = new Item {Name = name, SellIn = sellIn, Quality = quality};
        var gildedRose = new GildedRose() {Items = new List<Item> {typicalItem}};

        gildedRose.UpdateQuality();

        Assert.AreEqual(sellIn - 1, typicalItem.SellIn);
        Assert.AreEqual(quality - 1, typicalItem.Quality);
    }
    ...
    public void decreases_by_2_every_day_past_its_sell_by_date(string name, int sellIn, int quality)
    ...
    public void never_decreases_below_0(string name, int sellIn, int quality)
    ...
}
```

3As structuring of
test-case narrative

Arrange

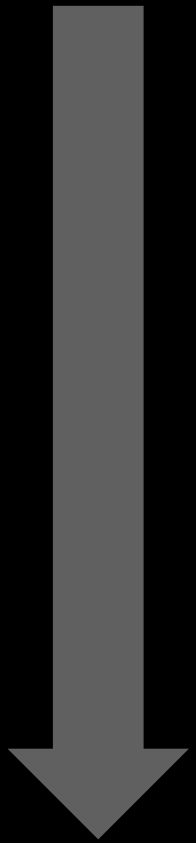
Set up initial state and objects involved in the test

Act

Perform the operation that is the central focus of the test case

Assert

Assert on the expected values and outcomes of performing the test



BDD structuring of
scenarios and tests

Given

... a particular set-up and initial state...

When

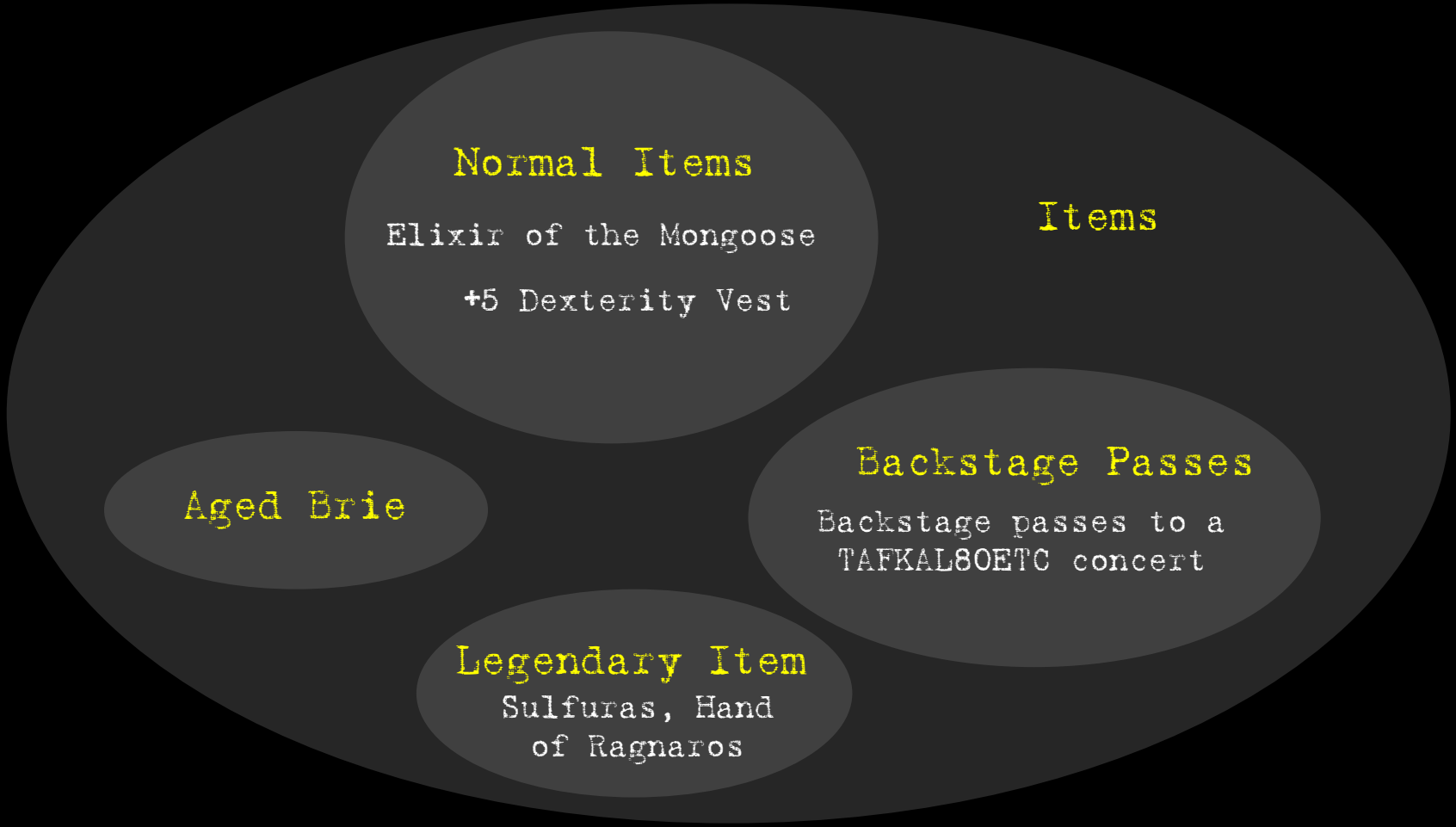
... a particular operation is performed...

Then

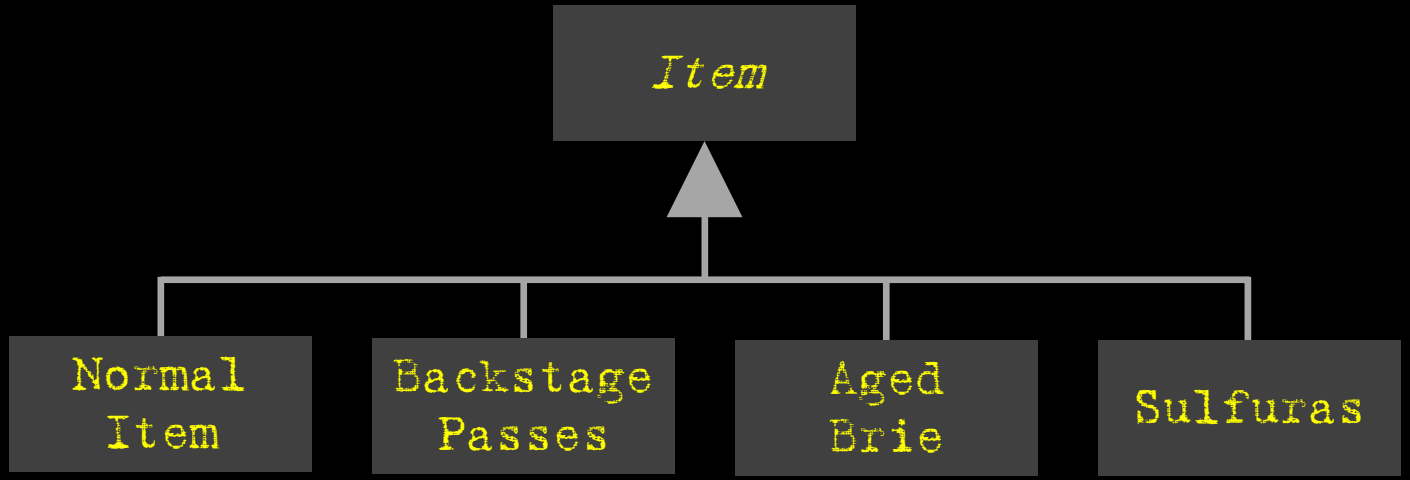
... a particular outcome is expected

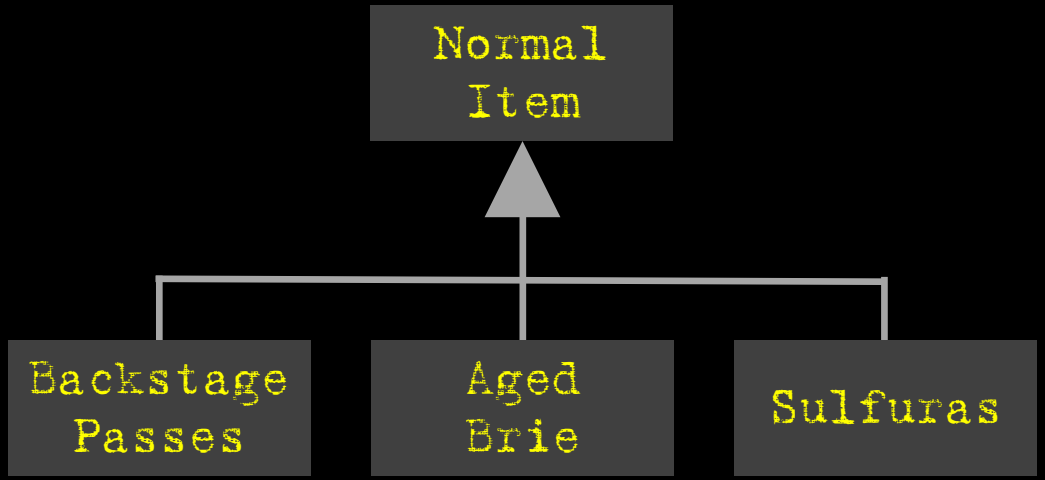
```
public class The_quality_of_any_item
{
    [Test]
    public void changes_independently_of_other_items()
    {
        var dexterityVest = new Item {Name = "+5 Dexterity Vest", SellIn = 10, Quality = 20};
        var agedBrie = new Item {Name = "Aged Brie", SellIn = 2, Quality = 0};
        var elixir = new Item {Name = "Elixir of the Mongoose", SellIn = 5, Quality = 7};
        var sulfuras = new Item {Name = "Sulfuras, Hand of Ragnaros", SellIn = 0, Quality = 80};
        var backstagePasses = new Item {Name = "Backstage passes to a TAFKAL80ETC concert",  SellIn = 15, Quality = 20};
        var gildedRose = new GildedRose()
        {
            Items = new List<Item> {dexterityVest, agedBrie, elixir, sulfuras, backstagePasses}
        };

        gildedRose.UpdateQuality();

        Assert.AreEqual(9, dexterityVest.SellIn);
        Assert.AreEqual(19, dexterityVest.Quality);
        Assert.AreEqual(1, agedBrie.SellIn);
        Assert.AreEqual(1, agedBrie.Quality);
        Assert.AreEqual(4, elixir.SellIn);
        Assert.AreEqual(6, elixir.Quality);
        Assert.AreEqual(0, sulfuras.SellIn);
        Assert.AreEqual(80, sulfuras.Quality);
        Assert.AreEqual(14, backstagePasses.SellIn);
        Assert.AreEqual(21, backstagePasses.Quality);
    }
}
```

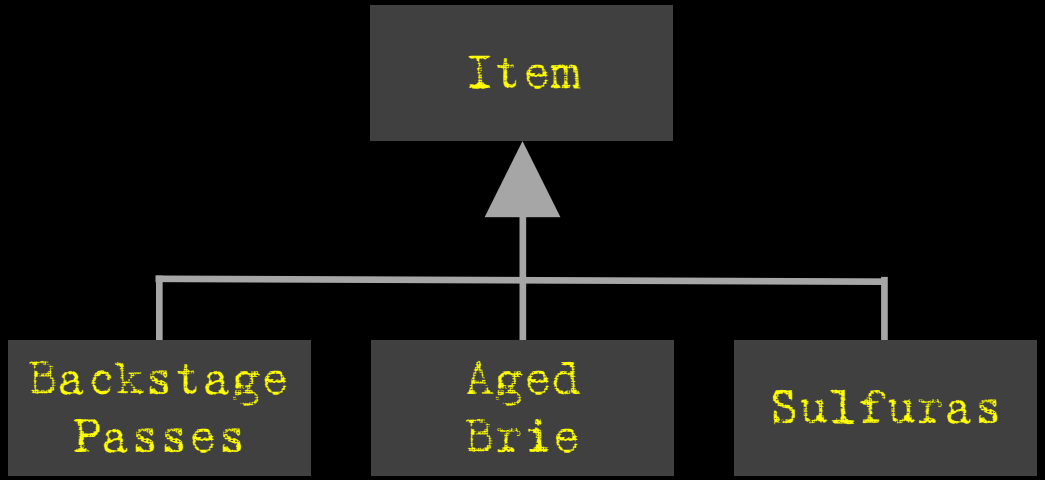

Items
Normal Items
Elixir of the Mongoose
+5 Dexterity Vest
Aged Brie
Backstage Passes
Backstage passes to a TAFKAL80ETC concert
Sulfuras

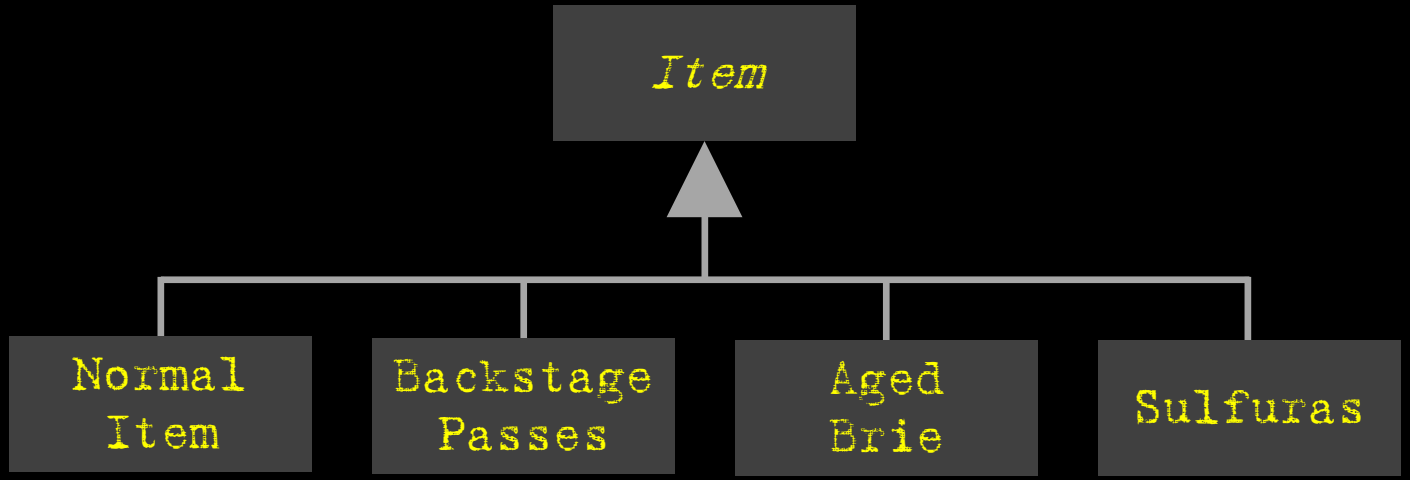

Items
Normal Items
Elixir of the Mongoose
+5 Dexterity Vest
Aged Brie
Backstage Passes
Backstage passes to a TAFKAL80ETC concert
Legendary Item
Sulfuras, Hand of Ragnaros


Items
Normal Items
Elixir of the Mongoose
+5 Dexterity Vest
Aged Brie
Backstage Passes
Backstage passes to a TAFKAL80ETC concert
Sulfuras


Item
Normal Item
Backstage Passes
Aged Brie
Sulfuras

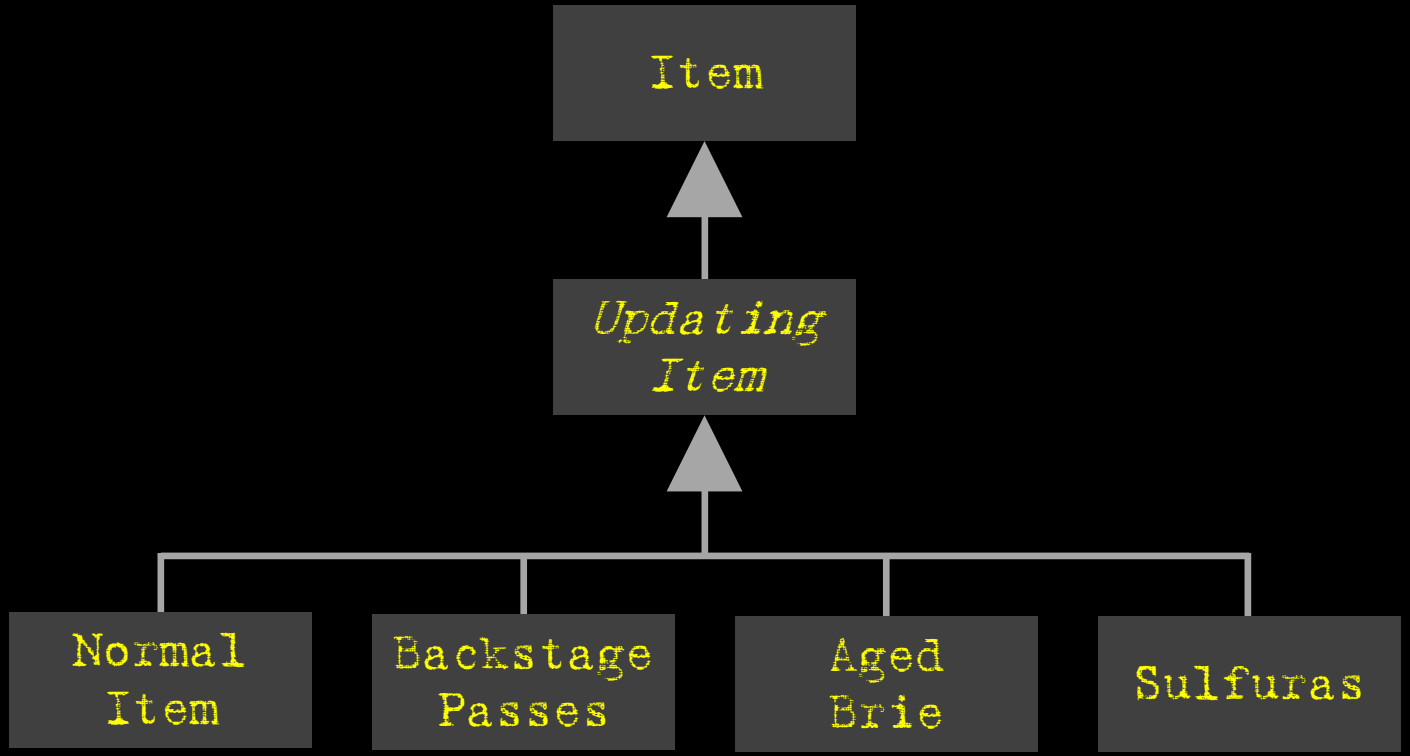

Normal Item
Backstage Passes
Aged Brie
Sulfuras


Item
Backstage Passes
Aged Brie
Sulfuras


Item
Normal Item
Backstage Passes
Aged Brie
Sulfuras

Goblin Rule

Do not alter the Item class or Items property as those belong to the goblin in the corner who will insta-rage and one-shot you as he doesn't believe in shared code ownership.


Item
Updating Item
Normal Item
Backstage Passes
Aged Brie
Sulfuras

I have yet to see any problem, however complicated, which, when you looked at it in the right way, did not become still more complicated.

Anderson's Law

```
public class GildedRose
{
    public IList<Item> Items;

    public void UpdateQuality()
    {
        for (var i = 0; i < Items.Count; i++)
        {
            if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;

                    if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].SellIn < 11)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }

                        if (Items[i].SellIn < 6)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }
                    }
                }
            }

            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].SellIn = Items[i].SellIn - 1;
            }

            if (Items[i].SellIn < 0)
            {
                if (Items[i].Name != "Aged Brie")
                {
                    if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].Quality > 0)
                        {
                            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                            {
                                Items[i].Quality = Items[i].Quality - 1;
                            }
                        }
                    }
                    else
                    {
                        Items[i].Quality = Items[i].Quality - Items[i].Quality;
                    }
                }
                else
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    public class Item
    {
        public string Name { get; set; }

        public int SellIn { get; set; }

        public int Quality { get; set; }
    }
}
```

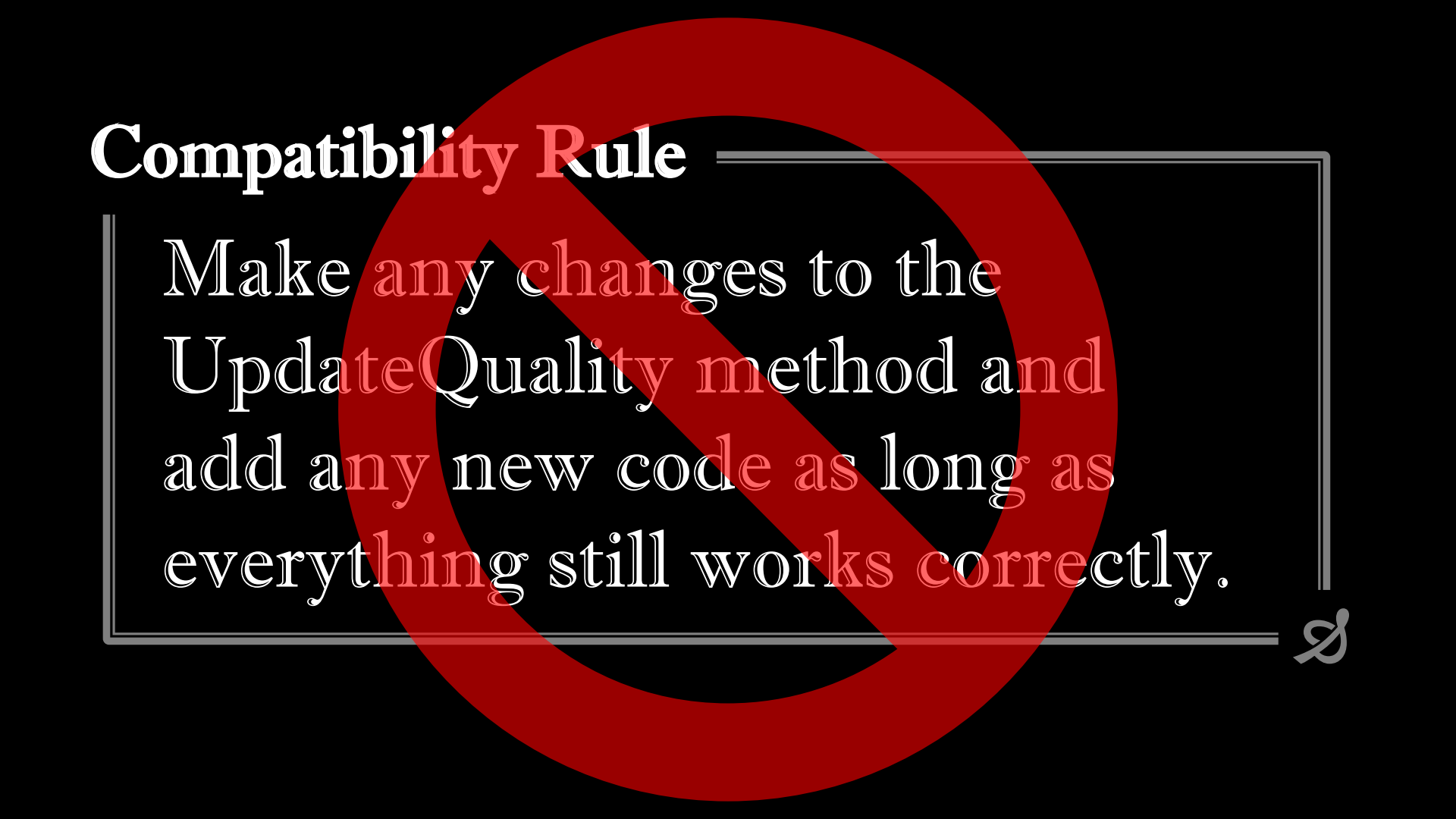
```
public class GildedRose
{
    public IList<Item> Items;

    public void UpdateQuality()
    {
        foreach (var item in Items)
        {
            item.Update();
        }
    }

    public class Item
    {
        public string Name { get; set; }

        public int SellIn { get; set; }

        public int Quality { get; set; }
    }

    public void AddItem(string name, int sellIn, int quality)
    {
        if (name == "Aged Brie")
        {
            Items.Add(new AgedBrie {Name = name, SellIn = sellIn, Quality = quality});
        }
        else if (name == "Backstage passes to a TAFKAL80ETC concert")
        {
            Items.Add(new BackstagePasses {Name = name, SellIn = sellIn, Quality = quality});
        }
        else if (name == "Sulfuras, Hand of Ragnaros")
        {
            Items.Add(new Sulfuras {Name = name, SellIn = sellIn, Quality = quality});
        }
        else
        {
            Items.Add(new NormalItem {Name = name, SellIn = sellIn, Quality = quality});
        }
    }
}

public abstract class UpdatingItem : GildedRose.Item
{
    public abstract void Update();
}

public class NormalItem : UpdatingItem
{
    public override void Update()
    {
        return SellIn-- > 0 ? -1 : -2
    }
}

public class AgedBrie : UpdatingItem
{
    public override void Update()
    {
        return SellIn-- > 0 ? 1 : 2
    }
}

public class BackstagePasses : UpdatingItem
{
    public override void Update()
    {
        --SellIn;

        if (SellIn >= 10)
            return 1;
        else if (SellIn >= 5)
            return 2;
        else if (SellIn >= 0)
            return 3;
        else
            return -Quality;
    }
}

public class Sulfuras : UpdatingItem
{
    public override void Update()
    {
    }
}
```

```
public class GildedRose
{
    public IList<Item> Items;

    public void UpdateQuality()
    {
        for (var i = 0; i < Items.Count; i++)
        {
            if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (Items[i].Quality > 0)
                {
                    if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                    {
                        Items[i].Quality = Items[i].Quality - 1;
                    }
                }
            }
            else
            {
                if (Items[i].Quality < 50)
                {
                    Items[i].Quality = Items[i].Quality + 1;

                    if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].SellIn < 11)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }

                        if (Items[i].SellIn < 6)
                        {
                            if (Items[i].Quality < 50)
                            {
                                Items[i].Quality = Items[i].Quality + 1;
                            }
                        }
                    }
                }
            }

            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].SellIn = Items[i].SellIn - 1;
            }

            if (Items[i].SellIn < 0)
            {
                if (Items[i].Name != "Aged Brie")
                {
                    if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert")
                    {
                        if (Items[i].Quality > 0)
                        {
                            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
                            {
                                Items[i].Quality = Items[i].Quality - 1;
                            }
                        }
                    }
                    else
                    {
                        Items[i].Quality = Items[i].Quality - Items[i].Quality;
                    }
                }
                else
                {
                    if (Items[i].Quality < 50)
                    {
                        Items[i].Quality = Items[i].Quality + 1;
                    }
                }
            }
        }
    }

    public class Item
    {
        public string Name { get; set; }

        public int SellIn { get; set; }

        public int Quality { get; set; }
    }
}
```

```
public class GildedRose
{
    public IList<Item> Items;

    public void UpdateQuality()
    {
        foreach (var item in Items)
        {
            item.Update();
        }
    }

    public class Item
    {
        public string Name { get; set; }

        public int SellIn { get; set; }

        public int Quality { get; set; }
    }

    public void AddItem(string name, int sellIn, int quality)
    {
        if (name == "Aged Brie")
        {
            Items.Add(new AgedBrie {Name = name, SellIn = sellIn, Quality = quality});
        }
        else if (name == "Backstage passes to a TAFKAL80ETC concert")
        {
            Items.Add(new BackstagePasses {Name = name, SellIn = sellIn, Quality = quality});
        }
        else if (name == "Sulfuras, Hand of Ragnaros")
        {
            Items.Add(new Sulfuras {Name = name, SellIn = sellIn, Quality = quality});
        }
        else
        {
            Items.Add(new NormalItem {Name = name, SellIn = sellIn, Quality = quality});
        }
    }
}

public abstract class UpdatingItem : GildedRose.Item
{
    public abstract void Update();
}

public class NormalItem : UpdatingItem
{
    public override void Update()
    {
        return SellIn-- > 0 ? -1 : -2
    }
}

public class AgedBrie : UpdatingItem
{
    public override void Update()
    {
        return SellIn-- > 0 ? 1 : 2
    }
}

public class BackstagePasses : UpdatingItem
{
    public override void Update()
    {
        --SellIn;

        if (SellIn >= 10)
            return 1;
        else if (SellIn >= 5)
            return 2;
        else if (SellIn >= 0)
            return 3;
        else
            return -Quality;
    }
}

public class Sulfuras : UpdatingItem
{
    public override void Update()
    {
    }
}
```

Compatibility Rule

Make any changes to the UpdateQuality method and add any new code as long as everything still works correctly.



Feel Stuck? Use the Rule of 5 Little Things to Start Being More Productive, Focused, and Happier

Sometimes seeing the seemingly impenetrable forest can keep you from seeing all the trees you can easily fix.

BY JEFF HADEN, CONTRIBUTING EDITOR, INC. @JEFF_HADEN

inc.com/jeff-haden/feel-stuck-use-rule-of-5-little-things-to-start-being-more-productive-focused-happier.html

Feel Stuck? Use the Rule of 5 Little Things to Start Being More Productive, Focused, and Happier

Sometimes seeing the seemingly impenetrable forest can keep you from seeing all the trees you can easily fix.

BY JEFF HADEN, CONTRIBUTING EDITOR, INC. @JEFF_HADEN

I picked up a great tip from producer Jim Scott. I wasn't happy with a song. Couldn't articulate why. Just knew I needed to record the whole thing over again.

Joe Satriani

inc.com/jeff-haden/feel-stuck-use-rule-of-5-little-things-to-start-being-more-productive-focused-happier.html

Feel Stuck? Use the Rule of 5 Little Things to Start Being More Productive, Focused, and Happier

He said, “Pick five little things that really bother you, we’ll fix them, and then you can listen to it again.” I picked five that he didn’t think detracted from the song.

Sometimes seeing the seemingly impenetrable forest can keep you from seeing all the trees you can easily fix.

Joe Satriani

BY JEFF HADEN, CONTRIBUTING EDITOR, INC. @JEFF_HADEN

So we fixed them. Some were just a half-second long, but they bothered me. And it turned out great. That was a really good lesson.

Joe Satriani

0 Tests are neither expressive nor extensive enough

```
public class GildedRoseTests ...
    public void Example_of_existing_functionality()
```

```
namespace GildedRose_spec ...
    public class The_quality_of_a_normal_item ...
        public void decreases_by_1_every_day_before_its_sell_by_date(...)
        public void decreases_by_2_every_day_past_its_sell_by_date(...)
        public void never_decreases_below_0(...)
    public class The_quality_of_aged_brie ...
        public void increases_by_1_every_day_before_its_sell_by_date(...)
        public void increases_by_2_past_its_sell_by_date(...)
        public void never_increases_beyond_50(...)
    public class The_quality_of_backstage_passes ...
        public void increases_by_2_every_day_between_10_and_5_days_before_its_sell_by_date(...)
        public void increases_by_3_every_day_within_5_days_of_its_sell_by_date(...)
        public void drops_to_0_after_its_sell_by_date(...)
        public void never_increases_beyond_50(...)
    public class The_quality_of_sulfuras ...
        public void never_changes_and_never_has_to_be_sold(...)
    public class The_quality_of_any_item ...
        public void changes_independently_of_other_items()
```

Counting *for* loop used to iterate collection content

```
for (var i = 0; i < Items.Count; i++)
{
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstag
    {
        if (Items[i].Quality > 0)
        {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros")
            {
                Items[i].Quality = Items[i].Quality - 1;
            }
        }
    }
    ...
}
```

```
foreach (var item in Items)
{
    if (item.Name != "Aged Brie" && item.Name != "Backstage passes
    {
        if (item.Quality > 0)
        {
            if (item.Name != "Sulfuras, Hand of Ragnaros")
            {
                item.Quality = item.Quality - 1;
            }
        }
    }
    ...
}
```

2 Non-idiomatic arithmetic

```
foreach (var item in Items)
{
    if (item.Name != "Aged Brie" && item.Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.Quality > 0)
        {
            if (item.Name != "Sulfuras, Hand of Ragnaros")
            {
                item.Quality = item.Quality - 1;
            }
        }
    }
    else
    {
        if (item.Quality < 50)
        {
            item.Quality = item.Quality + 1;

            if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (item.SellIn < 11)
                {
                    if (item.Quality < 50)
                    {
                        item.Quality = item.Quality + 1;
                    }
                }

                if (item.SellIn < 6)
                {
                    if (item.Quality < 50)
                    {
                        item.Quality = item.Quality + 1;
                    }
                }
            }
        }
    }

    if (item.Name != "Sulfuras, Hand of Ragnaros")
    {
        item.SellIn = item.SellIn - 1;
    }

    if (item.SellIn < 0)
    {
        if (item.Name != "Aged Brie")
        {
            if (item.Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (item.Quality > 0)
                {
                    if (item.Name != "Sulfuras, Hand of Ragnaros")
                    {
                        item.Quality = item.Quality - 1;
                    }
                }
            }
            else
            {
                item.Quality = item.Quality - item.Quality;
            }
        }
        else
        {
            if (item.Quality < 50)
            {
                item.Quality = item.Quality + 1;
            }
        }
    }
}
```

```
...
if (item.SellIn < 0)
{
    if (item.Name != "Aged Brie")
    {
        if (item.Name != "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.Quality > 0)
            {
                if (item.Name != "Sulfuras, Hand of Ragnaros")
                {
                    item.Quality = item.Quality - 1;
                }
            }
        }
        else
        {
            item.Quality = item.Quality - item.Quality;
        }
    }
    else
    {
        if (item.Quality < 50)
        {
            item.Quality = item.Quality + 1;
        }
    }
}
...
```

```
...
if (item.SellIn < 0)
{
    if (item.Name != "Aged Brie")
    {
        if (item.Name != "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.Quality > 0)
            {
                if (item.Name != "Sulfuras, Hand of Ragnaros")
                {
                    --item.Quality;
                }
            }
        }
        else
        {
            item.Quality = 0;
        }
    }
    else
    {
        if (item.Quality < 50)
        {
            ++item.Quality;
        }
    }
}
...
```

```
foreach (var item in Items)
{
    if (item.Name != "Aged Brie" && item.Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.Quality > 0)
        {
            if (item.Name != "Sulfuras, Hand of Ragnaros")
            {
                --item.Quality;
            }
        }
    }
    else
    {
        if (item.Quality < 50)
        {
            ++item.Quality;

            if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (item.SellIn < 11)
                {
                    if (item.Quality < 50)
                    {
                        ++item.Quality;
                    }
                }

                if (item.SellIn < 6)
                {
                    if (item.Quality < 50)
                    {
                        ++item.Quality;
                    }
                }
            }
        }
    }

    if (item.Name != "Sulfuras, Hand of Ragnaros")
    {
        --item.SellIn;
    }

    if (item.SellIn < 0)
    {
        if (item.Name != "Aged Brie")
        {
            if (item.Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (item.Quality > 0)
                {
                    if (item.Name != "Sulfuras, Hand of Ragnaros")
                    {
                        --item.Quality;
                    }
                }
            }
            else
            {
                item.Quality = 0;
            }
        }
        else
        {
            if (item.Quality < 50)
            {
                ++item.Quality;
            }
        }
    }
}
```

3 Clumsy conditional logic and structure

```
foreach (var item in Items)
{
    if (item.Name != "Aged Brie" && item.Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.Quality > 0)
        {
            if (item.Name != "Sulfuras, Hand of Ragnaros")
            {
                --item.Quality;
            }
        }
    }
    else
    {
        if (item.Quality < 50)
        {
            ++item.Quality;

            if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
            {
                if (item.SellIn < 11)
                {
                    if (item.Quality < 50)
                    {
                        ++item.Quality;
                    }
                }

                if (item.SellIn < 6)
                {
                    if (item.Quality < 50)
                    {
                        ++item.Quality;
                    }
                }
            }
        }
    }

    if (item.Name != "Sulfuras, Hand of Ragnaros")
    {
        --item.SellIn;
    }

    if (item.SellIn < 0)
    {
        if (item.Name != "Aged Brie")
        {
            if (item.Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (item.Quality > 0)
                {
                    if (item.Name != "Sulfuras, Hand of Ragnaros")
                    {
                        --item.Quality;
                    }
                }
            }
            else
            {
                item.Quality = 0;
            }
        }
        else
        {
            if (item.Quality < 50)
            {
                ++item.Quality;
            }
        }
    }
}
```

```
if (item.Name != "Aged Brie" && item.Name != "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.Quality > 0)
    {
        if (item.Name != "Sulfuras, Hand of Ragnaros")
        {
            --item.Quality;
        }
    }
}
else
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.SellIn < 11)
            {
                if (item.Quality < 50)
                {
                    ++item.Quality;
                }
            }

            if (item.SellIn < 6)
            {
                if (item.Quality < 50)
                {
                    ++item.Quality;
                }
            }
        }
    }
}
```

```
if (item.Name != "Aged Brie" && item.Name != "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.Quality > 0)
        if (item.Name != "Sulfuras, Hand of Ragnaros")
            --item.Quality;
}
else
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.SellIn < 11)
                if (item.Quality < 50)
                    ++item.Quality;

            if (item.SellIn < 6)
                if (item.Quality < 50)
                    ++item.Quality;
        }
    }
}
```

De Morgan's Laws

$$\neg(P \vee Q) \Leftrightarrow (\neg P) \wedge (\neg Q)$$
$$\neg(P \wedge Q) \Leftrightarrow (\neg P) \vee (\neg Q)$$

```
if (!(item.Name == "Aged Brie" || item.Name == "Backstage passes to a TAFKAL80ETC concert"))
{
    if (item.Quality > 0)
        if (item.Name != "Sulfuras, Hand of Ragnaros")
            --item.Quality;
}
else
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.SellIn < 11)
                if (item.Quality < 50)
                    ++item.Quality;

            if (item.SellIn < 6)
                if (item.Quality < 50)
                    ++item.Quality;
        }
    }
}
```

```
if (item.Name == "Aged Brie" || item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.SellIn < 11)
                if (item.Quality < 50)
                    ++item.Quality;

            if (item.SellIn < 6)
                if (item.Quality < 50)
                    ++item.Quality;
        }
    }
}
else
{
    if (item.Quality > 0)
        if (item.Name != "Sulfuras, Hand of Ragnaros")
            --item.Quality;
}
```

```
if (item.Name == "Aged Brie" || item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.SellIn < 11)
                if (item.Quality < 50)
                    ++item.Quality;

            if (item.SellIn < 6)
                if (item.Quality < 50)
                    ++item.Quality;
        }
    }
}
else
{
    if (item.Name != "Sulfuras, Hand of Ragnaros")
        if (item.Quality > 0)
            --item.Quality;
}
```

```
if (item.Name == "Aged Brie" || item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.SellIn < 11)
                if (item.Quality < 50)
                    ++item.Quality;

            if (item.SellIn < 6)
                if (item.Quality < 50)
                    ++item.Quality;
        }
    }
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    if (item.Quality > 0)
        --item.Quality;
}
```

```
if (item.Name == "Aged Brie")
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.SellIn < 11)
                if (item.Quality < 50)
                    ++item.Quality;

            if (item.SellIn < 6)
                if (item.Quality < 50)
                    ++item.Quality;
        }
    }
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.SellIn < 11)
                if (item.Quality < 50)
                    ++item.Quality;

            if (item.SellIn < 6)
                if (item.Quality < 50)
                    ++item.Quality;
        }
    }
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
```

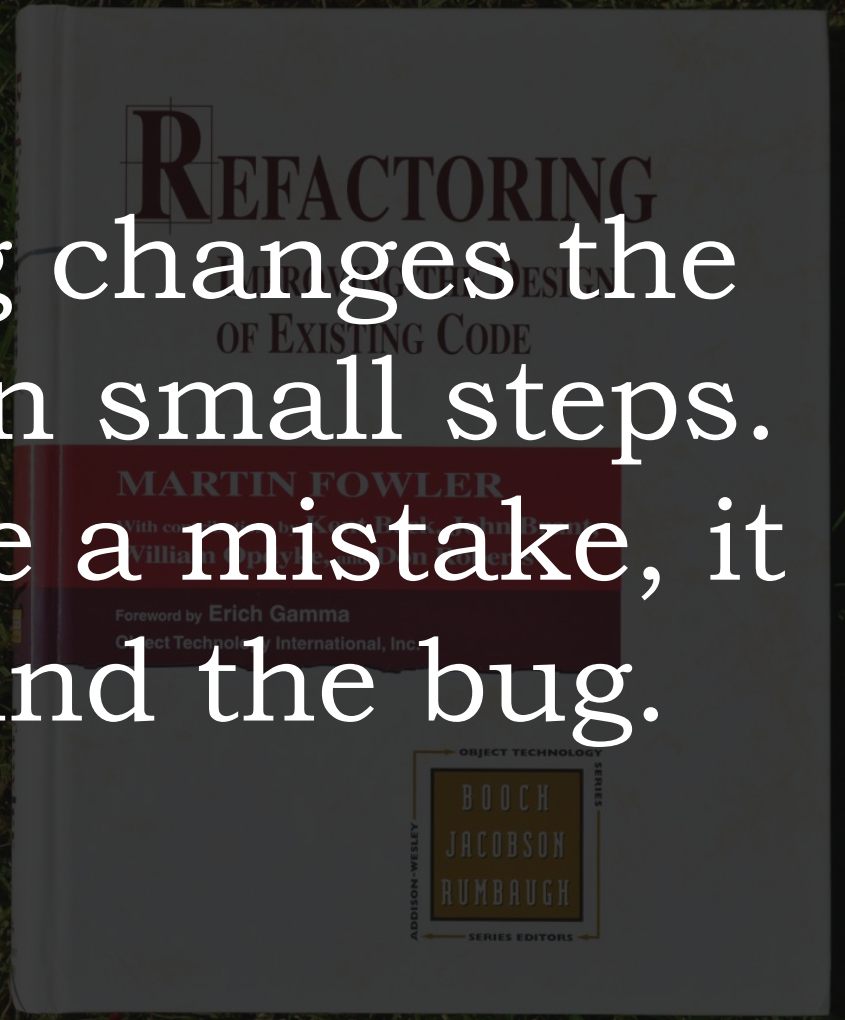

if (item.Name == "Aged Brie")
{
if (item.Quality < 50)
{
++item.Quality;
if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
if (item.SellIn < 11)
if (item.Quality < 50)
++item.Quality;
if (item.SellIn < 6)
if (item.Quality < 50)
++item.Quality;
}
}
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
if (item.Quality < 50)
{
++item.Quality;
if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
if (item.SellIn < 11)
if (item.Quality < 50)
++item.Quality;
if (item.SellIn < 6)
if (item.Quality < 50)
++item.Quality;
}
}
}
DRY?

```
if (item.Name == "Aged Brie")
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.SellIn < 11)
                if (item.Quality < 50)
                    ++item.Quality;

            if (item.SellIn < 6)
                if (item.Quality < 50)
                    ++item.Quality;
        }
    }
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.SellIn < 11)
                if (item.Quality < 50)
                    ++item.Quality;

            if (item.SellIn < 6)
                if (item.Quality < 50)
                    ++item.Quality;
        }
    }
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
```

WET!

```
if (item.Name == "Aged Brie")
{
    if (item.Quality < 50)
        ++item.Quality;
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.SellIn < 11)
            if (item.Quality < 50)
                ++item.Quality;

        if (item.SellIn < 6)
            if (item.Quality < 50)
                ++item.Quality;
    }
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    if (item.Quality > 0)
        --item.Quality;
}
```

Refactoring changes the programs in small steps. If you make a mistake, it is easy to find the bug.



```
foreach (var item in Items)
{
    if (item.Name == "Aged Brie")
    {
        if (item.Quality < 50)
            ++item.Quality;
    }
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.Quality < 50)
        {
            ++item.Quality;

            if (item.SellIn < 11)
                if (item.Quality < 50)
                    ++item.Quality;

            if (item.SellIn < 6)
                if (item.Quality < 50)
                    ++item.Quality;
        }
    }
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
    {
        if (item.Quality > 0)
            --item.Quality;
    }

    if (item.Name != "Sulfuras, Hand of Ragnaros")
    {
        --item.SellIn;
    }

    if (item.SellIn < 0)
    {
        if (item.Name != "Aged Brie")
        {
            if (item.Name != "Backstage passes to a TAFKAL80ETC concert")
            {
                if (item.Quality > 0)
                {
                    if (item.Name != "Sulfuras, Hand of Ragnaros")
                    {
                        --item.Quality;
                    }
                }
            }
            else
            {
                item.Quality = 0;
            }
        }
        else
        {
            if (item.Quality < 50)
            {
                ++item.Quality;
            }
        }
    }
}
```

```
if (item.SellIn < 0)
{
    if (item.Name != "Aged Brie")
    {
        if (item.Name != "Backstage passes to a TAFKAL80ETC concert")
        {
            if (item.Quality > 0)
            {
                if (item.Name != "Sulfuras, Hand of Ragnaros")
                {
                    --item.Quality;
                }
            }
        }
        else
        {
            item.Quality = 0;
        }
    }
    else
    {
        if (item.Quality < 50)
        {
            ++item.Quality;
        }
    }
}
```

```
if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
    {
        if (item.Quality < 50)
            ++item.Quality;
    }
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {
        item.Quality = 0;
    }
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
    {
        if (item.Quality > 0)
            --item.Quality;
    }
}
```

```
foreach (var item in Items)
{
    if (item.Name == "Aged Brie")&& item.Name != "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.Quality < 50)
        {   ++item.Quality;
    }       if (item.Name != "Sulfuras, Hand of Ragnaros")
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {           --item.Quality;
        if (item.Quality < 50)
        {
    }       ++item.Quality;
    else
    {       if (item.SellIn < 11)
        if (item.Name      if (item.Quality < 50)
        {               ++item.Quality;
        ++item.Quality;
        if (item.SellIn < 6)
        if (item.Name       if (item.Quality < 50)
        {       ++item.Quality;
        }       if (item.SellIn < 11)
        {
    }               {
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
    {               {
        if (item.Quality > 0)       ++item.Quality;
        --item.Quality;
    }               }

    if (item.Name != "Sulfuras, Hand of Ragnaros")
    {               {
        --item.SellIn; (item.Quality < 50)
    }               {
                        ++item.Quality;
    if (item.SellIn < 0)
    {               }
        if (item.Name == "Aged Brie")
        {
    }       if (item.Quality < 50)
            ++item.Quality;
    if (item.Name != "Sulfuras, Hand of Ragnaros")
    {   else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        {--item.SellIn;
    }       item.Quality = 0;
        }
    if (item.Name != "Sulfuras, Hand of Ragnaros")
    {   else if (item.Name != "Sulfuras, Hand of Ragnaros")
        {
        if (item.Name == "Aged Brie")
        {       if (item.Quality > 0)
            --item.Quality;
        }   if (item.Name != "Backstage passes to a TAFKAL80ETC concert")
        {
    }       if (item.Quality > 0)
            {
                if (item.Name != "Sulfuras, Hand of Ragnaros")
                {
                    --item.Quality;
                }
            }
        }
        else
        {
            item.Quality = 0;
        }
    }
    else
    {
        if (item.Quality < 50)
        {
            ++item.Quality;
        }
    }
}
```

4

Limiting of *Quality* value is verbose

```
if (item.Name == "Aged Brie")
{
    if (item.Quality < 50)
        ++item.Quality;
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.SellIn < 11)
            if (item.Quality < 50)
                ++item.Quality;

        if (item.SellIn < 6)
            if (item.Quality < 50)
                ++item.Quality;
    }
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    if (item.Quality > 0)
        --item.Quality;
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
    {
        if (item.Quality < 50)
            ++item.Quality;
    }
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {
        item.Quality = 0;
    }
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
    {
        if (item.Quality > 0)
            --item.Quality;
    }
}
```

```
if (item.Name == "Aged Brie")
{
    if (item.Quality < 50)
        ++item.Quality;
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.Quality < 50)
    {
        ++item.Quality;

        if (item.SellIn < 11)
            if (item.Quality < 50)
                ++item.Quality;

        if (item.SellIn < 6)
            if (item.Quality < 50)
                ++item.Quality;
    }
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    if (item.Quality > 0)
        --item.Quality;
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
    {
        if (item.Quality < 50)
            ++item.Quality;
    }
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {
        item.Quality = 0;
    }
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
    {
        if (item.Quality > 0)
            --item.Quality;
    }
}
```

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    item.Quality = Min(item.Quality + 1, 50);

    if (item.SellIn < 11)
        item.Quality = Min(item.Quality + 1, 50);

    if (item.SellIn < 6)
        item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

5

Quality is repeatedly readjusted for backstage passes

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    item.Quality = Min(item.Quality + 1, 50);

    if (item.SellIn < 11)
        item.Quality = Min(item.Quality + 1, 50);

    if (item.SellIn < 6)
        item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    item.Quality = Min(item.Quality + 1, 50);

    if (item.SellIn < 11)
        item.Quality = Min(item.Quality + 1, 50);

    if (item.SellIn < 6)
        item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    int qualityAdjustment = 1;

    if (item.SellIn < 11)
        ++qualityAdjustment;

    if (item.SellIn < 6)
        ++qualityAdjustment;

    item.Quality = Min(item.Quality + qualityAdjustment, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    int qualityAdjustment;

    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else
        qualityAdjustment = 3;

    item.Quality = Min(item.Quality + qualityAdjustment, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    int qualityAdjustment =
        if (item.SellIn > 10)
            1
        else if (item.SellIn > 5)
            2
        else
            3;

    item.Quality = Min(item.Quality + qualityAdjustment, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    int qualityAdjustment =
        if (item.SellIn > 10)
            1
        else if (item.SellIn > 5)
            2
        else
            3;

    item.Quality = Min(item.Quality + qualityAdjustment, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    int qualityAdjustment =
        item.SellIn > 10 ?
            1
        : item.SellIn > 5 ?
            2
        :
            3;

    item.Quality = Min(item.Quality + qualityAdjustment, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    int qualityAdjustment = (() =>
                            {
                                if (item.SellIn > 10)
                                    return 1;
                                else if (item.SellIn > 5)
                                    return 2;
                                else
                                    return 3;
                            })();
    item.Quality = Min(item.Quality + qualityAdjustment, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    int qualityAdjustment = (() =>
                            {
                                if (item.SellIn > 10)
                                    return 1;
                                else if (item.SellIn > 5)
                                    return 2;
                                else
                                    return 3;
                            })();
    item.Quality = Min(item.Quality + qualityAdjustment, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    Func<int> qualityAdjustment = () =>
                                  {
                                      if (item.SellIn > 10)
                                          return 1;
                                      else if (item.SellIn > 5)
                                          return 2;
                                      else
                                          return 3;
                                  };
    item.Quality = Min(item.Quality + qualityAdjustment(), 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

Developers are drawn to complexity like moths to a flame, often with the same outcome.

Neal Ford

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    int qualityAdjustment;

    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else
        qualityAdjustment = 3;

    item.Quality = Min(item.Quality + qualityAdjustment, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

- ...
- “Aged Brie” actually increases in Quality the older it gets
- ...
- “Backstage passes”, like aged brie, increases in Quality as it’s SellIn value approaches; Quality increases by 2 when there are 10 days or less and by 3 when there are 5 days or less...

Feel Stuck? Use the Rule of 5 Little Things to Start Being More Productive, Focused, and Happier

Sometimes seeing the seemingly impenetrable forest can keep you from seeing all the trees you can easily fix.

BY JEFF HADEN, CONTRIBUTING EDITOR, INC. @JEFF_HADEN

Instead of starting over, fix a few things that bug you. Often you'll find those little fixes totally change your perception of the whole.

Joe Satriani

inc.com/jeff-haden/feel-stuck-use-rule-of-5-little-things-to-start-being-more-productive-focused-happier.html

6

Generalise use of *qualityAdjustment*

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    int qualityAdjustment;

    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else
        qualityAdjustment = 3;

    item.Quality = Min(item.Quality + qualityAdjustment, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    int qualityAdjustment;

    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else
        qualityAdjustment = 3;

    item.Quality = Min(item.Quality + qualityAdjustment, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else
        qualityAdjustment = 3;

    item.Quality = Min(item.Quality + qualityAdjustment, 50);
}
else if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    item.Quality = Min(item.Quality + 1, 50);
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else
        qualityAdjustment = 3;

    item.Quality = Min(item.Quality + qualityAdjustment, 50);
}
else if (item.Name == "Sulfuras, Hand of Ragnaros")
{
}
else
{
    item.Quality = Max(item.Quality - 1, 0);
}

if (item.Name != "Sulfuras, Hand of Ragnaros")
{
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    qualityAdjustment = 1;
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else
        qualityAdjustment = 3;
}
else if (item.Name == "Sulfuras, Hand of Ragnaros")
{
    qualityAdjustment = 0;
}
else
{
    qualityAdjustment = -1;
}

if (qualityAdjustment != 0)
{
    item.Quality = Max(Min(item.Quality + qualityAdjustment, 50), 0);
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    qualityAdjustment = 1;
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else
        qualityAdjustment = 3;
}
else if (item.Name == "Sulfuras, Hand of Ragnaros")
{
    qualityAdjustment = 0;
}
else
{
    qualityAdjustment = -1;
}

if (qualityAdjustment != 0)
{
    item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

7

Further *Quality* adjustment should not depend on changing *SellIn*

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    qualityAdjustment = 1;
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else
        qualityAdjustment = 3;
}
else if (item.Name == "Sulfuras, Hand of Ragnaros")
{
    qualityAdjustment = 0;
}
else
{
    qualityAdjustment = -1;
}

if (qualityAdjustment != 0)
{
    item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
    else if (item.Name != "Sulfuras, Hand of Ragnaros")
        item.Quality = Max(item.Quality - 1, 0);
}
```

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    qualityAdjustment = 1;
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else
        qualityAdjustment = 3;
}
else if (item.Name == "Sulfuras, Hand of Ragnaros")
{
    qualityAdjustment = 0;
}
else
{
    qualityAdjustment = item.SellIn > 0 ? -1 : -2;
}

if (qualityAdjustment != 0)
{
    item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Aged Brie")
        item.Quality = Min(item.Quality + 1, 50);
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
}
```

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    qualityAdjustment = item.SellIn > 0 ? 1 : 2;
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else
        qualityAdjustment = 3;
}
else if (item.Name == "Sulfuras, Hand of Ragnaros")
{
    qualityAdjustment = 0;
}
else
{
    qualityAdjustment = item.SellIn > 0 ? -1 : -2;
}

if (qualityAdjustment != 0)
{
    item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
    --item.SellIn;
}

if (item.SellIn < 0)
{
    if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
        item.Quality = 0;
}
```

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    qualityAdjustment = item.SellIn > 0 ? 1 : 2;
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else if (item.SellIn > 0)
        qualityAdjustment = 3;
    else
        qualityAdjustment = -item.Quality;
}
else if (item.Name == "Sulfuras, Hand of Ragnaros")
{
    qualityAdjustment = 0;
}
else
{
    qualityAdjustment = item.SellIn > 0 ? -1 : -2;
}

if (qualityAdjustment != 0)
{
    item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
    --item.SellIn;
}
```

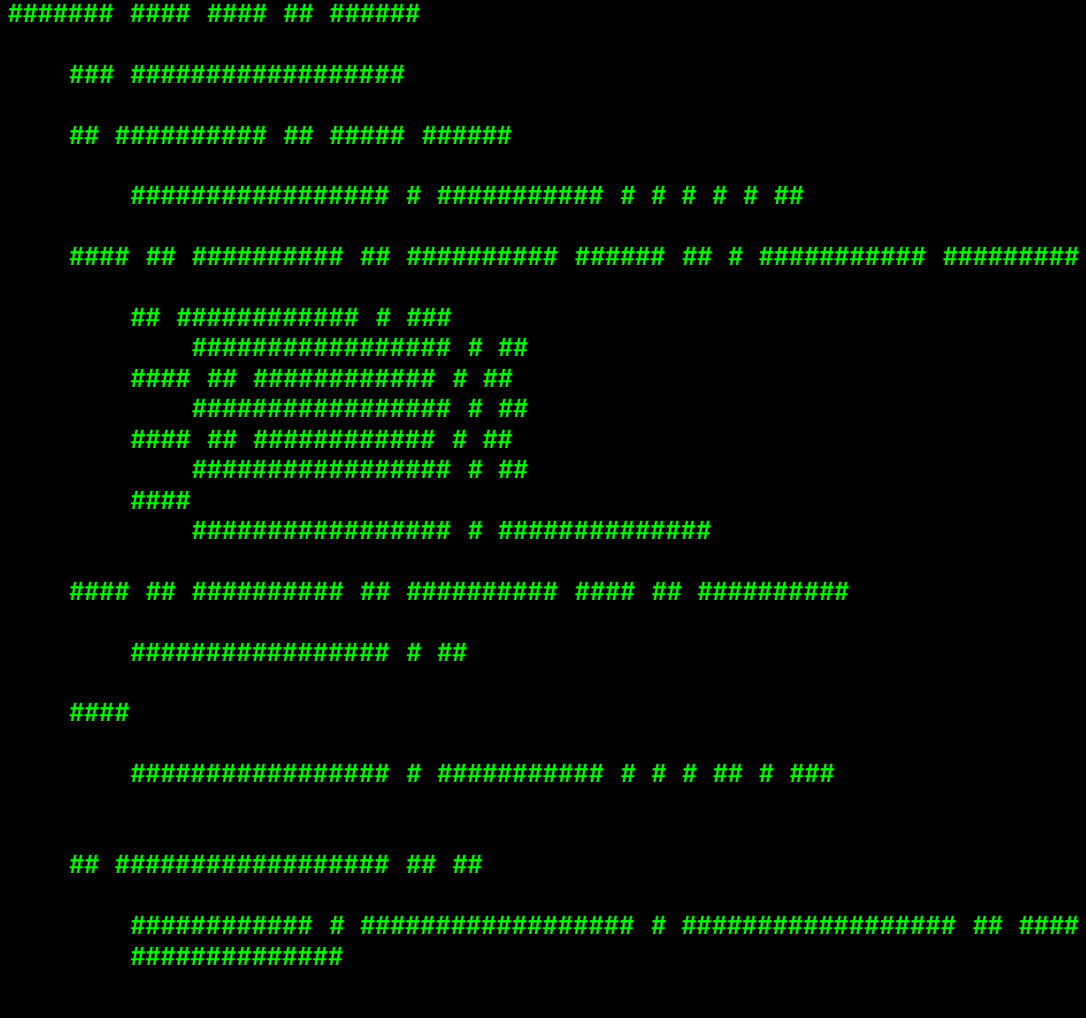
```
foreach (var item in Items)
{
    int qualityAdjustment;

    if (item.Name == "Aged Brie")
    {
        qualityAdjustment = item.SellIn > 0 ? 1 : 2;
    }
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.SellIn > 10)
            qualityAdjustment = 1;
        else if (item.SellIn > 5)
            qualityAdjustment = 2;
        else if (item.SellIn > 0)
            qualityAdjustment = 3;
        else
            qualityAdjustment = -item.Quality;
    }
    else if (item.Name == "Sulfuras, Hand of Ragnaros")
    {
        qualityAdjustment = 0;
    }
    else
    {
        qualityAdjustment = item.SellIn > 0 ? -1 : -2;
    }

    if (qualityAdjustment != 0)
    {
        item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
        --item.SellIn;
    }
}
```

;{;{;;{"""{{""{;}}}{{;""{{{;}}{{;}}}}""{;}{""{""{{""{;}}}{;}}{{;}}}}}{{;;}{;;}{;;}}}

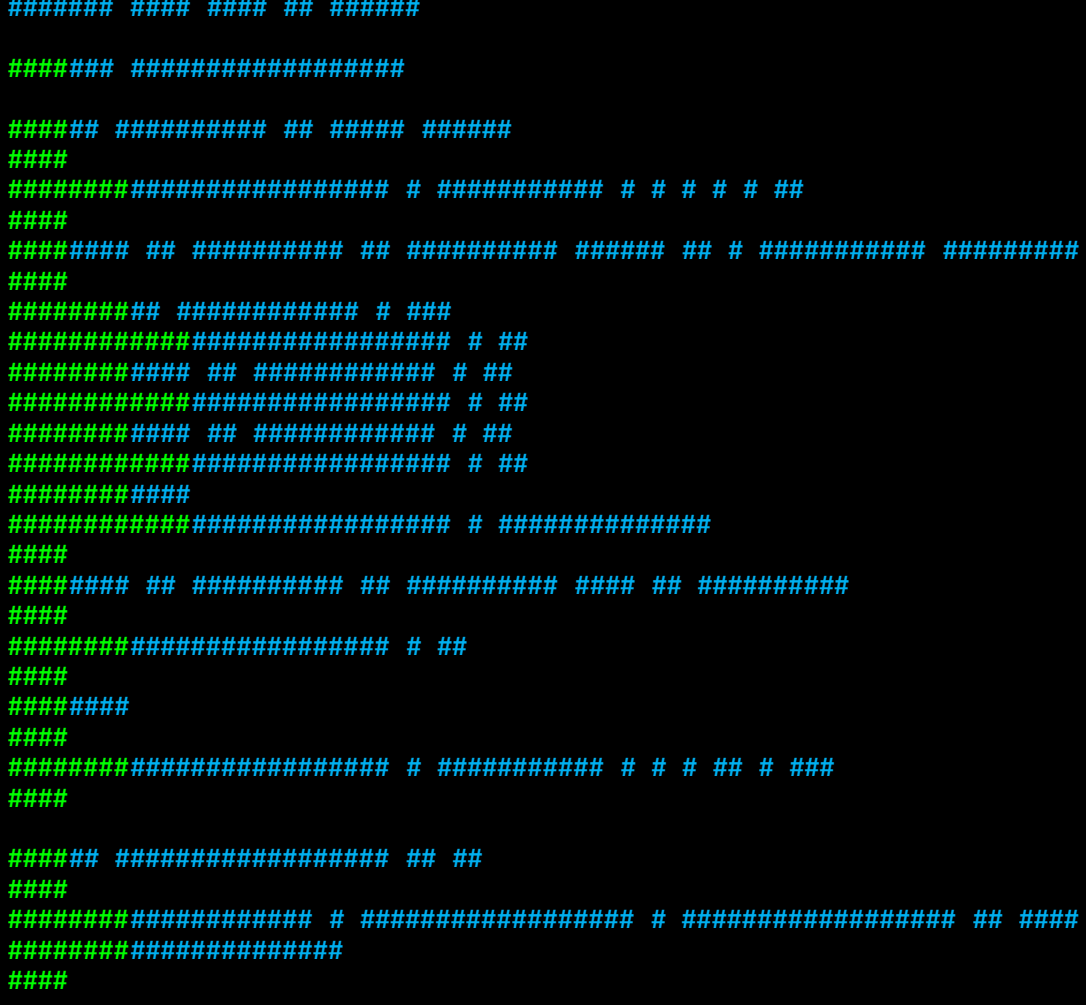
```
{;""{;}""{;;;;}""{;}{;}{;;}}
```

```
foreach (var item in Items)
{
    int qualityAdjustment;

    if (item.Name == "Aged Brie")
    {
        qualityAdjustment = item.SellIn > 0 ? 1 : 2;
    }
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.SellIn > 10)
            qualityAdjustment = 1;
        else if (item.SellIn > 5)
            qualityAdjustment = 2;
        else if (item.SellIn > 0)
            qualityAdjustment = 3;
        else
            qualityAdjustment = -item.Quality;
    }
    else if (item.Name == "Sulfuras, Hand of Ragnaros")
    {
        qualityAdjustment = 0;
    }
    else
    {
        qualityAdjustment = item.SellIn > 0 ? -1 : -2;
    }

    if (qualityAdjustment != 0)
    {
        item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
        --item.SellIn;
    }
}
```

```
foreach (var item in Items)
{
    int qualityAdjustment;

    if (item.Name == "Aged Brie")
    {
        qualityAdjustment = item.SellIn > 0 ? 1 : 2;
    }
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.SellIn > 10)
            qualityAdjustment = 1;
        else if (item.SellIn > 5)
            qualityAdjustment = 2;
        else if (item.SellIn > 0)
            qualityAdjustment = 3;
        else
            qualityAdjustment = -item.Quality;
    }
    else if (item.Name == "Sulfuras, Hand of Ragnaros")
    {
        qualityAdjustment = 0;
    }
    else
    {
        qualityAdjustment = item.SellIn > 0 ? -1 : -2;
    }

    if (qualityAdjustment != 0)
    {
        item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
        --item.SellIn;
    }
}
```



3



```
foreach (var item in Items)
{
    int qualityAdjustment;

    if (item.Name == "Aged Brie")
    {
        qualityAdjustment = item.SellIn > 0 ? 1 : 2;
    }
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.SellIn > 10)
            qualityAdjustment = 1;
        else if (item.SellIn > 5)
            qualityAdjustment = 2;
        else if (item.SellIn > 0)
            qualityAdjustment = 3;
        else
            qualityAdjustment = -item.Quality;
    }
    else if (item.Name == "Sulfuras, Hand of Ragnaros")
    {
        qualityAdjustment = 0;
    }
    else
    {
        qualityAdjustment = item.SellIn > 0 ? -1 : -2;
    }

    if (qualityAdjustment != 0)
    {
        item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
        --item.SellIn;
    }
}
```

await	0
break	0
case	0
catch	0
continue	0
default	0
do	0
else	6
finally	0
for	0
foreach	1
goto	0
if	7
lock	0
return	0
switch	0
throw	0
try	0
while	0
yield	0

```
foreach (var item in Items)
{
    int qualityAdjustment;

    if (item.Name == "Aged Brie")
    {
        qualityAdjustment = item.SellIn > 0 ? 1 : 2;
    }
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.SellIn > 10)
            qualityAdjustment = 1;
        else if (item.SellIn > 5)
            qualityAdjustment = 2;
        else if (item.SellIn > 0)
            qualityAdjustment = 3;
        else
            qualityAdjustment = -item.Quality;
    }
    else if (item.Name == "Sulfuras, Hand of Ragnaros")
    {
        qualityAdjustment = 0;
    }
    else
    {
        qualityAdjustment = item.SellIn > 0 ? -1 : -2;
    }

    if (qualityAdjustment != 0)
    {
        item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
        --item.SellIn;
    }
}
```

await	0
break	0
case	0
catch	0
continue	0
default	0
do	0
else	6
finally	0
for	0
foreach	1
goto	0
if	7
lock	0
return	0
switch	0
throw	0
try	0
while	0
yield	0

```
foreach (var item in Items)
{
    int qualityAdjustment;

    if (item.Name == "Aged Brie")
    {
        qualityAdjustment = item.SellIn > 0 ? 1 : 2;
    }
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.SellIn > 10)
            qualityAdjustment = 1;
        else if (item.SellIn > 5)
            qualityAdjustment = 2;
        else if (item.SellIn > 0)
            qualityAdjustment = 3;
        else
            qualityAdjustment = -item.Quality;
    }
    else if (item.Name == "Sulfuras, Hand of Ragnaros")
    {
        qualityAdjustment = 0;
    }
    else
    {
        qualityAdjustment = item.SellIn > 0 ? -1 : -2;
    }

    if (qualityAdjustment != 0)
    {
        item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
        --item.SellIn;
    }
}
```

await	0
break	0
case	0
catch	0
continue	0
default	0
do	0
else / :	8
finally	0
for	0
foreach	1
goto	0
if / ?	9
lock	0
return	0
switch	0
throw	0
try	0
while	0
yield	0

```
foreach (var item in Items)
{
    int qualityAdjustment;

    if (item.Name == "Aged Brie")
    {
        qualityAdjustment = item.SellIn > 0 ? 1 : 2;
    }
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.SellIn > 10)
            qualityAdjustment = 1;
        else if (item.SellIn > 5)
            qualityAdjustment = 2;
        else if (item.SellIn > 0)
            qualityAdjustment = 3;
        else
            qualityAdjustment = -item.Quality;
    }
    else if (item.Name == "Sulfuras, Hand of Ragnaros")
    {
        qualityAdjustment = 0;
    }
    else
    {
        qualityAdjustment = item.SellIn > 0 ? -1 : -2;
    }

    if (qualityAdjustment != 0)
    {
        item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
        --item.SellIn;
    }
}
```

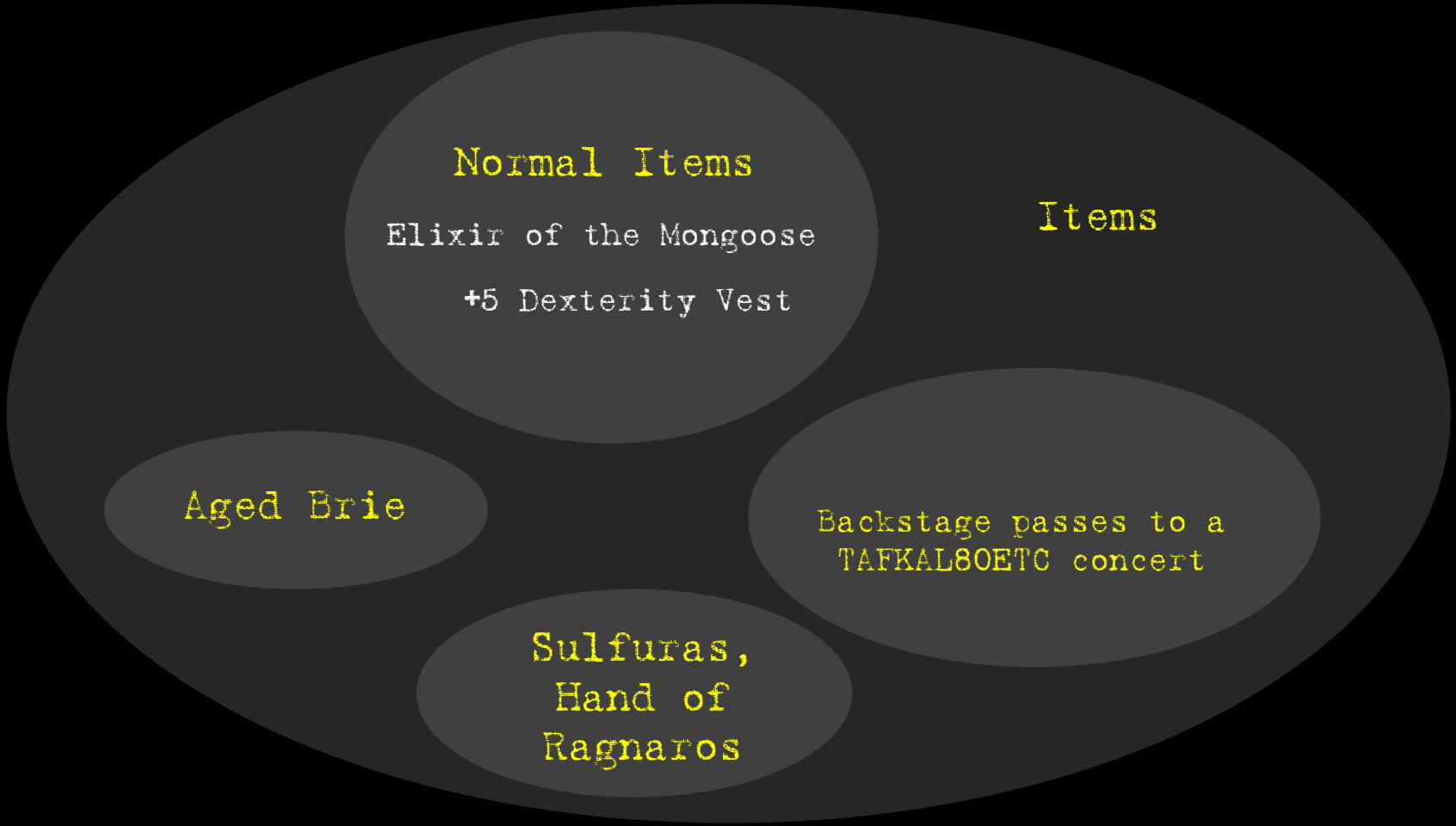
```
foreach (var item in Items)
{
    int qualityAdjustment;

    if (item.Name == "Aged Brie")
    {
        qualityAdjustment = item.SellIn > 0 ? 1 : 2;
    }
    else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
    {
        if (item.SellIn > 10)
            qualityAdjustment = 1;
        else if (item.SellIn > 5)
            qualityAdjustment = 2;
        else if (item.SellIn > 0)
            qualityAdjustment = 3;
        else
            qualityAdjustment = -item.Quality;
    }
    else if (item.Name == "Sulfuras, Hand of Ragnaros")
    {
        qualityAdjustment = 0;
    }
    else
    {
        qualityAdjustment = item.SellIn > 0 ? -1 : -2;
    }

    if (qualityAdjustment != 0)
    {
        item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
        --item.SellIn;
    }
}
```

Refactor, *verb*

to restructure software by applying a series of refactorings without changing the observable behavior of the software

8 Generalise matching of Sulfuras and backstage passes


Items
Normal Items
Elixir of the Mongoose
+5 Dexterity Vest
Aged Brie
Backstage Passes
Backstage passes to a TAFKAL80ETC concert
Sulfuras


Items
Normal Items
Elixir of the Mongoose
+5 Dexterity Vest
Aged Brie
Backstage passes to a TAFKAL80ETC concert
Sulfuras, Hand of Ragnaros

Never trust a test you haven't seen fail.

Marit van Dijk

"Use Testing to Develop Better Software Faster"

medium.com/97-things/use-testing-to-develop-better-software-faster-9dd2616543d3

```
public class The_quality_of_backstage_passes
{
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 11, 0)]
    public void increases_by_1_every_day_up_to_10_days_before_its_sell_by_date(...)
    ...
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 10, 0)]
    public void increases_by_2_every_day_between_10_and_5_days_before_its_sell_by_date(...)
    ...
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 5, 0)]
    public void increases_by_3_every_day_within_5_days_of_its_sell_by_date(...)
    ...
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 0, 10)]
    public void drops_to_0_after_its_sell_by_date(string name, int sellIn, int quality)
    ...
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 1, 49)]
    public void never_increases_beyond_50(string name, int sellIn, int quality)
    ...
}
```

```
public class The_quality_of_backstage_passes
{
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 11, 0)]
    [TestCase("Backstage passes to a Brian Mage gig", 20, 10)]
    public void increases_by_1_every_day_up_to_10_days_before_its_sell_by_date(...)
    ...
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 10, 0)]
    [TestCase("Backstage passes to a Blood Moon show", 8, 10)]
    public void increases_by_2_every_day_between_10_and_5_days_before_its_sell_by_date(...)
    ...
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 5, 0)]
    [TestCase("Backstage passes to a Blood Moon show", 1, 10)]
    public void increases_by_3_every_day_within_5_days_of_its_sell_by_date(...)
    ...
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 0, 10)]
    [TestCase("Backstage passes to an All-Seeing Eye performance", -1, 5)]
    public void drops_to_0_after_its_sell_by_date(...)
    ...
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 1, 49)]
    [TestCase("Backstage passes to a Brian Mage gig", 6, 49)]
    [TestCase("Backstage passes to a Blood Moon show", 11, 50)]
}
```

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    qualityAdjustment = item.SellIn > 0 ? 1 : 2;
}
else if (item.Name == "Backstage passes to a TAFKAL80ETC concert")
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else if (item.SellIn > 0)
        qualityAdjustment = 3;
    else
        qualityAdjustment = -item.Quality;
}
else if (item.Name == "Sulfuras, Hand of Ragnaros")
{
    qualityAdjustment = 0;
}
else
{
    qualityAdjustment = item.SellIn > 0 ? -1 : -2;
}

if (qualityAdjustment != 0)
{
    item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
    --item.SellIn;
}
```

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    qualityAdjustment = item.SellIn > 0 ? 1 : 2;
}
else if (item.Name.StartsWith("Backstage passes"))
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else if (item.SellIn > 0)
        qualityAdjustment = 3;
    else
        qualityAdjustment = -item.Quality;
}
else if (item.Name == "Sulfuras, Hand of Ragnaros")
{
    qualityAdjustment = 0;
}
else
{
    qualityAdjustment = item.SellIn > 0 ? -1 : -2;
}

if (qualityAdjustment != 0)
{
    item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
    --item.SellIn;
}
```

```
public class The_quality_of_backstage_passes
{
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 11, 0)]
    [TestCase("Backstage passes to a Brian Mage gig", 20, 10)]
    public void increases_by_1_every_day_up_to_10_days_before_its_sell_by_date(...)
    ...
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 10, 0)]
    [TestCase("Backstage passes to a Blood Moon show", 8, 10)]
    public void increases_by_2_every_day_between_10_and_5_days_before_its_sell_by_date(...)
    ...
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 5, 0)]
    [TestCase("Backstage passes to a Blood Moon show", 1, 10)]
    public void increases_by_3_every_day_within_5_days_of_its_sell_by_date(...)
    ...
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 0, 10)]
    [TestCase("Backstage passes to an All-Seeing Eye performance", -1, 5)]
    public void drops_to_0_after_its_sell_by_date(...)
    ...
    [TestCase("Backstage passes to a TAFKAL80ETC concert", 1, 49)]
    [TestCase("Backstage passes to a Brian Mage gig", 6, 49)]
    [TestCase("Backstage passes to a Blood Moon show", 11, 50)]
}
```

```
public class The_quality_of_sulfuras
{
    [TestCase("Sulfuras, Hand of Ragnaros", 0, 80)]
    [TestCase("Sulfuras, Hand of Ragnaros", -1, 80)]
    public void never_changes_and_never_has_to_be_sold(...)
    ...
}
```

```
public class The_quality_of_sulfuras
{
    [TestCase("Sulfuras, Hand of Ragnaros", 0, 80)]
    [TestCase("Sulfuras", 1, 80)]
    [TestCase("Sulfuras, Hand of Ragnaros", -1, 80)]
    public void never_changes_and_never_has_to_be_sold(...)
    ...
}
```

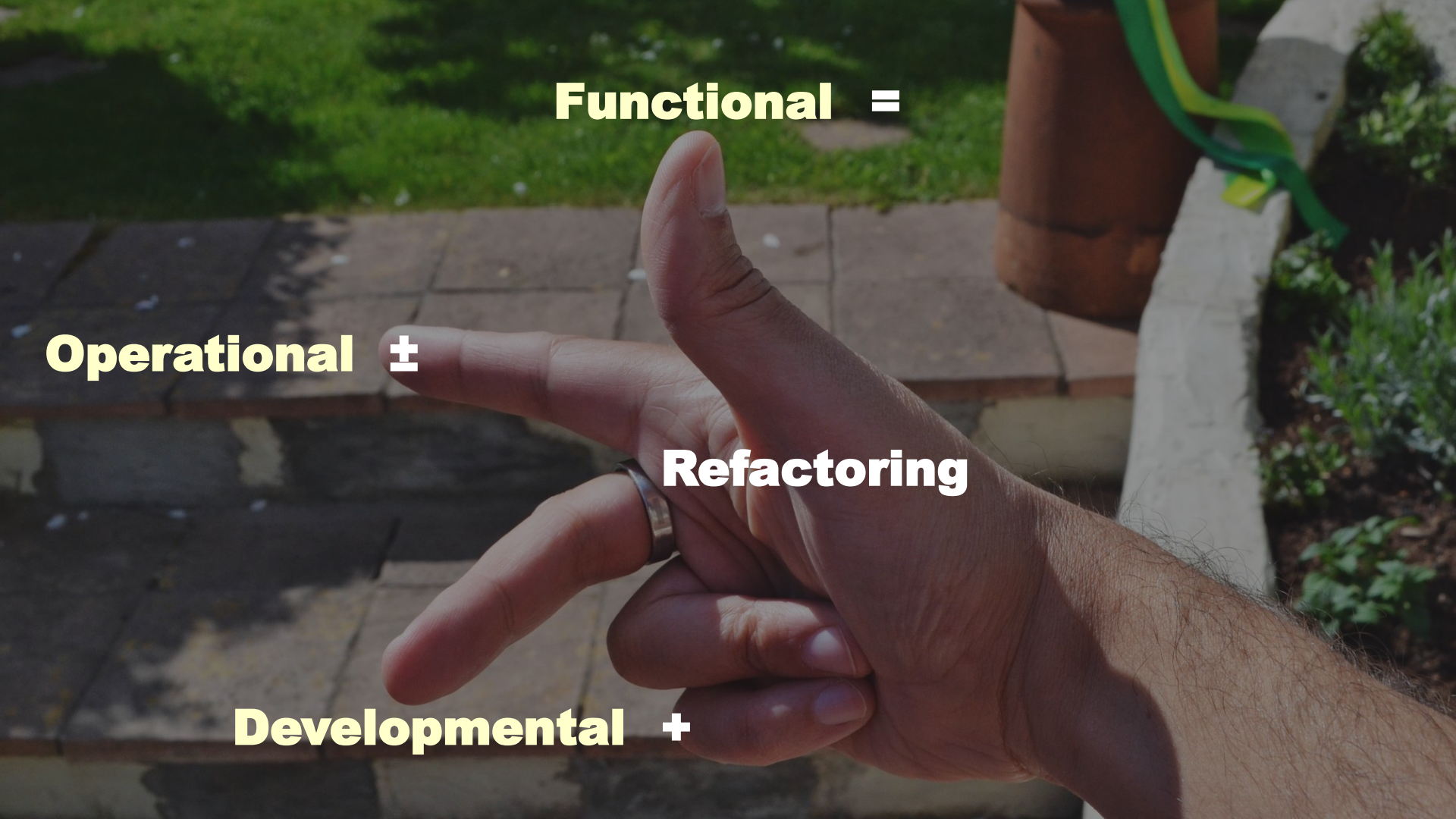
```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    qualityAdjustment = item.SellIn > 0 ? 1 : 2;
}
else if (item.Name.StartsWith("Backstage passes"))
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else if (item.SellIn > 0)
        qualityAdjustment = 3;
    else
        qualityAdjustment = -item.Quality;
}
else if (item.Name == "Sulfuras, Hand of Ragnaros")
{
    qualityAdjustment = 0;
}
else
{
    qualityAdjustment = item.SellIn > 0 ? -1 : -2;
}

if (qualityAdjustment != 0)
{
    item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
    --item.SellIn;
}
```

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    qualityAdjustment = item.SellIn > 0 ? 1 : 2;
}
else if (item.Name.StartsWith("Backstage passes"))
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else if (item.SellIn > 0)
        qualityAdjustment = 3;
    else
        qualityAdjustment = -item.Quality;
}
else if (item.Name.StartsWith("Sulfuras"))
{
    qualityAdjustment = 0;
}
else
{
    qualityAdjustment = item.SellIn > 0 ? -1 : -2;
}

if (qualityAdjustment != 0)
{
    item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
    --item.SellIn;
}
```

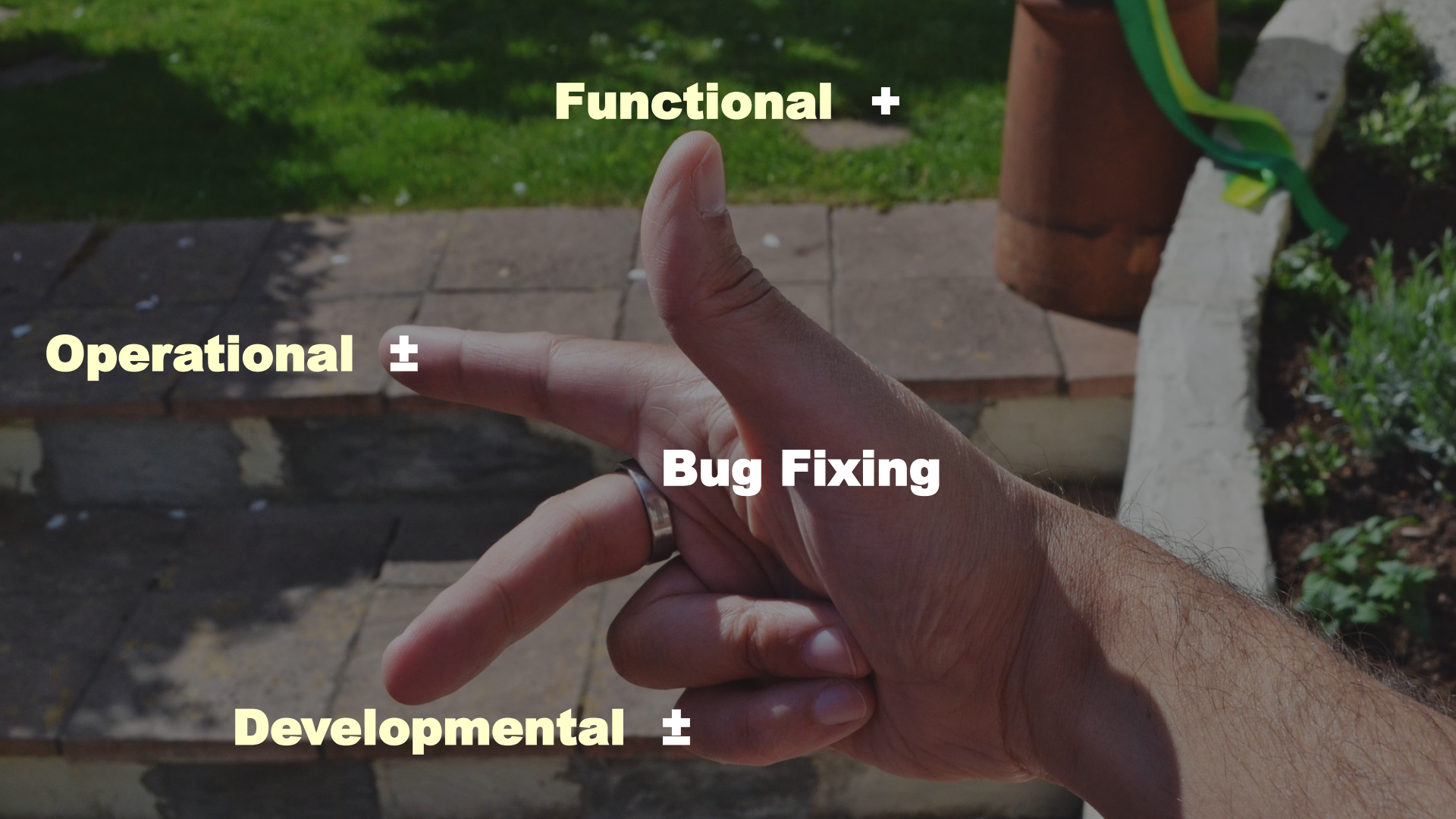
```
public class The_quality_of_sulfuras
{
    [TestCase("Sulfuras, Hand of Ragnaros", 0, 80)]
    [TestCase("Sulfuras", 1, 80)]
    [TestCase("Sulfuras, Hand of Ragnaros", -1, 80)]
    public void never_changes_and_never_has_to_be_sold(...)
    ...
}
```

9 Add conjured items

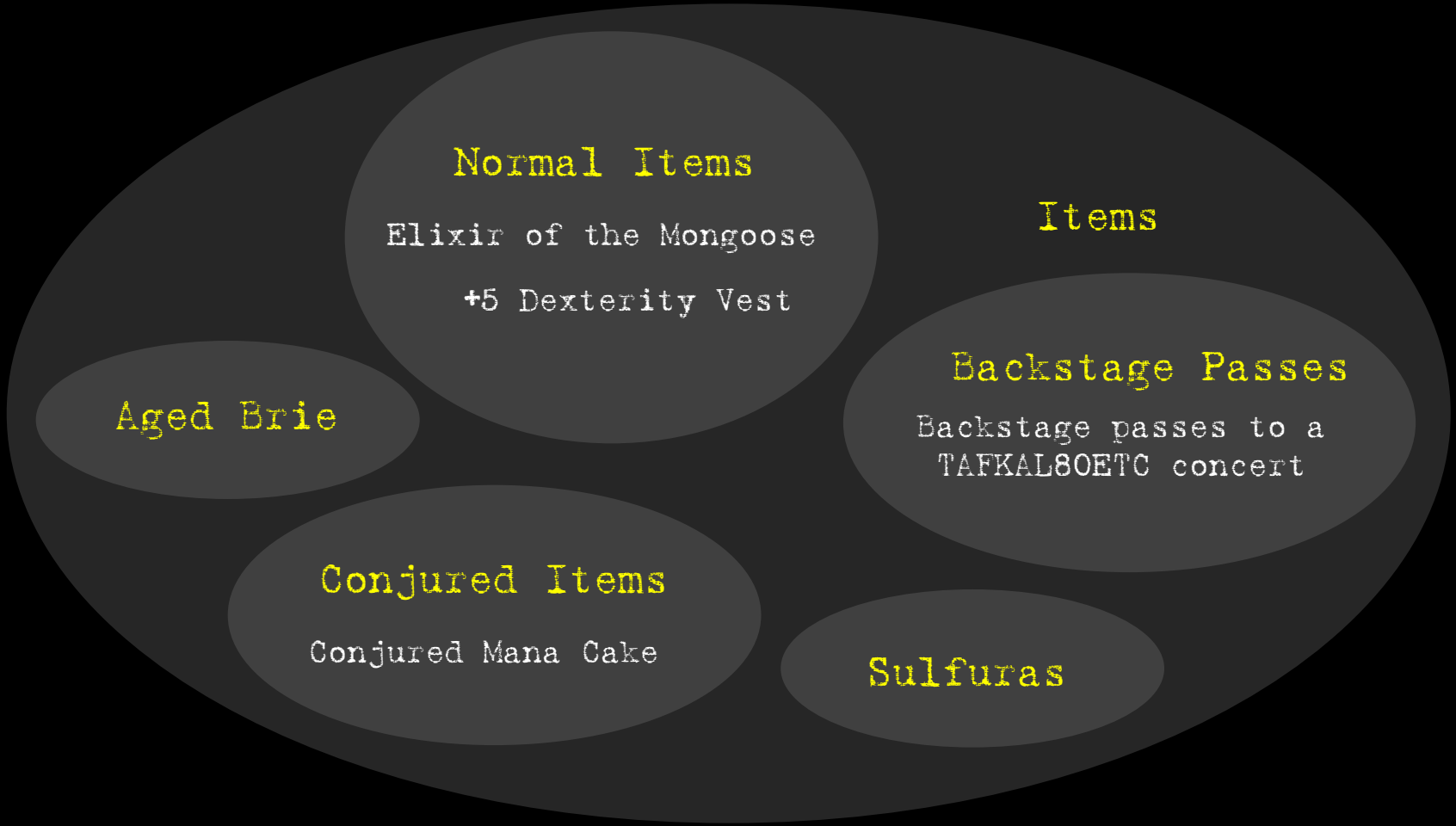

Functional
Operational
Developmental


Functional =
Operational ±
Refactoring
Developmental +


Functional =
Operational +
Optimisation
Developmental ±


Functional +
Operational ±
Bug Fixing
Developmental ±


Functional +
Operational ±
Feature Addition
Developmental ±


Items
Normal Items
Elixir of the Mongoose
+5 Dexterity Vest
Aged Brie
Backstage Passes
Backstage passes to a TAFKAL80ETC concert
Conjured Items
Conjured Mana Cake
Sulfuras

```
public class The_quality_of_a_conjured_item
{
    [TestCase("Conjured Mana Cake", 10, 20)]
    [TestCase("Conjured Overnight Oats", 5, 7)]
    [TestCase("Conjured Spawn Spread", 1, 2)]
    [TestCase("Conjured Dry Seed", 3, 5)]
    public void decreases_by_2_every_day_before_its_sell_by_date(...)
    ...
    [TestCase("Conjured Mana Cake", 0, 10)]
    [TestCase("Conjured Overnight Oats", -1, 5)]
    [TestCase("Conjured Spawn Spread", 0, 4)]
    [TestCase("Conjured Dry Seed", -2, 4)]
    public void decreases_by_4_every_day_past_its_sell_by_date(...)
    ...
    [TestCase("Conjured Mana Cake", 1, 0)]
    [TestCase("Conjured Overnight Oats", -5, 2)]
    [TestCase("Conjured Spawn Spread", 0, 2)]
    [TestCase("Conjured Dry Seed", -2, 1)]
    public void never_decreases_below_0(...)
    ...
}
```

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    qualityAdjustment = item.SellIn > 0 ? 1 : 2;
}
else if (item.Name.StartsWith("Backstage passes"))
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else if (item.SellIn > 0)
        qualityAdjustment = 3;
    else
        qualityAdjustment = -item.Quality;
}
else if (item.Name.StartsWith("Sulfuras"))
{
    qualityAdjustment = 0;
}
else
{
    qualityAdjustment = item.SellIn > 0 ? -1 : -2;
}

if (qualityAdjustment != 0)
{
    item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
    --item.SellIn;
}
```

```
int qualityAdjustment;

if (item.Name == "Aged Brie")
{
    qualityAdjustment = item.SellIn > 0 ? 1 : 2;
}
else if (item.Name.StartsWith("Backstage passes"))
{
    if (item.SellIn > 10)
        qualityAdjustment = 1;
    else if (item.SellIn > 5)
        qualityAdjustment = 2;
    else if (item.SellIn > 0)
        qualityAdjustment = 3;
    else
        qualityAdjustment = -item.Quality;
}
else if (item.Name.StartsWith("Conjured"))
{
    qualityAdjustment = item.SellIn > 0 ? -2 : -4;
}
else if (item.Name.StartsWith("Sulfuras"))
{
    qualityAdjustment = 0;
}
else
{
    qualityAdjustment = item.SellIn > 0 ? -1 : -2;
}

if (qualityAdjustment != 0)
{
    item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
    --item.SellIn;
}
```

```
public class The_quality_of_a_conjured_item
{
    [TestCase("Conjured Mana Cake", 10, 20)]
    [TestCase("Conjured Overnight Oats", 5, 7)]
    [TestCase("Conjured Spawn Spread", 1, 2)]
    [TestCase("Conjured Dry Seed", 3, 5)]
    public void decreases_by_2_every_day_before_its_sell_by_date(...)
    ...
    [TestCase("Conjured Mana Cake", 0, 10)]
    [TestCase("Conjured Overnight Oats", -1, 5)]
    [TestCase("Conjured Spawn Spread", 0, 4)]
    [TestCase("Conjured Dry Seed", -2, 4)]
    public void decreases_by_4_every_day_past_its_sell_by_date(...)
    ...
    [TestCase("Conjured Mana Cake", 1, 0)]
    [TestCase("Conjured Overnight Oats", -5, 2)]
    [TestCase("Conjured Spawn Spread", 0, 2)]
    [TestCase("Conjured Dry Seed", -2, 1)]
    public void never_decreases_below_0(...)
    ...
}
```

10

Shift paradigm

The Paradigms of Programming

Robert W. Floyd
Stanford University

Paradigm(pæ·radim, −dəim) . . . [a. F. *paradigme*, ad. L. *paradigma*, a. Gr. παραδειγμα pattern, example, f. παραδεικνυ·ναι to exhibit beside, show side by side. . .]
1. A pattern, exemplar, example.
1752 J. Gill *Trinity* v. 91
The archetype, paradigm, exemplar, and idea, according to which all things were made.
From the Oxford English Dictionary.

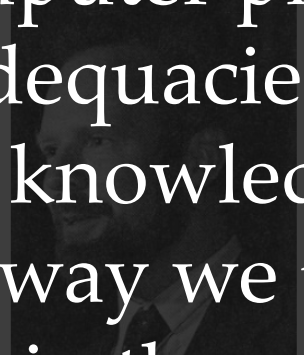
Today I want to talk about the paradigms of programming, how they affect our success as designers of computer programs, how they should be taught, and how they should be embodied in our programming languages.

A familiar example of a paradigm of programming is the technique of *structured programming*, which appears to be the dominant paradigm in most current treatments of programming methodology. Structured programming, as formulated by Dijkstra [6], Wirth [27, 29], and Parnas [21], among others, consists of two phases.

In the first phase, that of top-down design, or stepwise refinement, the problem is decomposed into a very small number of simpler subproblems. In programming the solution of simultaneous linear equations, say, the first level of decomposition would be into a stage of triangularizing the equations and a following stage of back-substitution in the triangularized system. This gradual decomposition is continued until the subproblems that arise are simple enough to cope with directly. In the simultaneous equation example, the back substitution process would be further decomposed as a backwards iteration of a process which finds and stores the value of the *i*th variable from the *i*th equation. Yet further decomposition would yield a fully detailed algorithm.

The Paradigms of Programming

Robert W. Floyd



The archetype, paradigm, exemplar, and idea,
according to which all things were made.
From the Oxford English Dictionary.

I believe that the current state of the art of computer programming reflects inadequacies in our stock of paradigms, in our knowledge of existing paradigms, in the way we teach programming paradigms, and in the way our programming languages support, or fail to support, the paradigms of their user communities.

stringly typed

```
public void UpdateQuality()
{
    foreach (var item in Items)
    {
        int qualityAdjustment;

        if (item.Name == "Aged Brie")
        {
            qualityAdjustment = item.SellIn > 0 ? 1 : 2;
        }
        else if (item.Name.StartsWith("Backstage passes"))
        {
            if (item.SellIn > 10)
                qualityAdjustment = 1;
            else if (item.SellIn > 5)
                qualityAdjustment = 2;
            else if (item.SellIn > 0)
                qualityAdjustment = 3;
            else
                qualityAdjustment = -item.Quality;
        }
        else if (item.Name.StartsWith("Conjured"))
        {
            qualityAdjustment = item.SellIn > 0 ? -2 : -4;
        }
        else if (item.Name.StartsWith("Sulfuras"))
        {
            qualityAdjustment = 0;
        }
        else
        {
            qualityAdjustment = item.SellIn > 0 ? -1 : -2;
        }

        if (qualityAdjustment != 0)
        {
            item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
            --item.SellIn;
        }
    }
}
```

```
public void UpdateQuality()
{
    foreach (var item in Items)
    {
        int qualityAdjustment;

        if (IsMatch(item.Name, "^Aged Brie$"))
        {
            qualityAdjustment = item.SellIn > 0 ? 1 : 2;
        }
        else if (item.Name.StartsWith("Backstage passes"))
        {
            if (item.SellIn > 10)
                qualityAdjustment = 1;
            else if (item.SellIn > 5)
                qualityAdjustment = 2;
            else if (item.SellIn > 0)
                qualityAdjustment = 3;
            else
                qualityAdjustment = -item.Quality;
        }
        else if (IsMatch(item.Name, "^Conjured"))
        {
            qualityAdjustment = item.SellIn > 0 ? -2 : -4;
        }
        else if (IsMatch(item.Name, "^Sulfuras"))
        {
            qualityAdjustment = 0;
        }
        else
        {
            qualityAdjustment = item.SellIn > 0 ? -1 : -2;
        }

        if (qualityAdjustment != 0)
        {
            item.Quality = Clamp(item.Quality + qualityAdjustment, 0, 50);
            --item.SellIn;
        }
    }
}
```

```
public void UpdateQuality()
{
    foreach (var item in Items)
    {
        Func<int> qualityAdjustment;

        if (IsMatch(item.Name, "^Aged Brie$"))
        {
            qualityAdjustment = () => item.SellIn > 0 ? 1 : 2;
        }
        else if (item.Name.StartsWith("Backstage passes"))
        {
            qualityAdjustment = () =>
                                {
                                    if (item.SellIn > 10)
                                        return 1;
                                    else if (item.SellIn > 5)
                                        return 2;
                                    else if (item.SellIn > 0)
                                        return 3;
                                    else
                                        return -item.Quality;
                                };
        }
        else if (IsMatch(item.Name, "^Conjured"))
        {
            qualityAdjustment = () => item.SellIn > 0 ? -2 : -4;
        }
        else if (IsMatch(item.Name, "^Sulfuras"))
        {
            qualityAdjustment = null;
        }
        else
        {
            qualityAdjustment = () => item.SellIn > 0 ? -1 : -2;
        }

        if (qualityAdjustment != null)
        {
            item.Quality = Clamp(item.Quality + qualityAdjustment(), 0, 50);
            --item.SellIn;
        }
    }
}
```

```
private static (string rule, Func<Item, int> adjustment)[] qualityAdjustments =
{
    ("^Aged Brie$",       item => item.SellIn > 0 ? 1 : 2),
    ("^Backstage passes", item =>
                          {
                              if (item.SellIn > 10)
                                  return 1;
                              else if (item.SellIn > 5)
                                  return 2;
                              else if (item.SellIn > 0)
                                  return 3;
                              else
                                  return -item.Quality;
                          }),
    ("^Conjured",         item => item.SellIn > 0 ? -2 : -4),
    ("^Sulfuras",         null),
    (".*",                item => item.SellIn > 0 ? -1 : -2),
};

public void UpdateQuality()
{
    foreach (var item in Items)
    {
        foreach (var matching in qualityAdjustments)
        {
            var (rule, adjustment) = matching;

            if (IsMatch(item.Name, rule))
            {
                if (adjustment != null)
                {
                    item.Quality = Clamp(item.Quality + adjustment(item), 0, 50);
                    --item.SellIn;
                }
                break;
            }
        }
    }
}
```

```
private static (string rule, Func<Item, int> adjustment)[] qualityAdjustments =
{
    ("^Aged Brie$",       item => item.SellIn > 0 ? 1 : 2),
    ("^Backstage passes", item =>
                          {
                              if (item.SellIn > 10)
                                  return 1;
                              else if (item.SellIn > 5)
                                  return 2;
                              else if (item.SellIn > 0)
                                  return 3;
                              else
                                  return -item.Quality;
                          }),
    ("^Conjured",         item => item.SellIn > 0 ? -2 : -4),
    ("^Sulfuras",         null),
    (".*",                item => item.SellIn > 0 ? -1 : -2),
};

public void UpdateQuality()
{
    foreach (var item in Items)
    {
        var adjustment = qualityAdjustments.First(matching => IsMatch(item.Name, matching.rule)).adjustment;

        if (adjustment != null)
        {
            item.Quality = Clamp(item.Quality + adjustment(item), 0, 50);
            --item.SellIn;
        }
    }
}
```

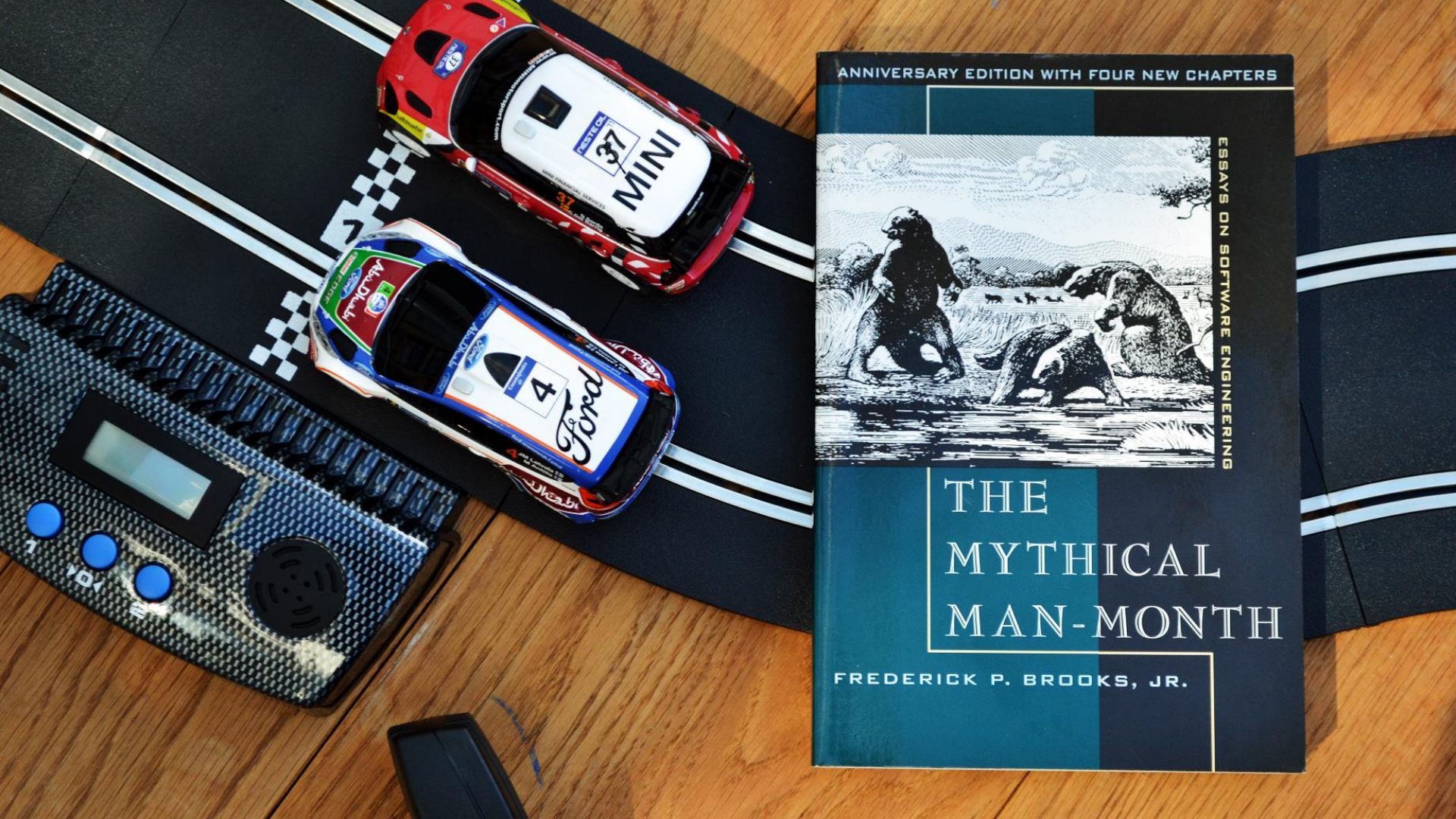
The purpose of abstraction is not to be vague, but to create a new semantic level in which one can be absolutely precise.

Edsger W Dijkstra

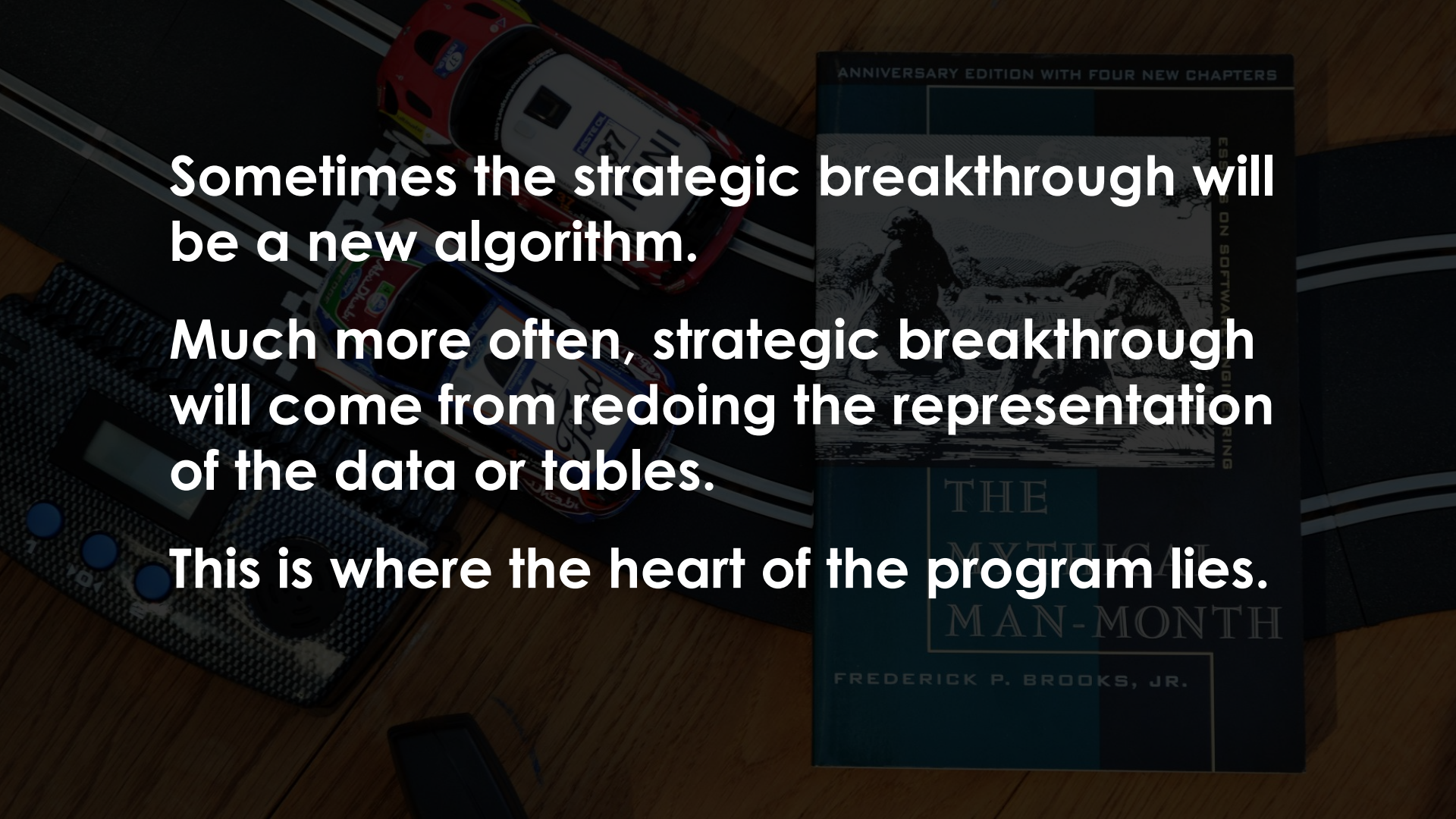
```
private static (string rule, Func<Item, int> adjustment)[] qualityAdjustments =
{
    ("^Aged Brie$",       item => item.SellIn > 0 ? 1 : 2),
    ("^Backstage passes", item =>
                          {
                              if (item.SellIn > 10)
                                  return 1;
                              else if (item.SellIn > 5)
                                  return 2;
                              else if (item.SellIn > 0)
                                  return 3;
                              else
                                  return -item.Quality;
                          }),
    ("^Conjured",         item => item.SellIn > 0 ? -2 : -4),
    ("^Sulfuras",         null),
    (".*",                item => item.SellIn > 0 ? -1 : -2),
};

public void UpdateQuality()
{
    foreach (var item in Items)
    {
        var adjustment = qualityAdjustments.First(matching => IsMatch(item.Name, matching.rule)).adjustment;

        if (adjustment != null)
        {
            item.Quality = Clamp(item.Quality + adjustment(item), 0, 50);
            --item.SellIn;
        }
    }
}
```

```
private static (string rule, Func<Item, int> adjustment)[] qualityAdjustments =
{
    ("^Aged Brie$",       item => item.SellIn > 0 ? 1 : 2),
    ("^Backstage passes", item =>
                          {
                              if (item.SellIn > 10)
                                  return 1;
                              else if (item.SellIn > 5)
                                  return 2;
                              else if (item.SellIn > 0)
                                  return 3;
                              else
                                  return -item.Quality;
                          }),
    ("^Conjured",         item => item.SellIn > 0 ? -2 : -4),
    ("^Sulfuras",         null),
    (".*",                item => item.SellIn > 0 ? -1 : -2),
};

public void UpdateQuality()
{
    foreach (var item in Items)
    {
        var adjustment = qualityAdjustments.First(matching => IsMatch(item.Name, matching.rule)).adjustment;

        if (adjustment != null)
        {
            item.Quality = Clamp(item.Quality + adjustment(item), 0, 50);
            --item.SellIn;
        }
    }
}
```

```
private static (string rule, Func<Item, int> adjustment)[] qualityAdjustments =
{
    ("^Aged Brie$",       item => item.SellIn > 0 ? 1 : 2),
    ("^Backstage passes", item =>
                          {
                              if (item.SellIn > 10)
                                  return 1;
                              else if (item.SellIn > 5)
                                  return 2;
                              else if (item.SellIn > 0)
                                  return 3;
                              else
                                  return -item.Quality;
                          }),
    ("^Conjured",         item => item.SellIn > 0 ? -2 : -4),
    ("^Sulfuras",         null),
    (".*",                item => item.SellIn > 0 ? -1 : -2),
};

public void UpdateQuality()
{
    foreach (var item in Items)
    {
        var adjustment = qualityAdjustments.First(matching => IsMatch(item.Name, matching.rule)).adjustment;

        if (adjustment != null)
        {
            item.Quality = Clamp(item.Quality + adjustment(item), 0, 50);
            --item.SellIn;
        }
    }
}
```


ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS
ESSAYS ON SOFTWARE ENGINEERING
THE MYTHICAL MAN-MONTH
FREDERICK P. BROOKS, JR.
MINI
37
Ford
4

Representation Is the Essence of Programming



Sometimes the strategic breakthrough will be a new algorithm.

Much more often, strategic breakthrough will come from redoing the representation of the data or tables.

This is where the heart of the program lies.

Architecture is inhabited sculpture.

Constantin Brancusi

There is a beautiful angel in that block of marble, and I am going to find it. All I have to do is to knock off the outside pieces of marble, and be very careful not to cut into the angel with my chisel.

George F Pentecost

"The Angel in the Marble"

software

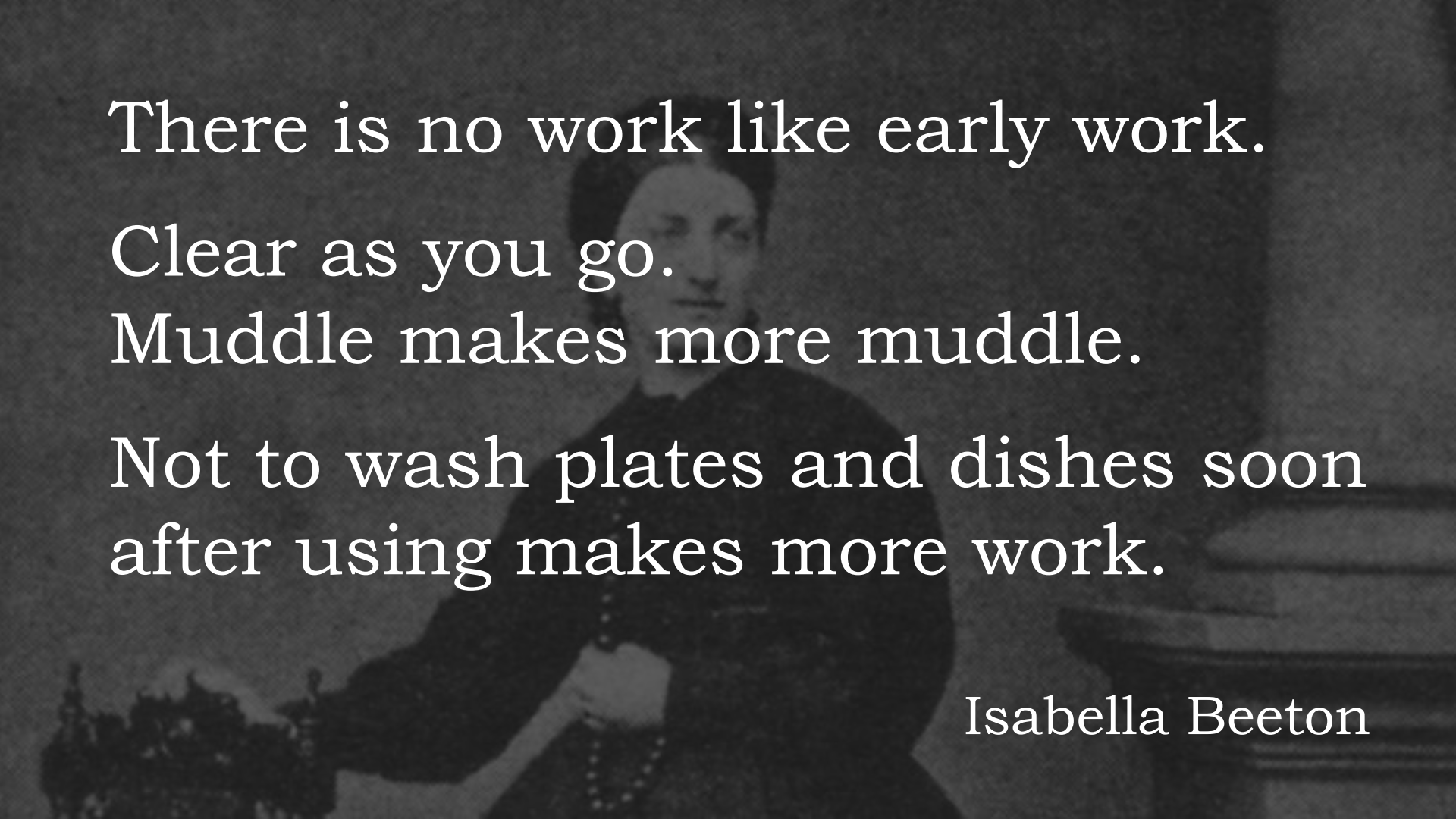
soft

soft

How we spend our days is, of course, how we spend our lives.

Annie Dillard


Isabella Beeton



There is no work like early work.

Clear as you go.
Muddle makes more muddle.

Not to wash plates and dishes soon after using makes more work.

Isabella Beeton

The only kind of writing is rewriting.

Ernest Hemingway