

# ThoughtWorks®

*DDD Brisbane 2016*

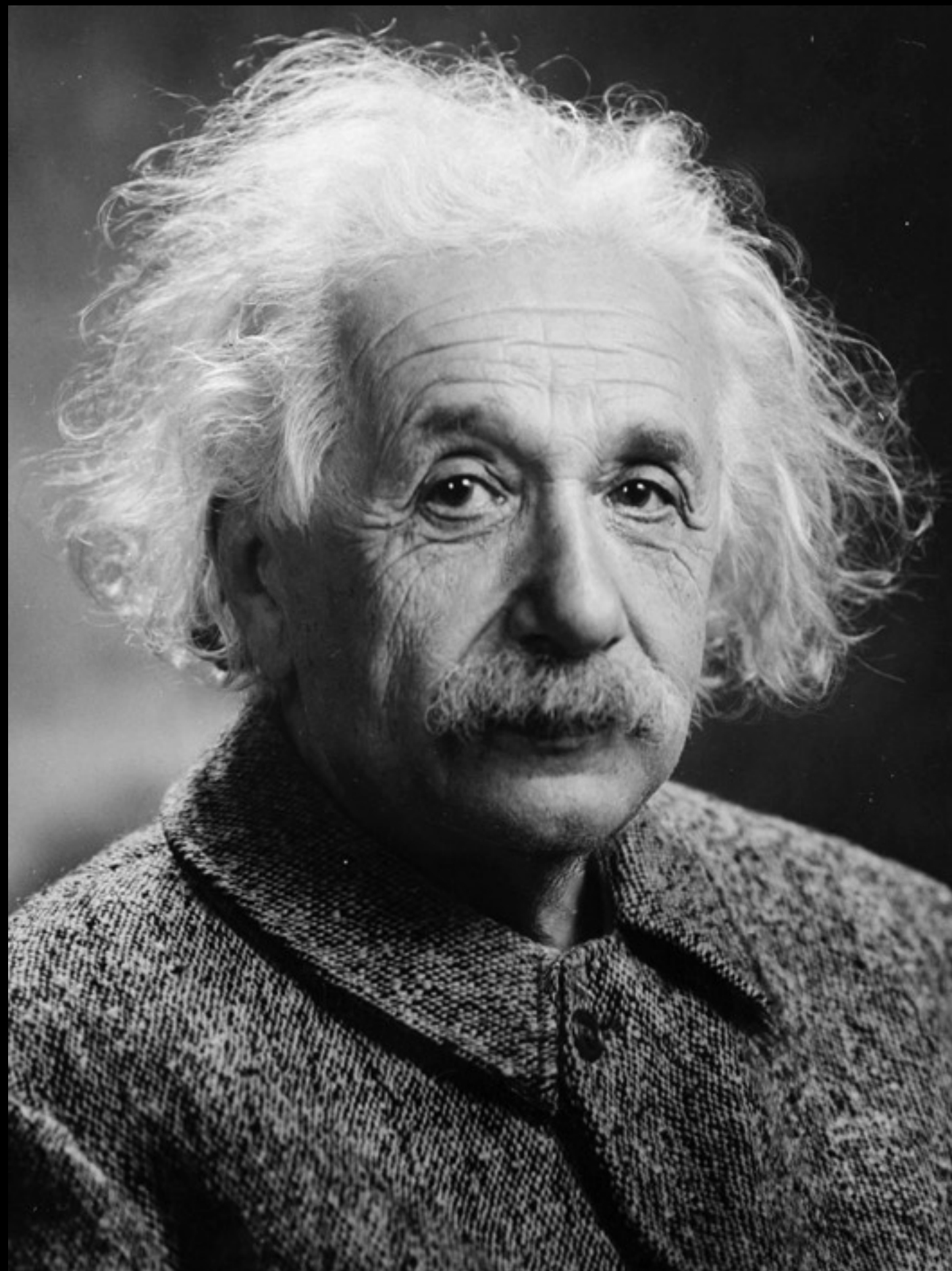
---

**IF SIMPLE WERE EASY,  
WHY IS MAINTENANCE SO HARD?**

---

*Sarah Taraporewalla*  
[www.sarahtaraporewalla.com](http://www.sarahtaraporewalla.com)

“Everything should be made as simple as possible,  
but not simpler”



2 6 8

2

6

8

1

3

4

2

6

8

1

3

4

2

5

7

2	6	8
1	3	4
2	5	7
5	8	11

2 6 8  
1 3 4  
**x** < **y** < **z**  
2 7  
5 8 11

# TODAY WE WILL EXPLORE

---

Simple and Complex

Simple Code

Simple Explanations



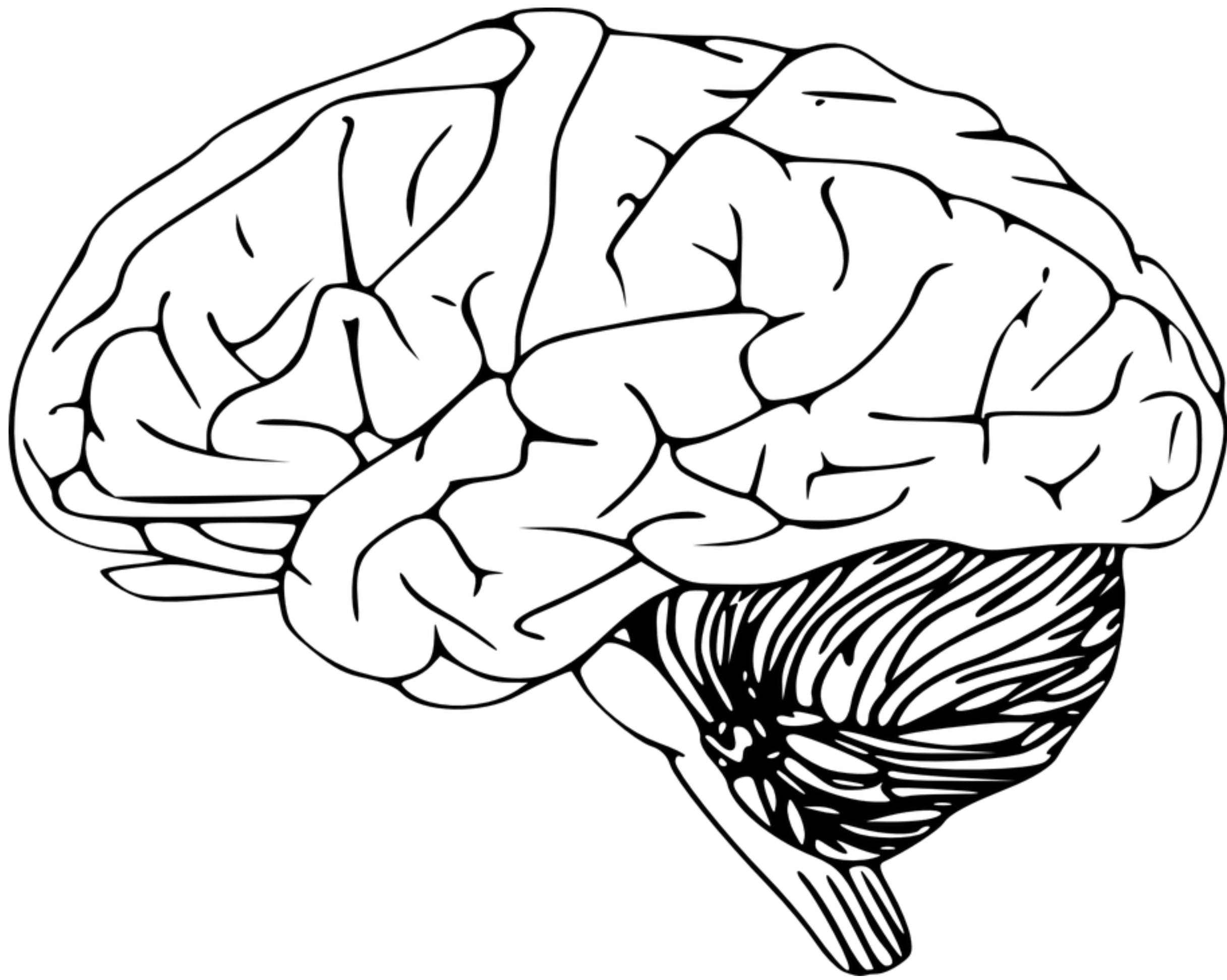
# SIMPLE AND COMPLEX

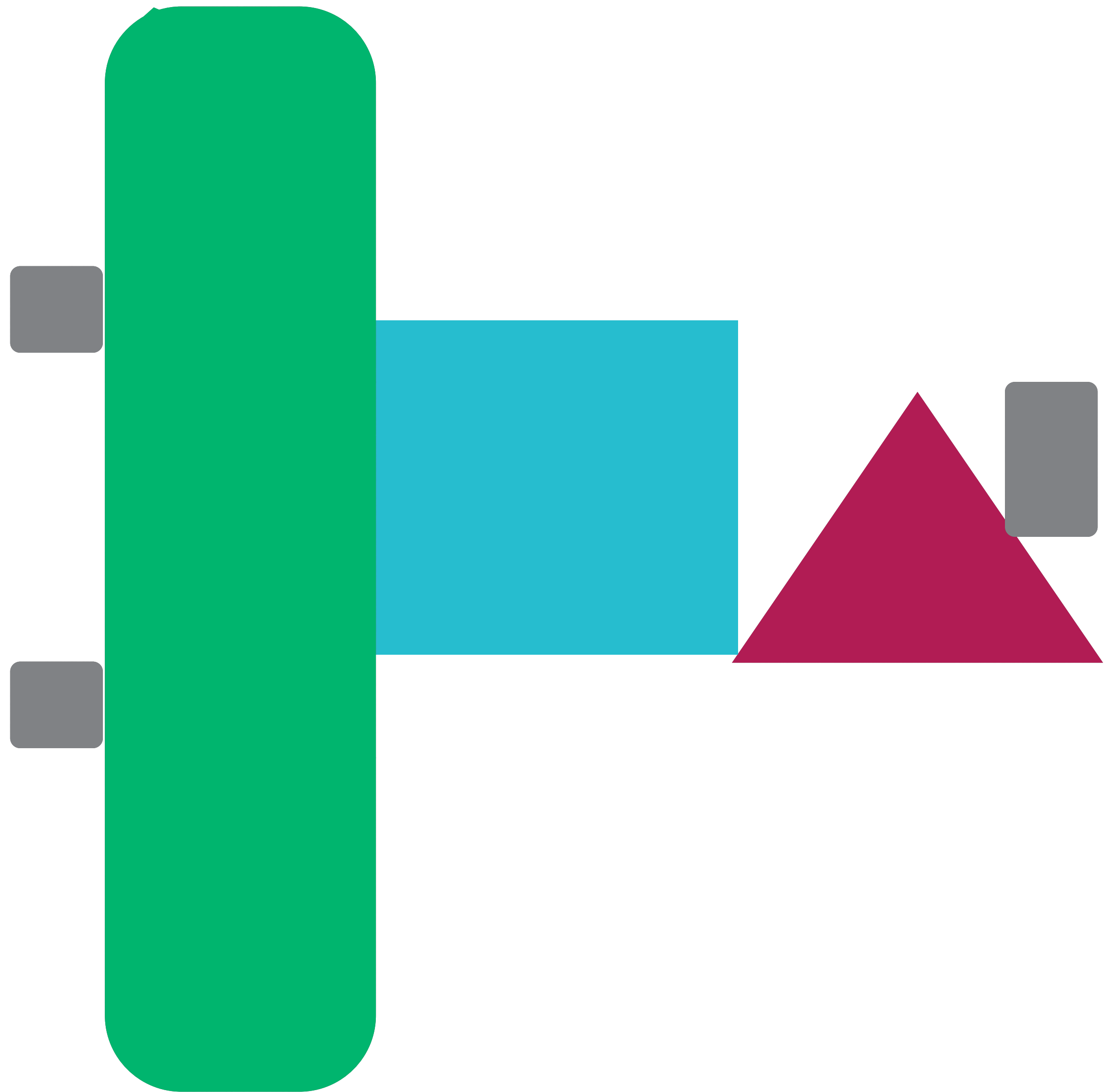
---

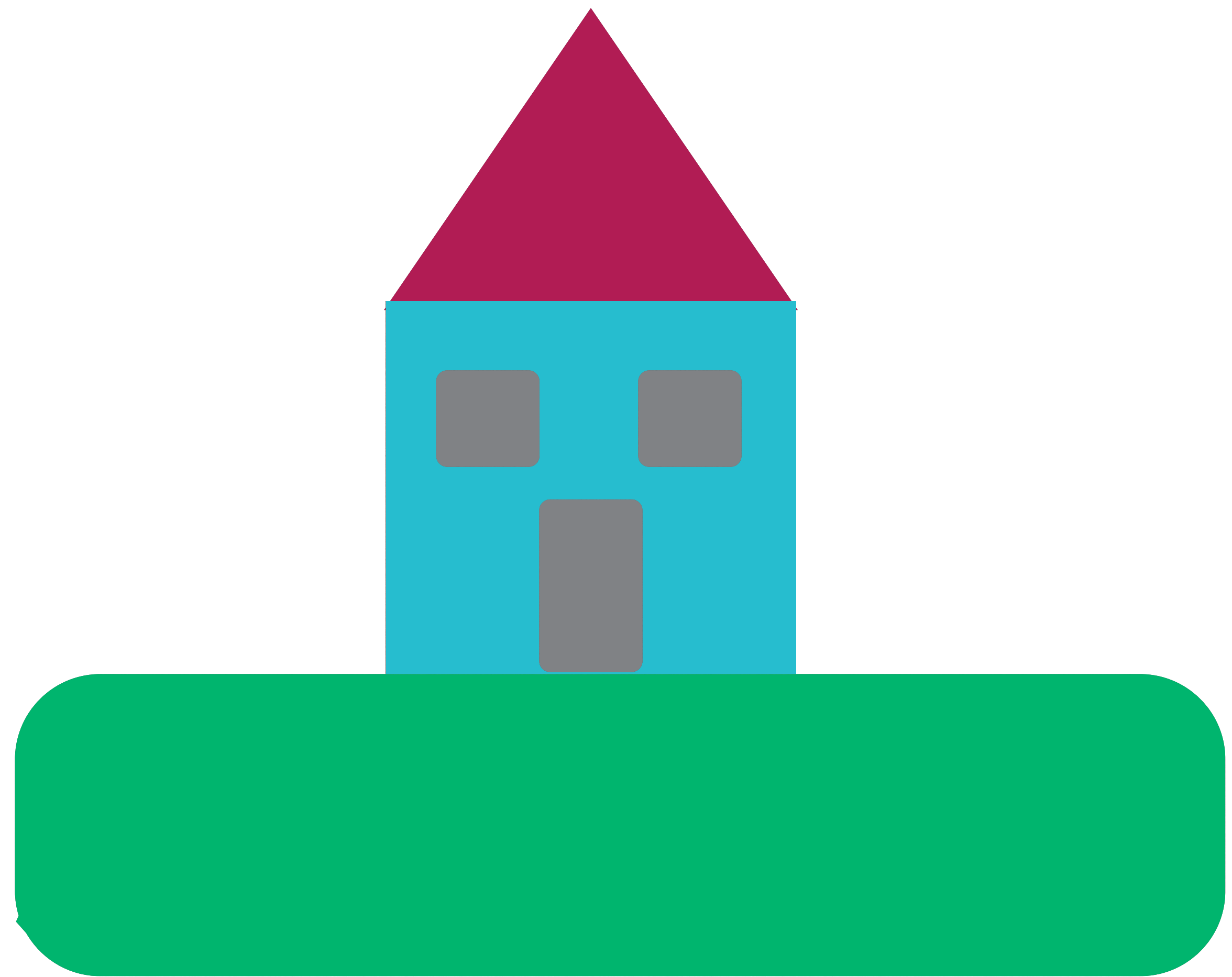
“Simplicity is a great virtue but it requires hard work  
to achieve it”

- Edsger W Dijkstra

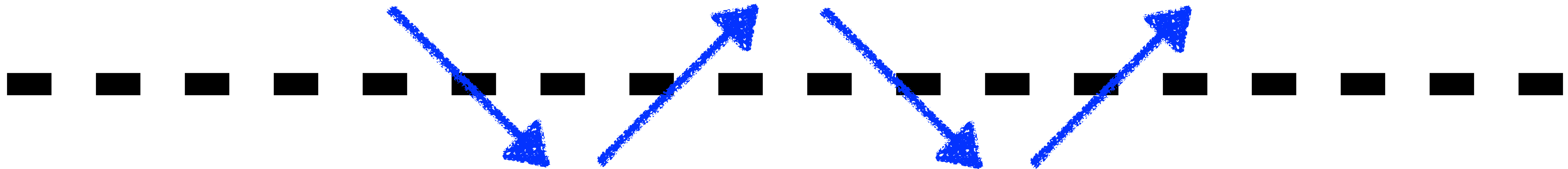








# GENERALISE



# SPECIALISE

## PROBLEM WITH COMPLEXITY

---

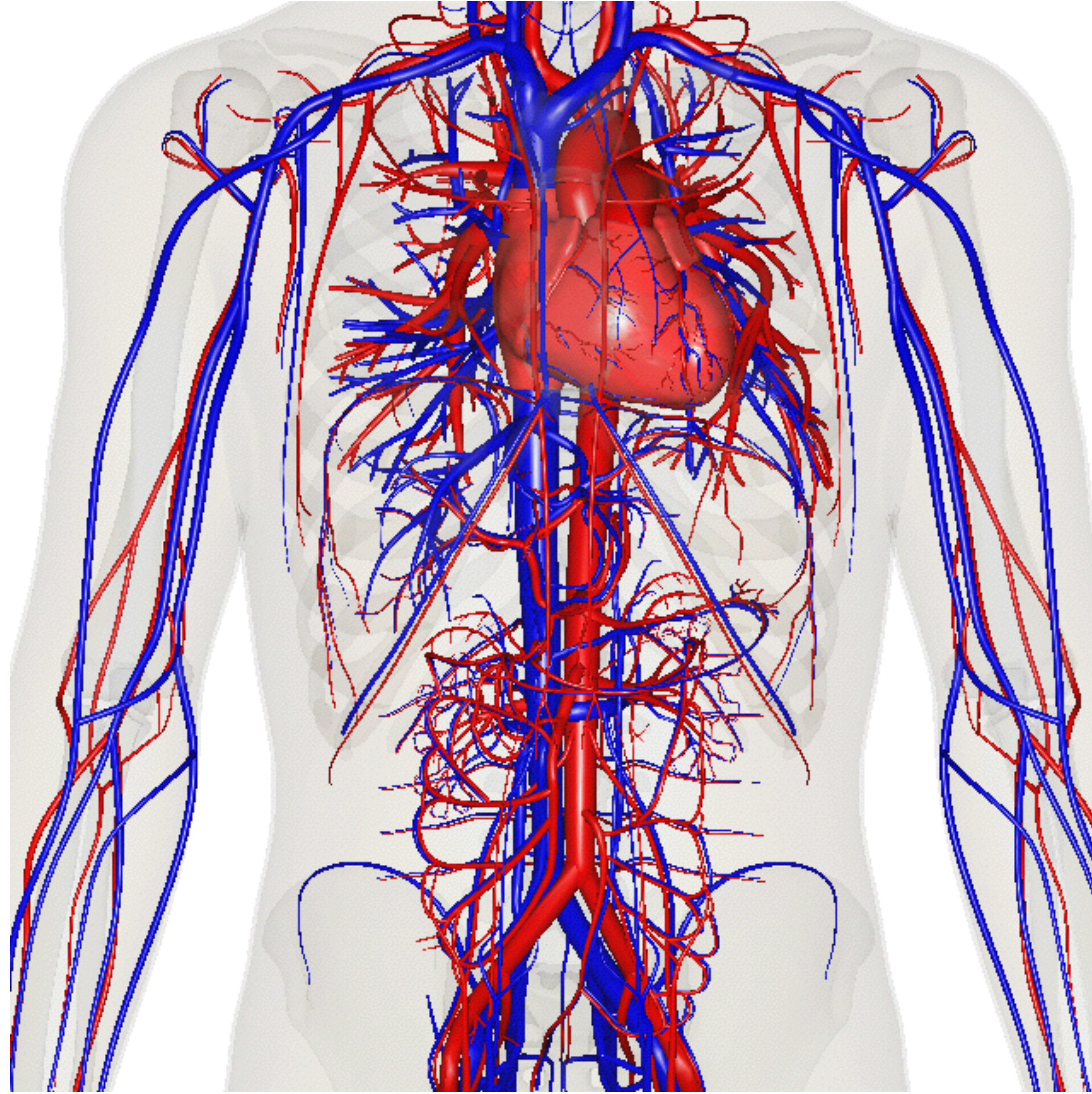
Slows down future work

Harder to onboard new people

Harder to come back to after an absence

Negatively affects response in an emergency



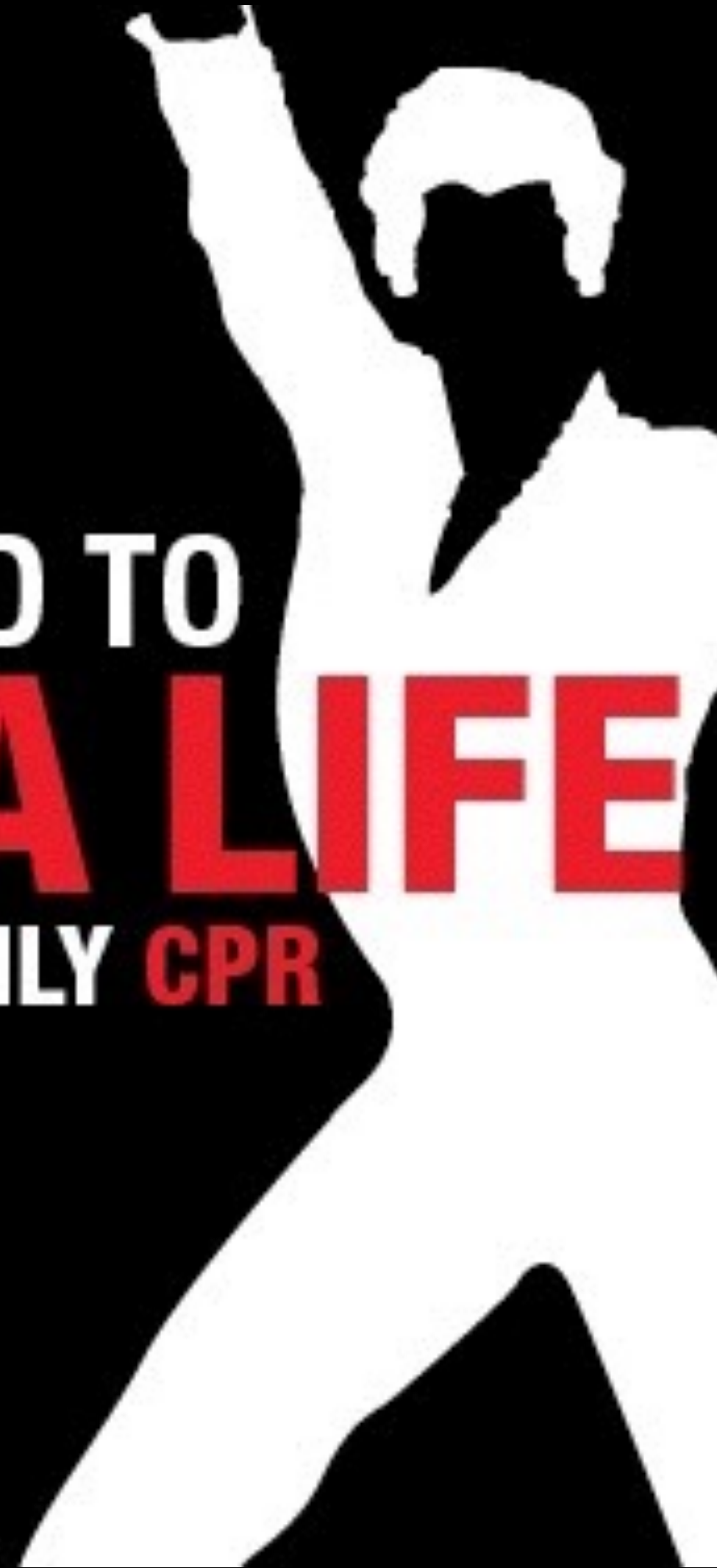








**BE PREPARED TO**  
**SAVE A LIFE**  
**LEARN HANDS-ONLY CPR**





# SIMPLE CODE

---



“The function of good software is to make the complex appear to be simple.”

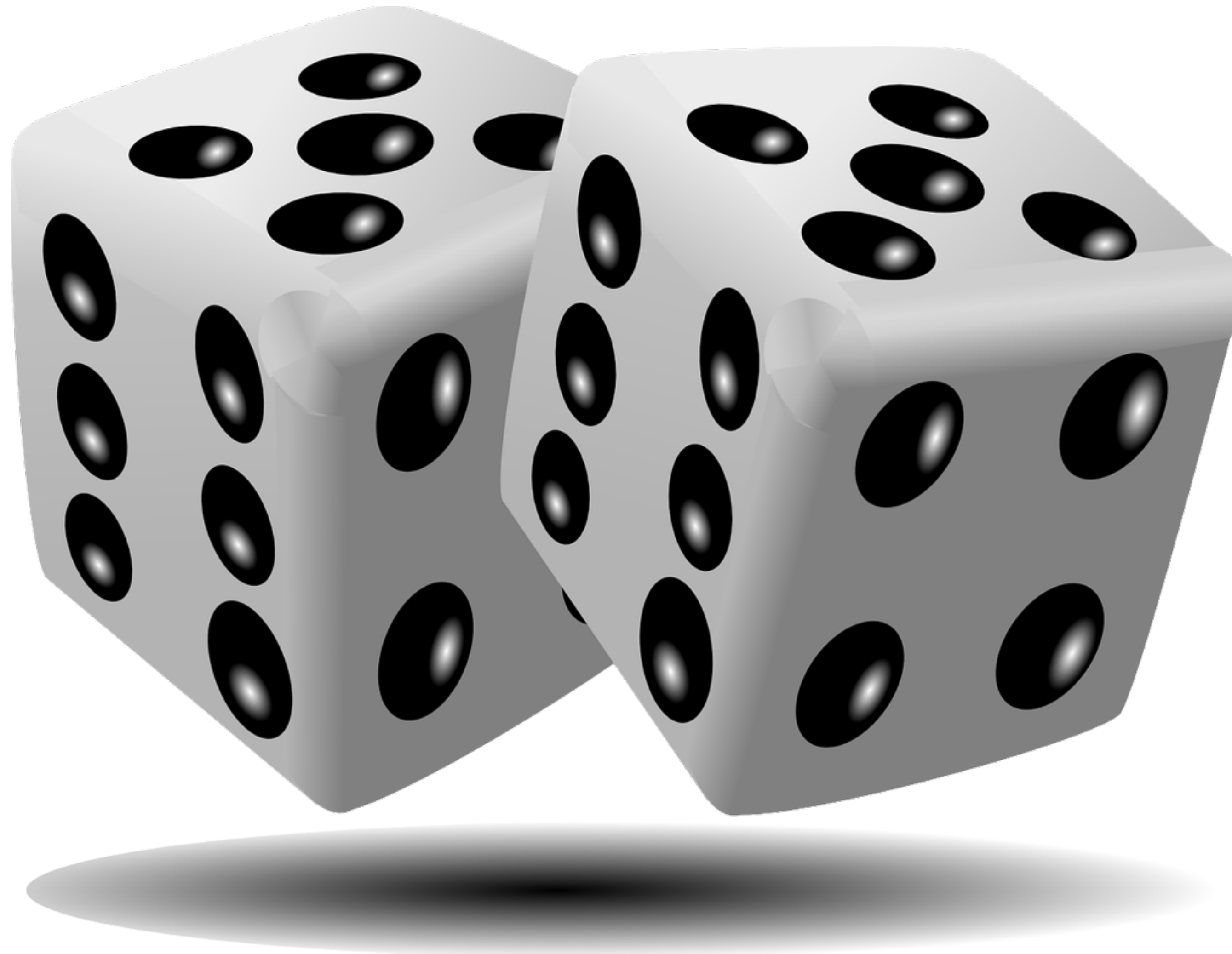
- Grady Booch



---

# WONDERFUL GAME

---



←

→

↺

GitHub, Inc. [US] | https://github.com/starapor/WonderfulGame

☆

ⓘ

📌

C

🔊

🔌

🔌

⋮

Apps

UnoTelly

Suncorp Internet B...

Best Rainy Day Idea...

GMail

🐙

This repository

Search

Pull requests

Issues

Gist

🔔

+ ▾

🔌 ▾

📁

starapor / WonderfulGame

👁 Unwatch ▾

1

★ Star

0

🍴 Fork

0

<> Code

🔔 Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

⚡ Pulse

📊 Graphs

⚙ Settings

Code samples for the presentation "If simple were easy, why is maintenance so hard?" — Edit

🕒 1 commit

🌿 1 branch

🏷 0 releases

👤 1 contributor

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

🔌 starapor Initial commit

Latest commit b61788d 10 minutes ago

📁 WonderfulGame

Initial commit

10 minutes ago

📄 .gitignore

Initial commit

10 minutes ago

📄 WonderfulGame.sln

Initial commit

10 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

© 2016 GitHub, Inc. Terms Privacy Security Status Help

🐙

Contact GitHub API Training Shop Blog About

---

# IF ELSE...

---

```
public void TakeTurn(Player player)
{
    if (player.InPenaltyBox())
        ReducePenaltyTime();
    else
        PlayARound();
}
```



---

## ... GROWS TO ...

---

```
public void TakeTurn(Player player)
{
    if (player.InPenaltyBox())
        ReducePenaltyTime();
    else if (player.WearingAHat())
        RemoveTheHat();
    else
        PlayARound();
}
```

---

# ... BUT MATCHERS ARE BETTER

---

```
{ p => p.InPenaltyBox(), ReducePenaltyTime },  
{ p => p.WearingAHat(), RemoveTheHat },  
{ p => true, PlayARound }
```

---

## ... BUT MATCHERS ARE BETTER

---

```
private readonly Matcher<Player> _matcher;

public Game()
{
    _matcher = new Matcher<Player>
    {
        { p => p.InPenaltyBox(), ReducePenaltyTime },
        { p => p.WearingAHat(), RemoveTheHat },
        { p => true, PlayARound }
    };
}

public void TakeTurn(Player player)
{
    _matcher.DoActionThatMatches(player);
}
```

---

## ... BUT MATCHERS ARE BETTER

---

```
public class Matcher<T> : IEnumerable<KeyValuePair<Predicate<T>, Action>>
{
    private readonly IDictionary<Predicate<T>, Action> _matcher
        = new Dictionary<Predicate<T>, Action>();

    public void DoActionThatMatches(T t)
    {
        _matcher.First(m => m.Key(t)).Value();
    }

    public void Add(Predicate<T> predicate, Action action)
    {
        _matcher.Add(predicate, action);
    }
}
```

---

# PRIMITIVES ...

---

```
public Player(string name)
{
    _name = name;
}
```

---

## ... GROWS TO ...

---

```
public Player(string name, string nickname, string email)
{
    _name = name;
    _nickname = nickname;
    _email = email;
}
```

---

## ... BUT TINY TYPES ARE BETTER

---

```
public Player(Name name, Nickname nickname, Email email)
{
    _name = name;
    _nickname = nickname;
    _email = email;
}
```

---

# TINY TYPES

---

```
public class Name : TinyType<string>
{ public Name(string value) : base(value) { } }
```

```
public class Nickname : TinyType<string>
{ public Nickname(string value) : base(value) { } }
```

```
public class Email : TinyType<string>
{ public Email(string value) : base(value) { } }
```



---

# TINY TYPES

---

```
public class TinyType<T>
{
    private readonly T _value;

    public TinyType(T value)
    {
        _value = value;
    }

    public static implicit operator T(TinyType<T> tt)
    {
        return tt._value;
    }
    public static implicit operator TinyType<T>(T value)
    {
        return new TinyType<T>(value);
    }
}
```

---

# FAT OBJECTS ...

---

```
public interface IRuleHandler
{
    void NotWearingAHatHandler();
}
```

---

## ... GROW TO ...

---

```
public interface IRuleHandler
{
    void NotWearingAHatHandler();
    void SpinAgainHandler();
    void JumpUpAndDownHandler();
    void DoMoreCoolStuffHandler();
}
```

---

## ... BUT SRP IS BETTER

---

```
public interface NotWearingAHatHandler { void Handle(); }
```

```
public interface SpinAgainHandler      { void Handle(); }
```

```
public interface JumpUpAndDownHandler  { void Handle(); }
```

```
public interface DoMoreCoolStuffHandler { void Handle(); }
```

---

# MIXED CONCERNS...

---

```
class Dice : IDice
{
    private readonly Random _random;
    public Dice(Random random) {_random = random;}

    public int Roll()
    {
        var randomNumber = _random.Next(1,6);
        return randomNumber;
    }
}
```

---

## ... GROWS TO ...

---

```
public Dice(Random random, ILogger logger, bool skipEverySecondRoll)
{
    _random = random;
    _logger = logger;
    _skipEverySecondRoll = skipEverySecondRoll;
}

public int Roll()
{
    var randomNumber = _random.Next(1,6);
    _logger.Log("We just rolled {0}", randomNumber);
    if (_skipEverySecondRoll)
    {
        randomNumber = _random.Next(1, 6);
    }
    return randomNumber;
}
```

---

# ... BUT DECORATORS ARE BETTER

---

```
public interface IDice
{
    int Roll();
}
```

---

## ... BUT DECORATORS ARE BETTER

---

```
class Dice : IDice
{
    private readonly Random _random;
    public Dice(Random random)
    {
        _random = random;
    }

    public int Roll()
    {
        return _random.Next(1,6);
    }
}
```



---

## ... BUT DECORATORS ARE BETTER

---

```
class SkipEverySecondRoll : IDice
{
    private readonly IDice _dice;
    public SkipEverySecondRoll(IDice dice)
    {
        _dice = dice;
    }
    public int Roll()
    {
        var ignoreThisValue = _dice.Roll();
        return _dice.Roll();
    }
}
```

---

## ... BUT DECORATORS ARE BETTER

---

```
public DiceLogger(IDice dice, ILogger logger)
{
    _dice = dice;
    _logger = logger;
}

public int Roll()
{
    var rolledNumber = _dice.Roll();
    _logger.Log("We just rolled an awesome random number {0}",
        rolledNumber);
    return rolledNumber;
}
```

---

## ... BUT DECORATORS ARE BETTER

---

```
builder.RegisterType<Dice>().Named<IDice>("real");
```

```
builder.RegisterType<IDice>((c, inner)  
    => new SkipEverySecondRoll(inner),  
    fromKey: "implementor", toKey: "decorator1");
```

```
builder.RegisterType<IDice>((c, inner)  
    => new DiceLogger(inner, new Logger()),  
    fromKey: "decorator1", toKey: null);
```

---

# BOY SCOUT RULE

---



---

# GOOD CITIZEN

---

Trailing commas in lists

```
_matcher = new Matcher<Player>
{
    { p => p.InPenaltyBox(), ReducePenaltyTime },
    { p => p.WearingAHat(), RemoveTheHat },
    { p => true, PlayARound },
};
```

Brackets { } around conditionals

```
public void TakeTurn(Player player)
{
    if (player.InPenaltyBox())
    {
        ReducePenaltyTime();
    }
}
```

---

# SQUINT TEST

---



---

# USE CONVENTIONS UNTIL YOU DON'T

---

IFooHandler => IHandleLotsOfFoos  
IGameRepository => ICanFindTheGame



←

→

↺

GitHub, Inc. [US] | https://github.com/starapor/WonderfulGame

☆

i

📌

C

🔊

🔌

⚙️

⋮


Apps

UnoTelly

Suncorp Internet B...

Best Rainy Day Idea...

GMail



This repository

Search

Pull requests


Issues

Gist

🔔

+ ▾

🔌 ▾

 starapor / WonderfulGame

👁️ Unwatch ▾

1

★ Star

0

🍴 Fork

0

<> Code

🔔 Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

⚡ Pulse

📊 Graphs

⚙️ Settings

Code samples for the presentation "If simple were easy, why is maintenance so hard?" — Edit

🕒 1 commit

🌿 1 branch

🏷️ 0 releases

👤 1 contributor

Branch: master ▾


New pull request

Create new file

Upload files

Find file

Clone or download ▾

 starapor Initial commit


Latest commit b61788d 10 minutes ago

📁 WonderfulGame	Initial commit	10 minutes ago
📄 .gitignore	Initial commit	10 minutes ago
📄 WonderfulGame.sln	Initial commit	10 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

© 2016 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)



[Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)



# SIMPLE EXPLANATIONS

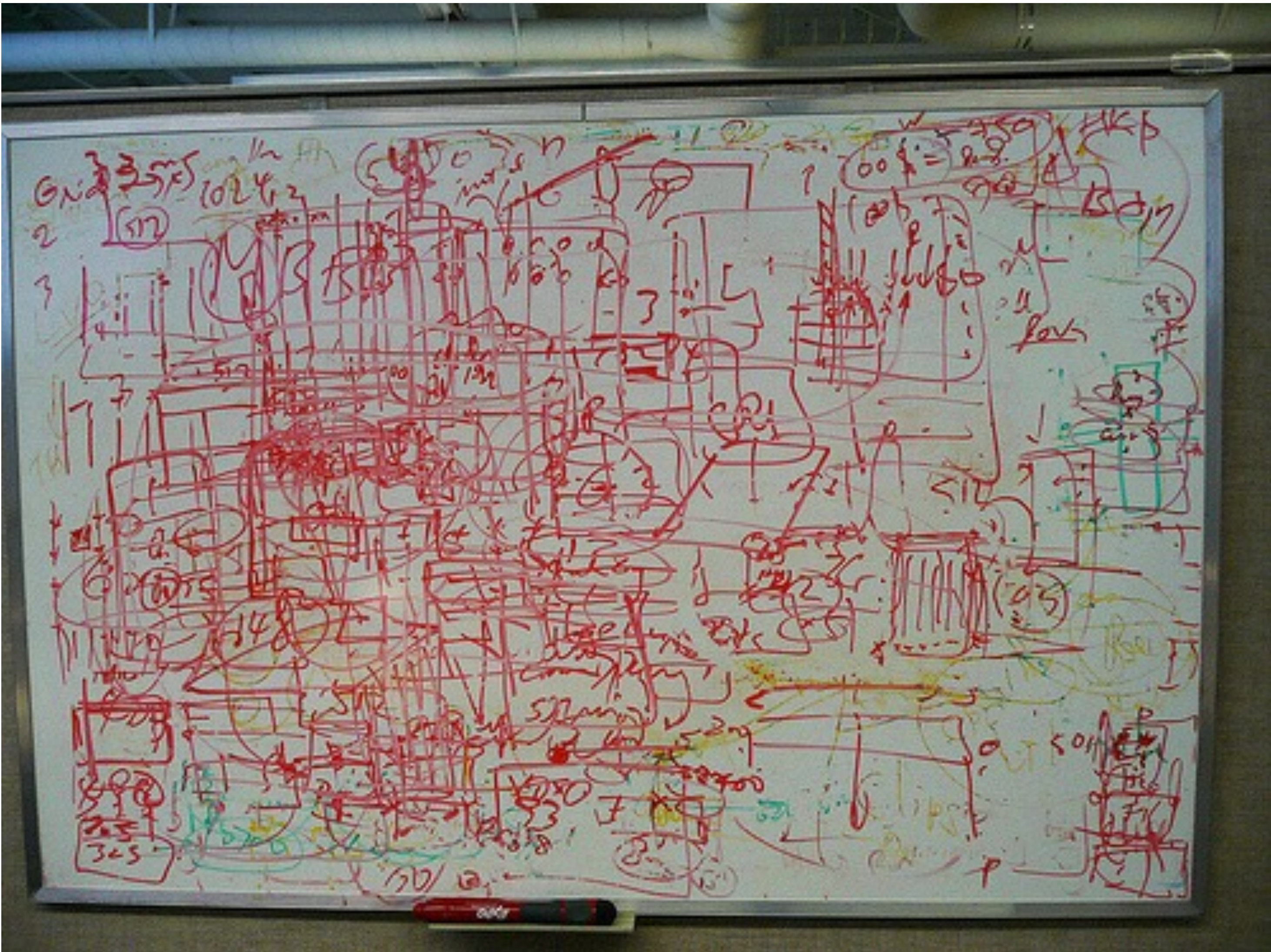
---

“I have made this longer than usual because  
I have not had time to make it shorter.”





# EXPLAIN YOUR SYSTEM





## HOW?

---

Use a whiteboard where possible

Find ways that a developer can relate to the concepts

Not too much, too early

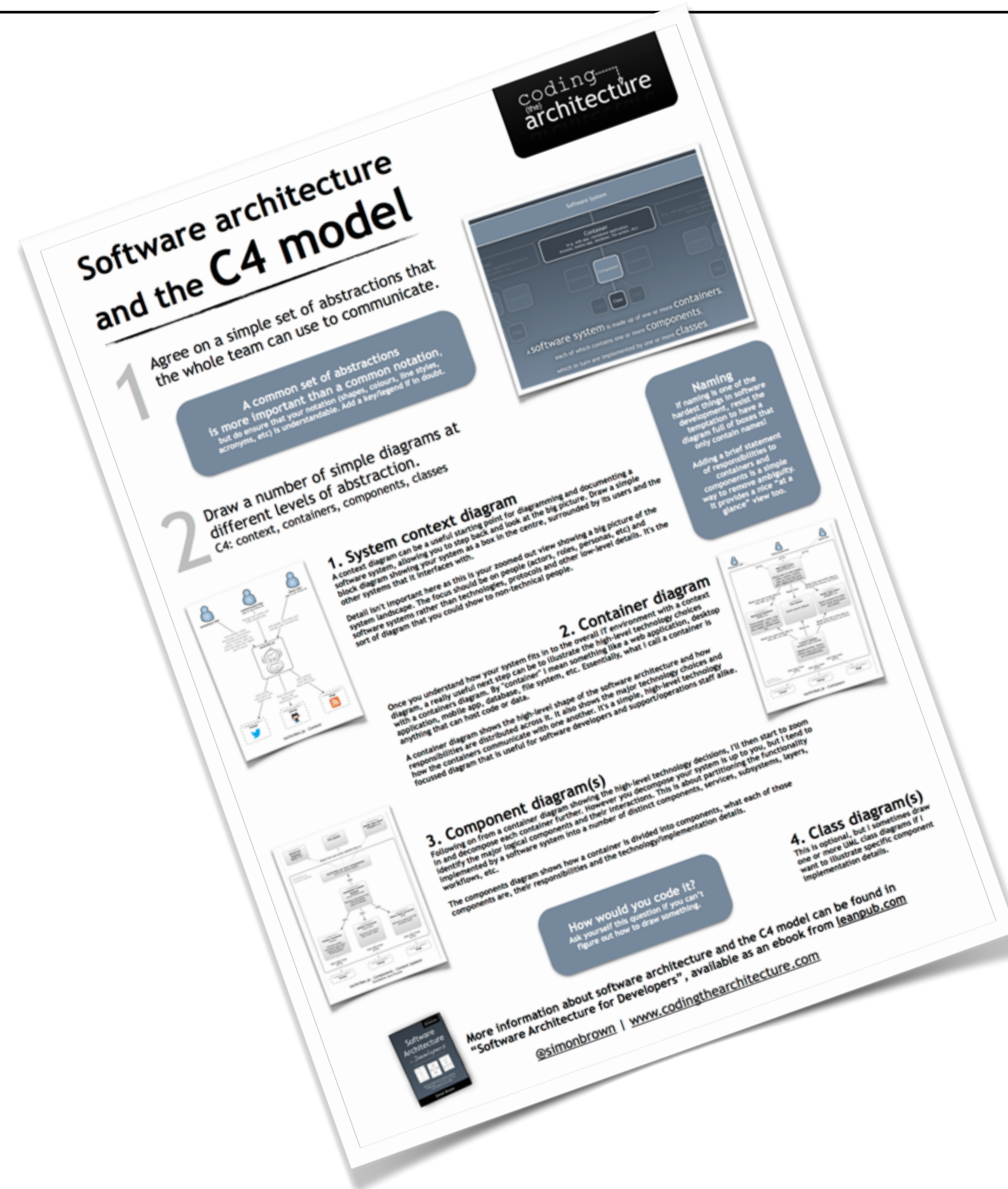
Iterate (revisit this after having had some code)

See Simon Brown's C4 model

Use diagramming/modelling syntax consistently

No need for strict UML - as long as everyone shares the same understanding

# SIMON BROWN'S C4 MODEL



---

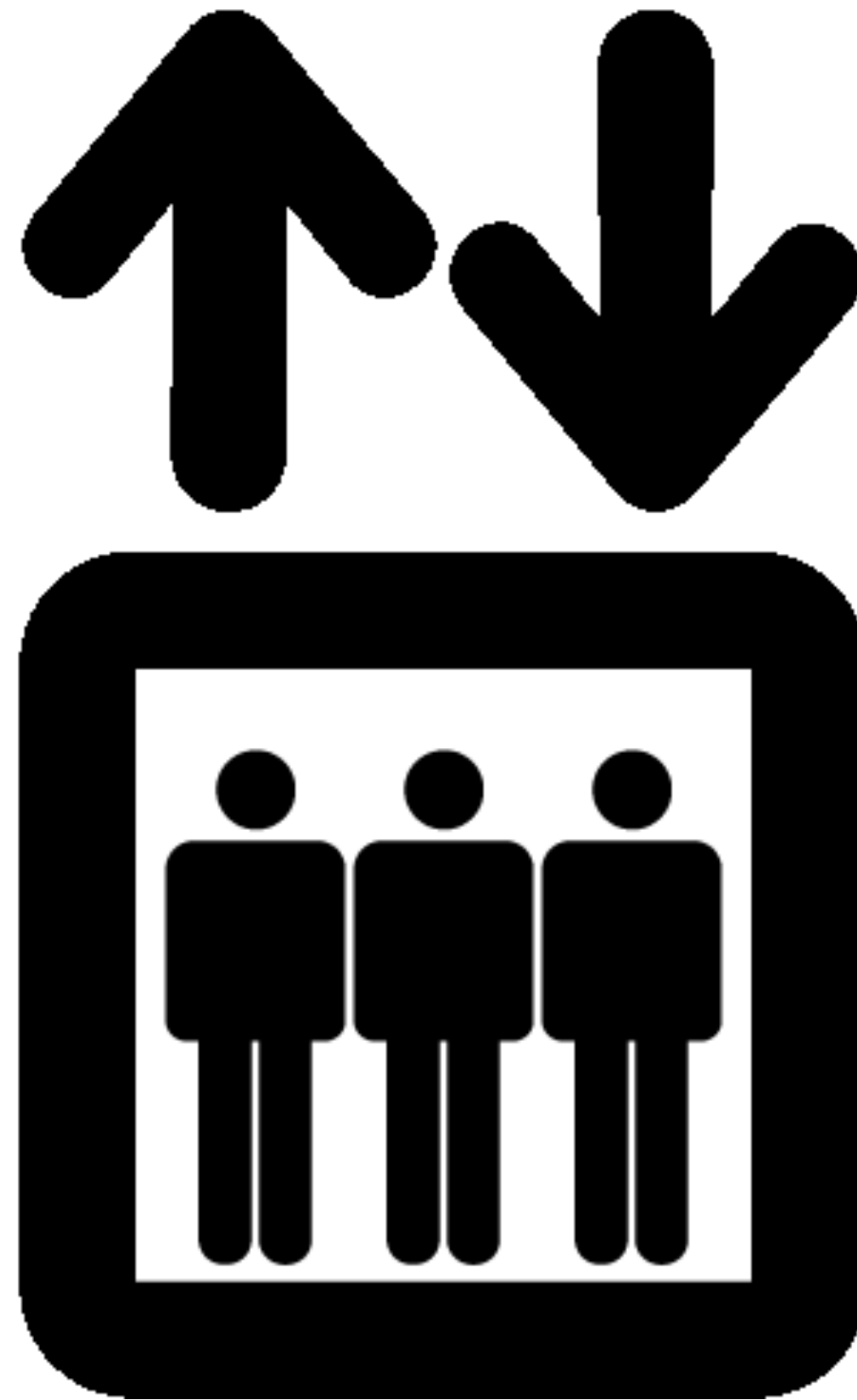
# METAPHORS

---

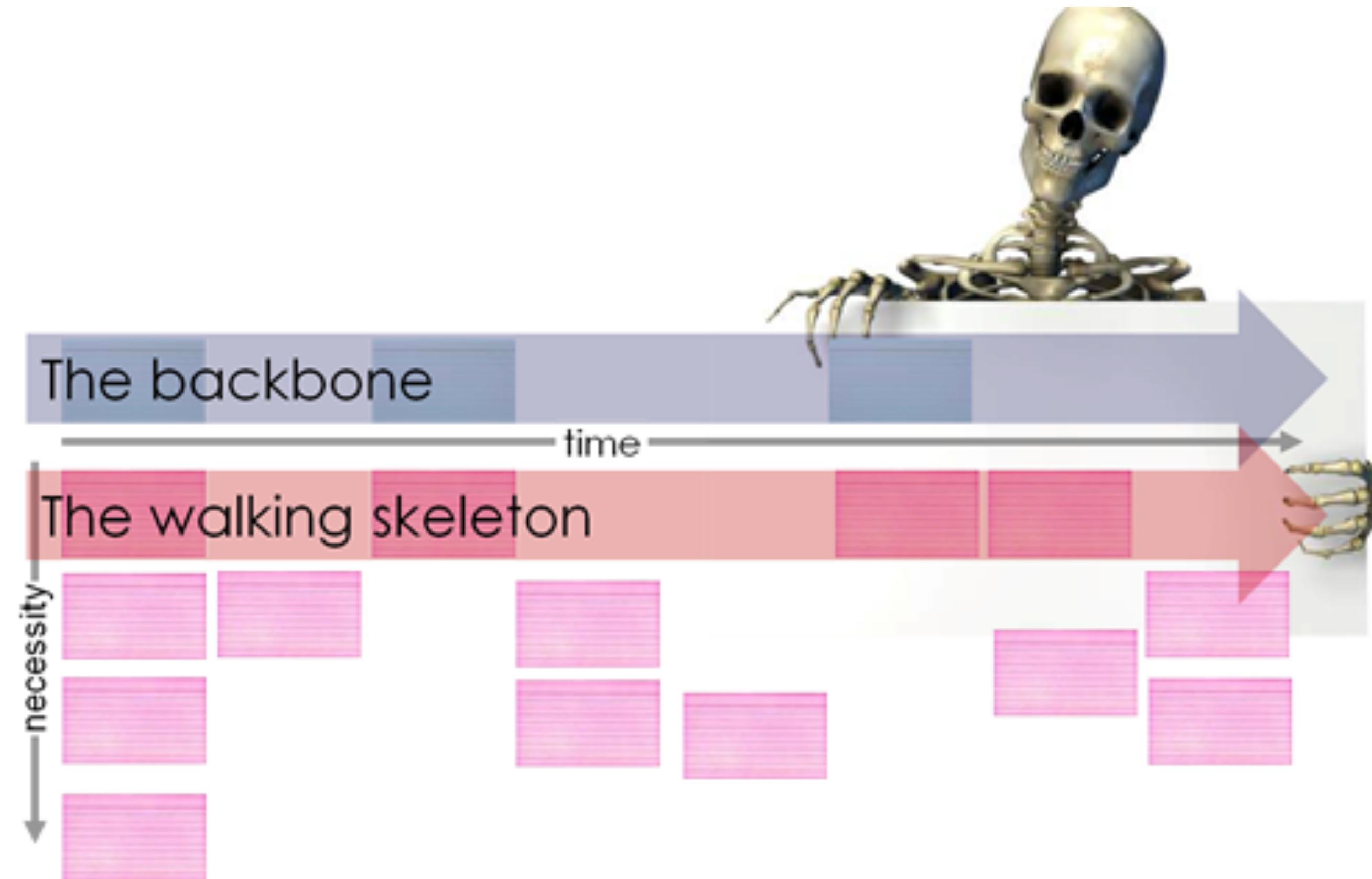
---

# ELEVATOR PITCH

---



# WALKING SKELETON





---

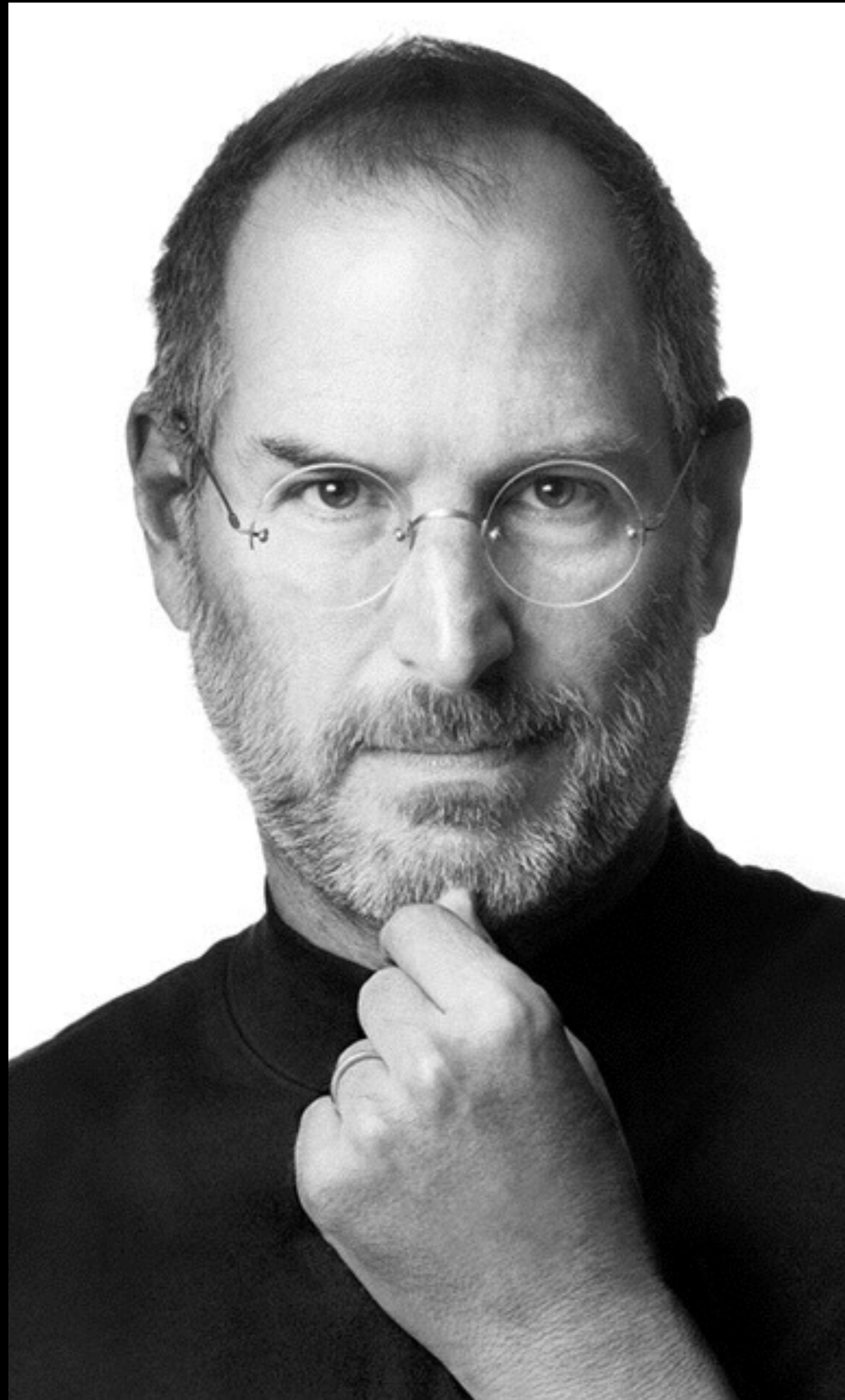
# EXPLAIN TO A 3 YEAR OLD

---



# SUMMARY

---



"That's been one of my mantras -  
focus and simplicity.  
Simple can be harder than complex:  
You have to work hard to get your  
thinking clean to make it simple. But  
it's worth it in the end because once  
you get there, you can move  
mountains."

# TODAY WE EXPLORED

---

Simple and Complex

Simple Code

Simple Explanations



---

# PLEASE SUPPORT OUR SPONSORS

---



NDC { Sydney }  
14-18 August 2017



---

# FILL OUT YOUR FEEDBACK

---

To go into the draw for prizes, please remember to complete your feedback at:

**<http://www.dddbrisbane.com/feedback>**

No feedback = No Prizes!

# THANK YOU

*Sarah Taraporewalla*  
[www.sarahtaraporewalla.com](http://www.sarahtaraporewalla.com)

**ThoughtWorks®**