Stop Writing So Much Code

Coding is hard
You are bad at it

Coding is hard
You are bad at it
I am monumentally lazy

Writing code leads to errors

Bugs

Doing things the wrong way

Bugs

Steve McConnell "Code Complete"

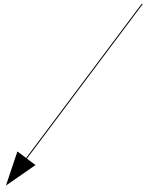
“1 to 5 bugs per 100 lines of code”

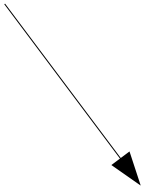
Casper Jones "Applied Software Measurement"

“1.5 bugs per 100 lines of code”

```
sum(x, y)
{
    return(x+y)
}
```

```
sum(x, y)
{
    return(x+y)
}
```





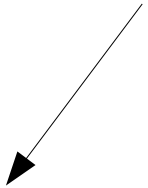
Java, C, C++, Haskell

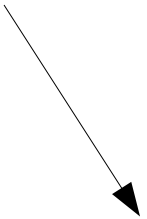
```
public float sum(float x, float y)
{
    return(x+y);
}
```

Ruby, Python, Perl, R, javascript

```
HA HA HA! Actual chaos
```

```
sum(x, y)
{
    return(x+y)
}
```





Java, C, C++, Haskell

```
public float sum(float x, float y)
{
    return(x+y);
}
```

Ruby, Python, Perl, R, javascript

```
sum_floats(x, y)
{
    if ! x.istype('float')
      raise
    if ! y.istype('float')
      raise

    return(x+y)
}

try
    sum(1.2, 4.5)
catch
    break
```

```
def __check_arguments(self, kwargs):
        # flags = (strings,)
        if os.path.isdir(kwargs['tmp_path']):
            self.tmp_path = kwargs.pop('tmp_path', '')
        else:
            raise OSError('tmp_path provided does not exist')

        if isinstance(kwargs['tmp_id'], str):
            self.tmp_id = kwargs.pop('tmp_id', '')
        else:
            raise TypeError('tmp_id must be a string')

        if isinstance(kwargs['command'], str):
            self.command = kwargs.pop('command', '')
        else:
            raise TypeError('command must be a string')

        if 'std_out_str' in kwargs:
            if isinstance(kwargs['std_out_str'], str):
                self.std_out_str = kwargs.pop('std_out_str', '')
            else:
                raise TypeError('std_out_str must be a str')

        if 'input_data' in kwargs:
            if isinstance(kwargs['input_data'], dict):
                self.input_data = kwargs.pop('input_data', '')
            else:
                raise TypeError('input_data must be a dict')
```

Gary Bernhardt
https://www.destroyallsoftware.com/talks/wat

Willingly Doing Things Badly

Choosing to get involved in things you know nothing about

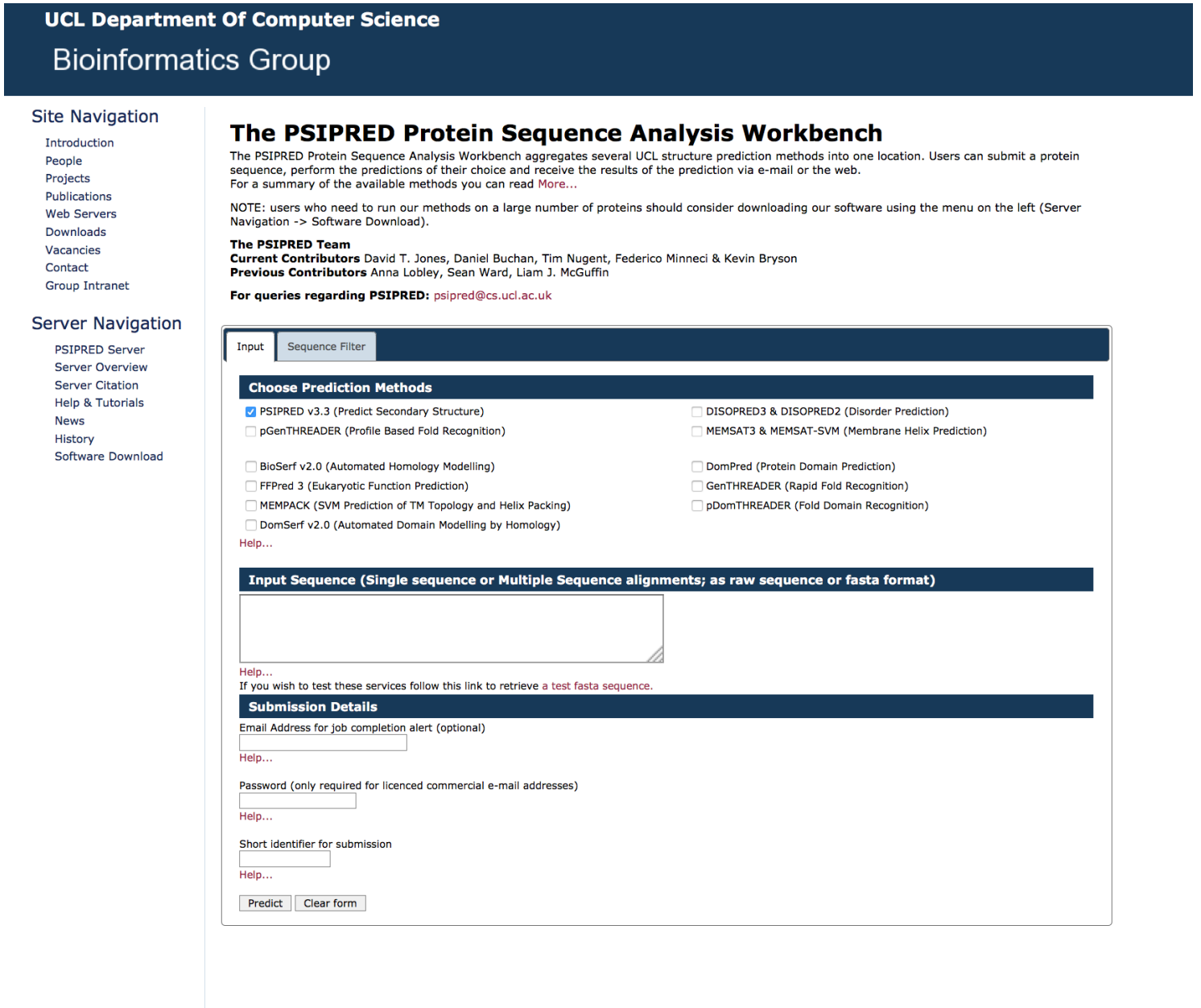
Choosing the wrong language/technology

Things that never save time

Someone else has done it before you

The solution is in a book you've not bothered to open (or even heard of!)

Terrible Thing #1



We know nothing about queuing!!!!

Alternatives

- RabbitMQ/Celery
- Beanstalkd
- GridEngine
- Apache Hadoop
- Apache Spark

...

Writing code leads to errors

Bugs

Doing things the wrong way

Solution

Let other people do the work for you

(while you get the credit!)

Not Invented Here Syndrome

The rejection of other's work in favour of doing it yourself

Leading to…

Why reinvent?

- General distrust of other's work
- Uncertainty of future availability
- Unaware of what is available

External

Terrible Thing #2

I have written an x-fold validation library in PERL

R : Caret
Python : Scikit-Learn
Matlab
Mathematica
...

Why reinvent?

-
-
-
- Coding is fun!
- Reading docs is boring
- Bad at estimating
- Smart people like a challenge
-

Coding!

Reading

Internal

Why reinvent?

- General Distrust of other's work
- Uncertainty of future availability
- Unaware of what is available
- Coding is fun!
- Reading docs is boring
- Bad at estimating
- Smart people like a challenge
- Hubris!

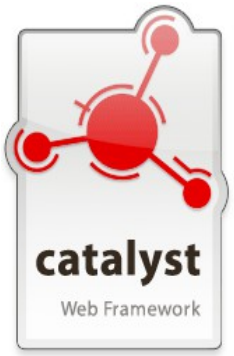
Terrible Thing #3

St Bart's Genome Centre

Laboratory Information Management System

Essentially reimplemented a webframework

Things That Exist



Why reinvent?

- ~~General Distrust of other's work~~
- Uncertainty of future availability
- ~~Unaware of what is available~~
- Coding is fun!
- ~~Reading docs is boring~~
- ~~Smart people like a challenge~~
- ~~Hubris!~~

Benefits of lettings others do the work

- **You write less code**
  - c.f. coding is hard and buggy
- It's quicker to read the docs than build it yourself
- **They are the experts**
- **You can concentrate on the domain of your problem and not on issues you know nothing about**
- **You get to spend more time with puppies and kittens**

Terrible Thing #4

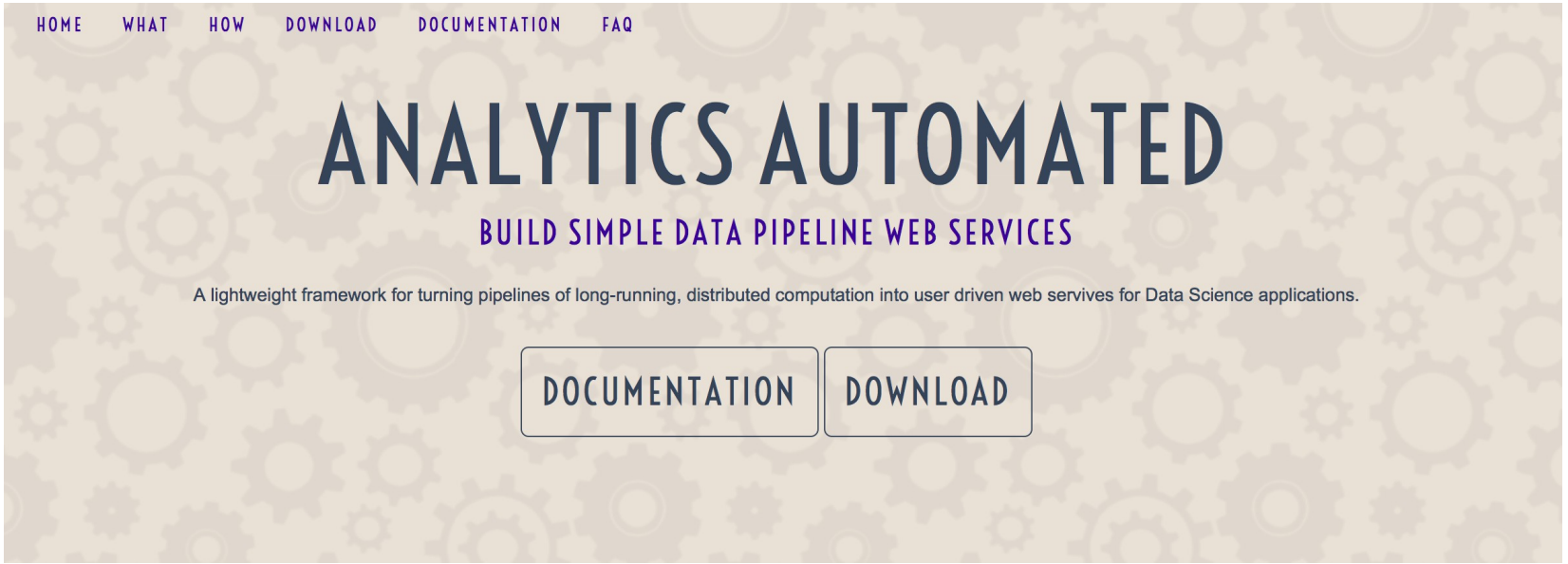
I once tried to use BioPERL

Benefits of lettings others do the work

- **You write less code**
  - c.f. coding is hard and buggy
- It's quicker to read the docs than build it yourself
- **They are the experts**
- **You can concentrate on the domain of your problem and not on issues you know nothing about**
- **You get to spend more time with puppies and kittens**

Using other's work

1000s lines of Java → 600 lines of Python



http://analyticsautomated.github.io

How do I avoid writing code?

- The Download and learn
  - Great CV fodder
- Frameworks; always try to reduce your problem to one of configuration
- Better workflow
  - Language
  - IDE : Eclipse/Netbeans/Atom/SublimeText2/Vi/Emacs
- Work with others
- Code review
- Tests!
- Goodbye XML

When to write code yourself

- There literally is no solution available
  - The available thing does not work
  - The available thing is not maintained
- Future availability is uncertain
- You are provably the world expert on the thing

- UI

In Summary

Less Code == Less Bugs

Less Code == Faster development

Don't make the same mistakes I have