

# Intelligent System Construction

Alexandre Bergel  
University of Chile  
<http://bergel.eu>

“Computer programs are the most complex things that humans make”

— Douglas Crockford, JavaScript: The Good Parts



The European Space Agency took 10 years and \$7bn to produce Ariane 5



The rocket was destroyed after 39 seconds due to **software failure**

KERNENERGIE

# Softwarefout veroorzaakte stilvallen Doel 3

22-04-16, 12.20u - JVH

LEES LATER ★



©ANP



Een softwarefout lag aan de basis van het stilvallen van Doel 3,



A software bug caused Belgian nuclear reactor to switch off during its diagnosis scan



GRUPO LATAM AIRLINES



The software industry is young

Every line of code is a potential point of failure

A software application is like an iceberg:  
90% of it is not visible

Margaret Hamilton, lead software engineer of the Apollo Project, stands next to the code she wrote by hand and that was used to take humanity to the moon. [1969]



<https://github.com/chrislgarry/Apollo-11>



```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```



```
...  
0x0000000107ee62a0: mov    0x8(%rsi),%r10d  
0x0000000107ee62a4: shl    $0x3,%r10  
0x0000000107ee62a8: cmp    %r10,%rax  
0x0000000107ee62ab: jne    0x0000000107ebda60 ; {runtime_call}  
0x0000000107ee62b1: data32 xchg %ax,%ax  
0x0000000107ee62b4: nopl   0x0(%rax,%rax,1)  
0x0000000107ee62bc: data32 data32 xchg %ax,%ax  
0x0000000107ee62c0: mov    %eax,-0x14000(%rsp)  
0x0000000107ee62c7: push   %rbp  
0x0000000107ee62c8: sub    $0x20,%rsp      ;*synchronization entry  
; - java.lang.String::hashCode@-1 (line 1446)  
0x0000000107ee62cc: mov    %rsi,%r11  
0x0000000107ee62cf: mov    0x10(%rsi),%eax ;*getfield hash  
; - java.lang.String::hashCode@1 (line 1446)  
...  
...
```

19 threads 1533 classes  
800K of used RAM

# The reality is scary

---

We heavily rely on software

Software systems are complex, fragile, difficult to produce



# What can we do?



# A simple comparison of development tools

```

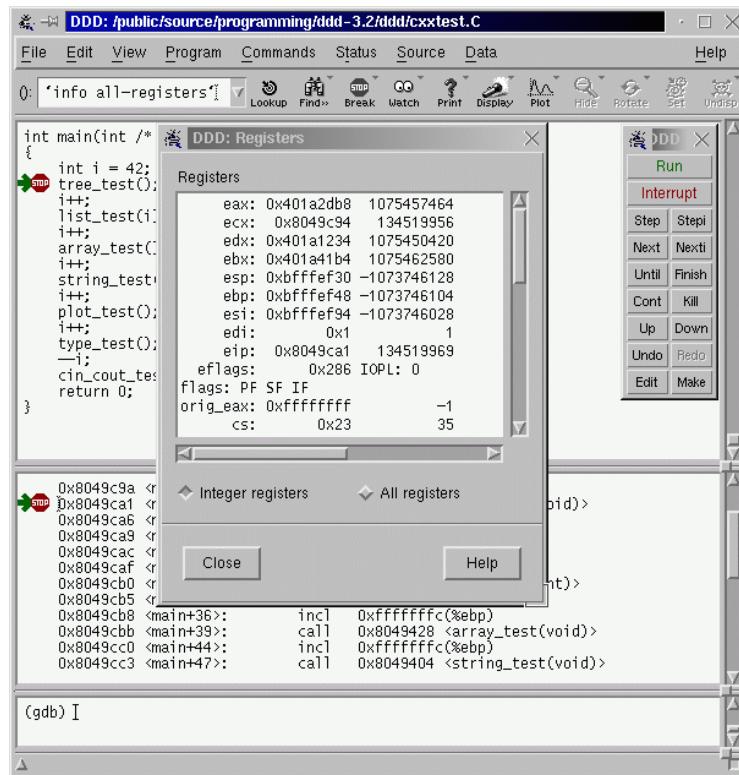
Press F7 for help.          // Copyright (c) Microsoft Open Technologies, Inc. All rights reserved.
.. (up a dir)             // Licensed under the Apache License, Version 2.0. See License.txt in the project root for license information.
<MicrosoftFramework>test
<MicrosoftFramework>api
<MicrosoftFramework>func
DirFree.cs
DirKpmPath.cs
MicrosoftFramework.Fxu
Project.json
TokenManager.cs
+ MicrosoftFramework.Pack
  KomPackTests.cs
  KomPackUtil.cs
  KpmTestUtil.cs
  MicrosoftFramework.Pax
  ProjUtil.cs
+ MicrosoftFramework.Pax
  AssemNeutralFacts.cs
  MicrosoftFramework.Rux
  ProjectTransitionsFacts.cs
  project.json
+ MicrosoftFramework.Runtime
  Something.cs
  var diagnostics = worker.OnGenerate();
  GetHashCode();
  void AssemblyNeutralWorker.Generate();
  GetHashCode();
  int GetHashCode();
  object GetHashCode();
  GetType();
  void AssemblyNeutralWorker.Generate();
  AssemblyNeutralWorker.Generate();
  Assert.Equal("Something", compilations[0].AssemblyName);
  Assert.Equal("Something.IBar", compilations[1].AssemblyName);
  Assert.Equal("Something.IFoo", compilations[2].AssemblyName);
}
{
  [Fact]
  public void CircularReferencesInMembers()
  {
    var worker = DoAssemblyNeutralCompilation();
    var diagnostics = worker.OnGenerate();
    GetHashCode();
    void AssemblyNeutralWorker.Generate();
    GetHashCode();
    int GetHashCode();
    object GetHashCode();
    GetType();
    void AssemblyNeutralWorker.Generate();
    AssemblyNeutralWorker.Generate();
    Assert.Equal("Something", compilations[0].AssemblyName);
    Assert.Equal("Something.IBar", compilations[1].AssemblyName);
    Assert.Equal("Something.IFoo", compilations[2].AssemblyName);
  }
}
namespace Something
{
  [AssemblyNeutral]
  public interface IHttpRequest
  {
    IHttpRequest Context { get; }
    string Verb { get; }
  }
}
namespace Something
{
  [AssemblyNeutral]
  public interface IHttpResponses
  {
    void OnGenerate();
  }
}
  // Copyright (c) Microsoft Open Technologies, Inc. All rights reserved.
  // Licensed under the Apache License, Version 2.0. See License.txt in the project root for license information.
  // model compilation; default directory: ~/src/KRuntime/test/Microsoft.Framework.Runtime.KRoslyn.Tests
  Compilation started at Wed Dec 31 04:12:01
  k test -test Microsoft.Framework.Runtime.Roslyn.Tests.AssemblyNeutralFacts
  Microsoft.Framework.Runtime.Roslyn.Tests.AssemblyNeutralFacts
  xunit.net Project K test runner (64-bit .NET,Asp.NET,WindowsVS6)
  Copyright (C) 2014 Outcurve Foundation, Microsoft Open Technologies, Inc.
  Starting: Microsoft.Framework.Runtime.Roslyn.Tests, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
  Microsoft.Framework.Runtime.Roslyn.Tests, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
  == TEST EXECUTION SUMMARY ==
  Microsoft.Framework.Runtime.Roslyn.Tests, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
  Total: 1, Failed: 0, Skipped: 0, Time: 0.409s
  Compilation finished at Wed Dec 31 04:12:03
  []

```

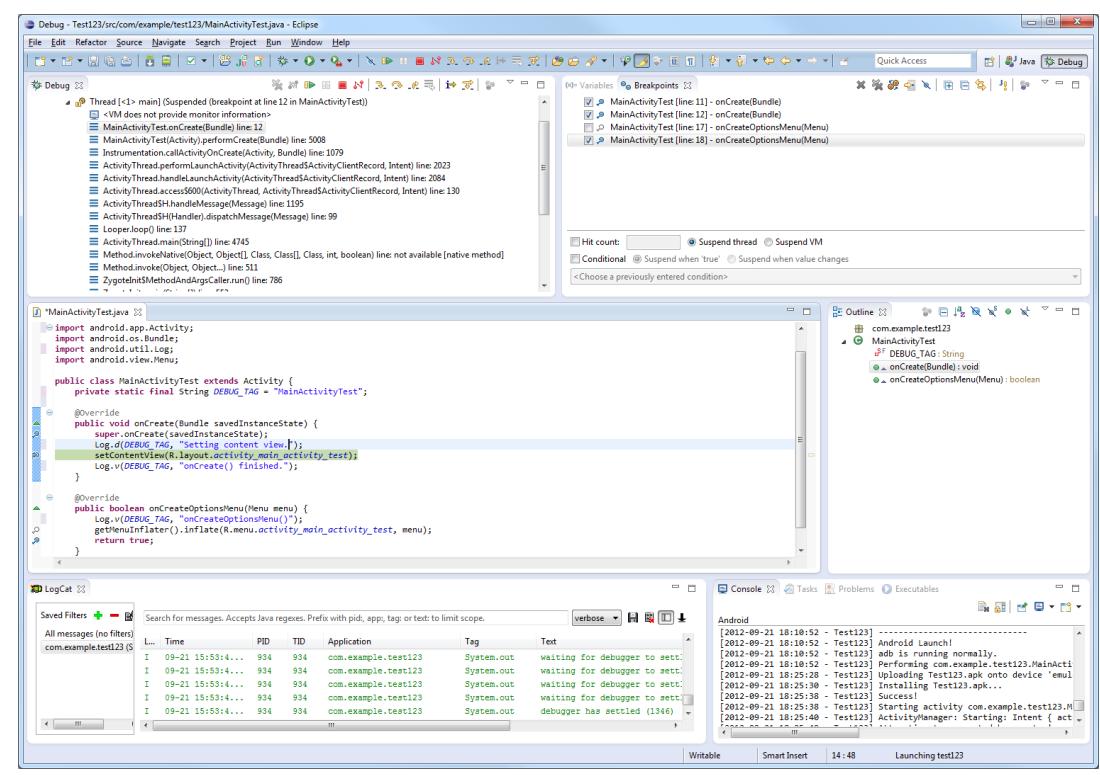
# emacs, 1996

# IntelliJ, today

# A simple comparison of development tools



ddd, 1996



Eclipse debugger, today

Our expertise is in producing better software engineering tools

Code

Issues 1,335

Pull requests 203

Projects 0

Insights

## Adjust expected size of stat memory structure in test given library o...

Browse files

...ptimizations.

Change: 130428863

master (#3866) v1.8.0 ... 0.12.0-rc0

 tensorflower-gardener committed on Aug 16, 2016

1 parent 8a77b23 commit 6205c2605c1367438e827298078312356f10b850

Showing 1 changed file with 2 additions and 2 deletions.

Unified Split

4 tensorflow/core/framework/allocator\_test.cc

View

@@ -67,14 +67,14 @@ TEST(CPUAllocatorTest, Simple) {

67       ptrs.push_back(raw);	67       ptrs.push_back(raw);
68 }	68 }
69       std::sort(ptrs.begin(), ptrs.end());	69       std::sort(ptrs.begin(), ptrs.end());
70 - CheckStats(a, 1023, 553920, 553920, 1024);	70 + CheckStats(a, 1023, 549056, 549056, 1024);
71       for (size_t i = 0; i < ptrs.size(); i++) {	71       for (size_t i = 0; i < ptrs.size(); i++) {
72           if (i > 0) {	72           if (i > 0) {
73               CHECK_NE(ptrs[i], ptrs[i - 1]); // No dups	73               CHECK_NE(ptrs[i], ptrs[i - 1]); // No dups
74       }	74       }
75       a->DeallocateRaw(ptrs[i]);	75       a->DeallocateRaw(ptrs[i]);
76     }	76     }
77 - CheckStats(a, 1023, 0, 553920, 1024);	77 + CheckStats(a, 1023, 0, 549056, 1024);
78       float* t1 = a->Allocate<float>(1024);	78       float* t1 = a->Allocate<float>(1024);
79       double* t2 = a->Allocate<double>(1048576);	79       double* t2 = a->Allocate<double>(1048576);
80       CheckStats(a, 1025, 1048576 * sizeof(double) + 1024 * sizeof(float),	80       CheckStats(a, 1025, 1048576 * sizeof(double) + 1024 * sizeof(float),

# Software visualization

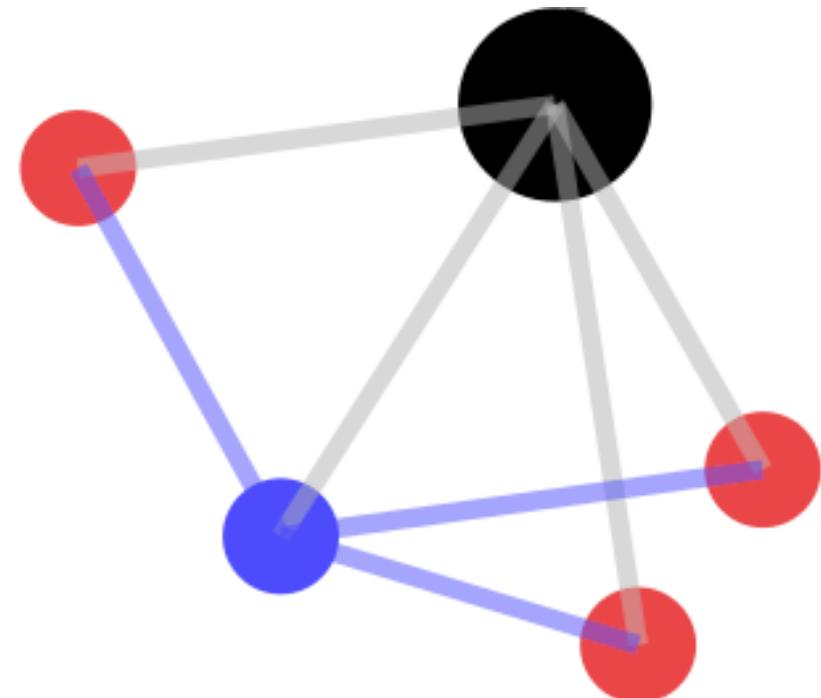
---

```
public class Counter {  
    private int value;  
    public Counter() { value = 0; }  
    public void increase() { value++; }  
    public void decrease() { value--; }  
}
```

# Software visualization

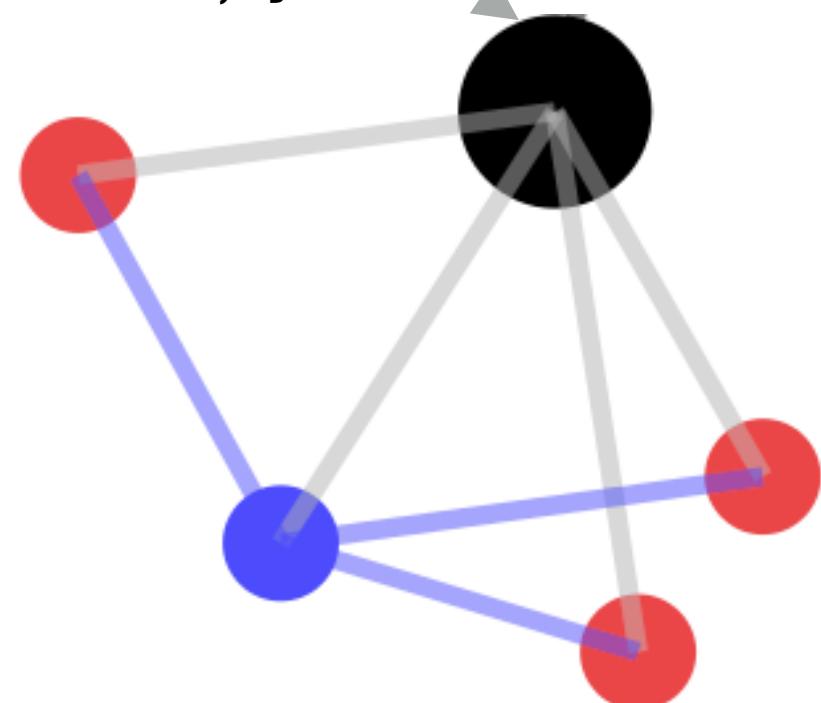
---

```
public class Counter {  
    private int value;  
    public Counter() { value = 0; }  
    public void increase() { value++; }  
    public void decrease() { value--; }  
}
```



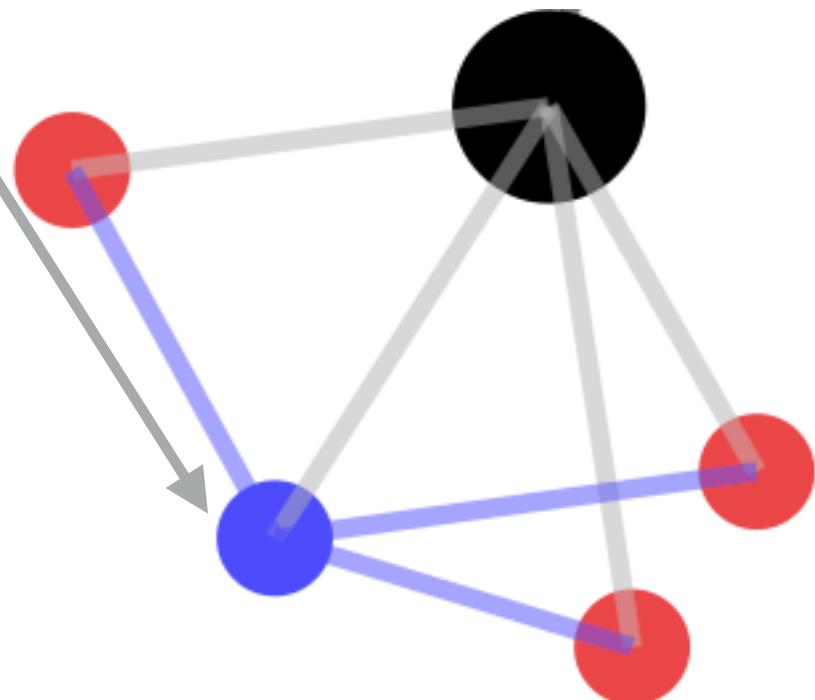
# Software visualization

```
public class Counter {  
    private int value;  
    public Counter() { value = 0; }  
    public void increase() { value++; }  
    public void decrease() { value--; }  
}
```



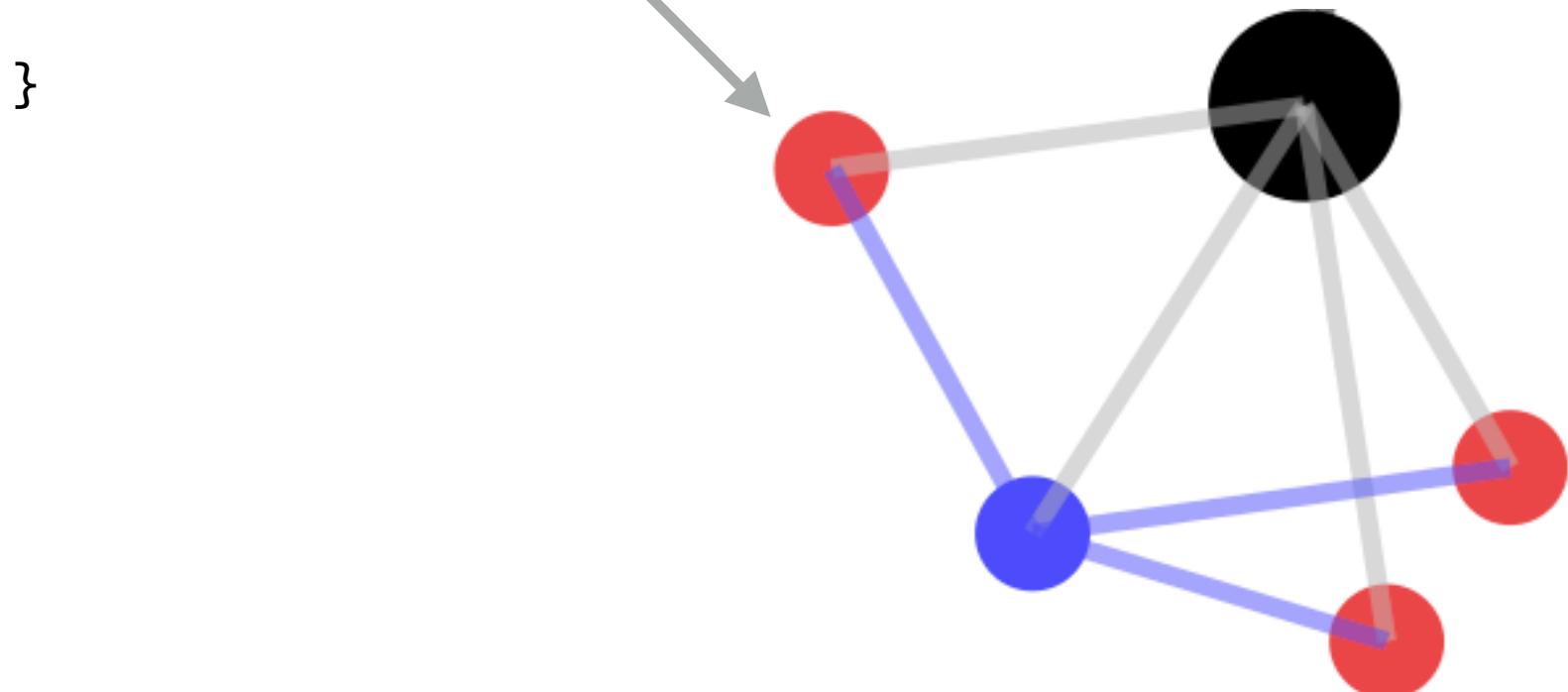
# Software visualization

```
public class Counter {  
    private int value;  
    public Counter() { value = 0; }  
    public void increase() { value++; }  
    public void decrease() { value--; }  
}
```



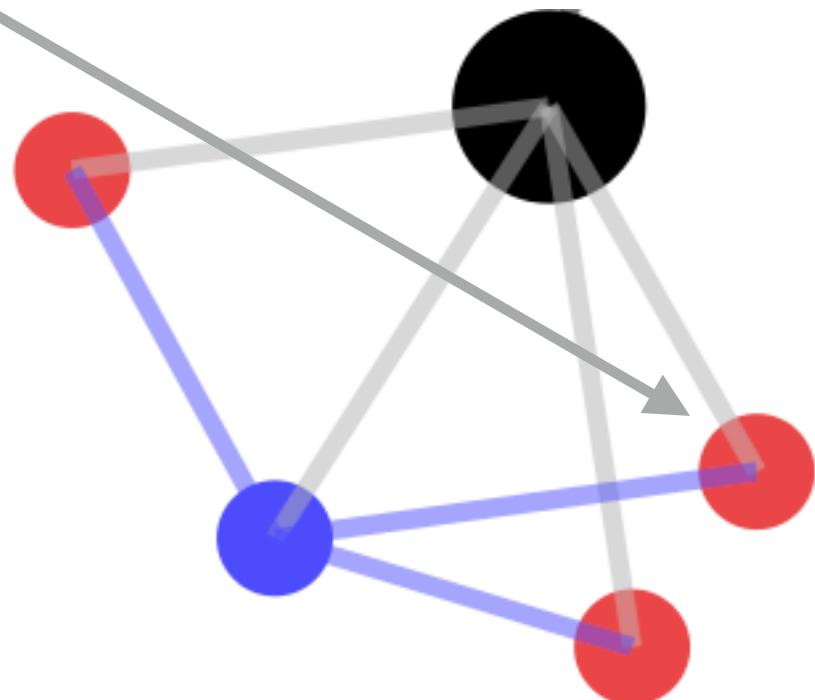
# Software visualization

```
public class Counter {  
    private int value;  
    public Counter() { value = 0; }  
    public void increase() { value++; }  
    public void decrease() { value--; }  
}
```



# Software visualization

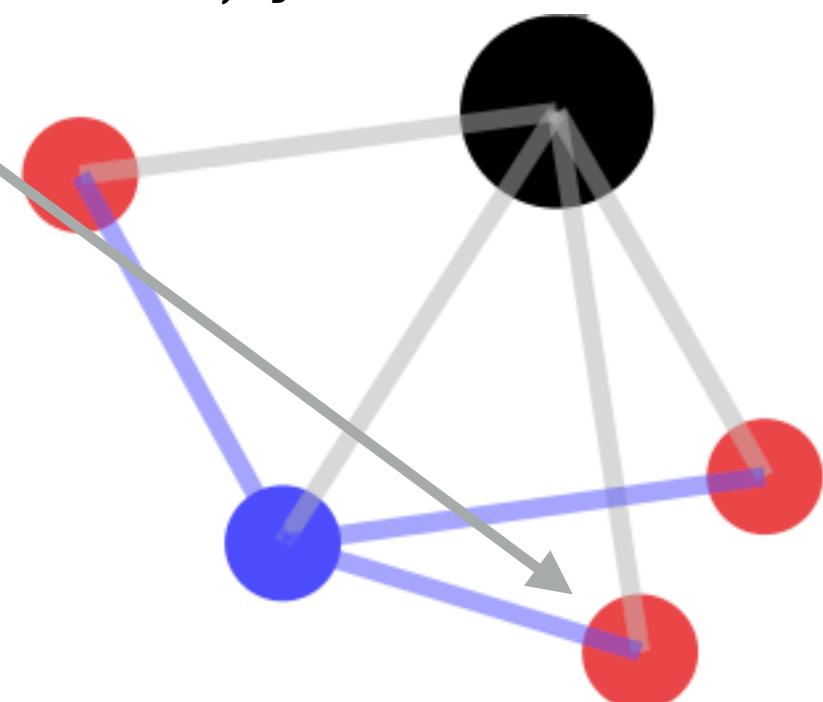
```
public class Counter {  
    private int value;  
    public Counter() { value = 0; }  
    public void increase() { value++; }  
    public void decrease() { value--; }  
}
```



# Software visualization

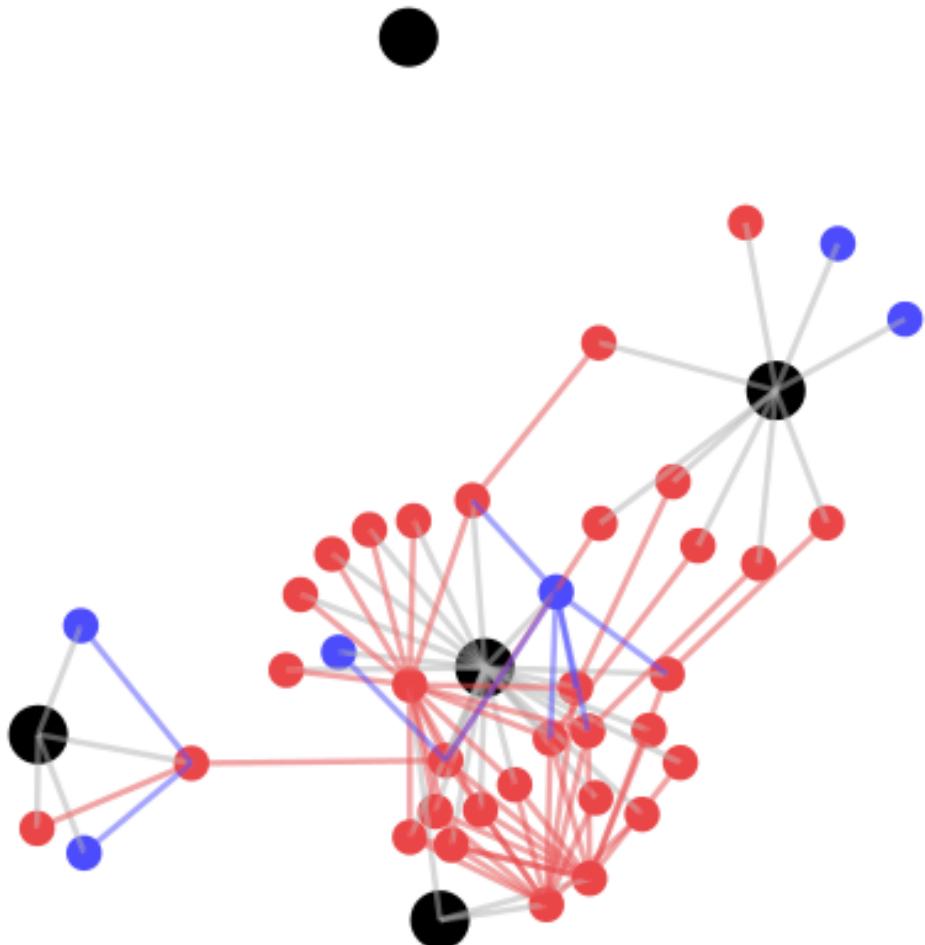
---

```
public class Counter {  
    private int value;  
    public Counter() { value = 0; }  
    public void increase() { value++; }  
    public void decrease() { value--; }  
}
```



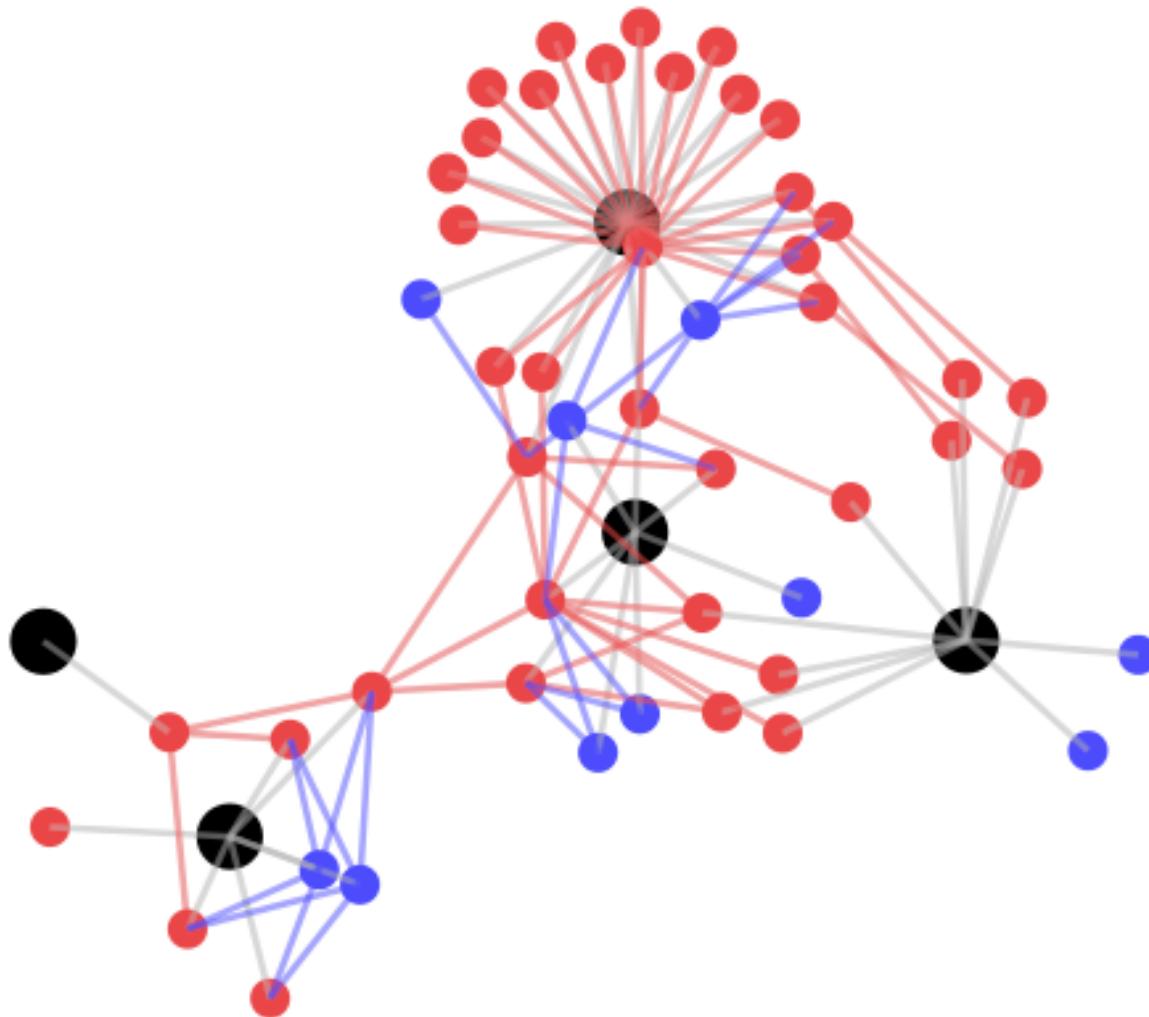
# Tareas of this semester

---



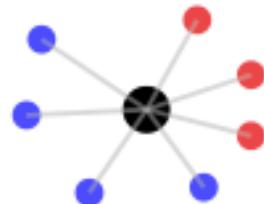
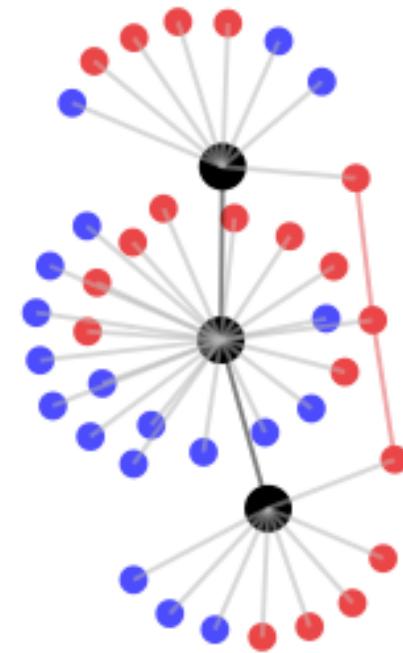
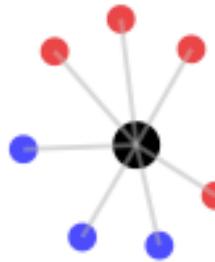
# Tareas of this semester

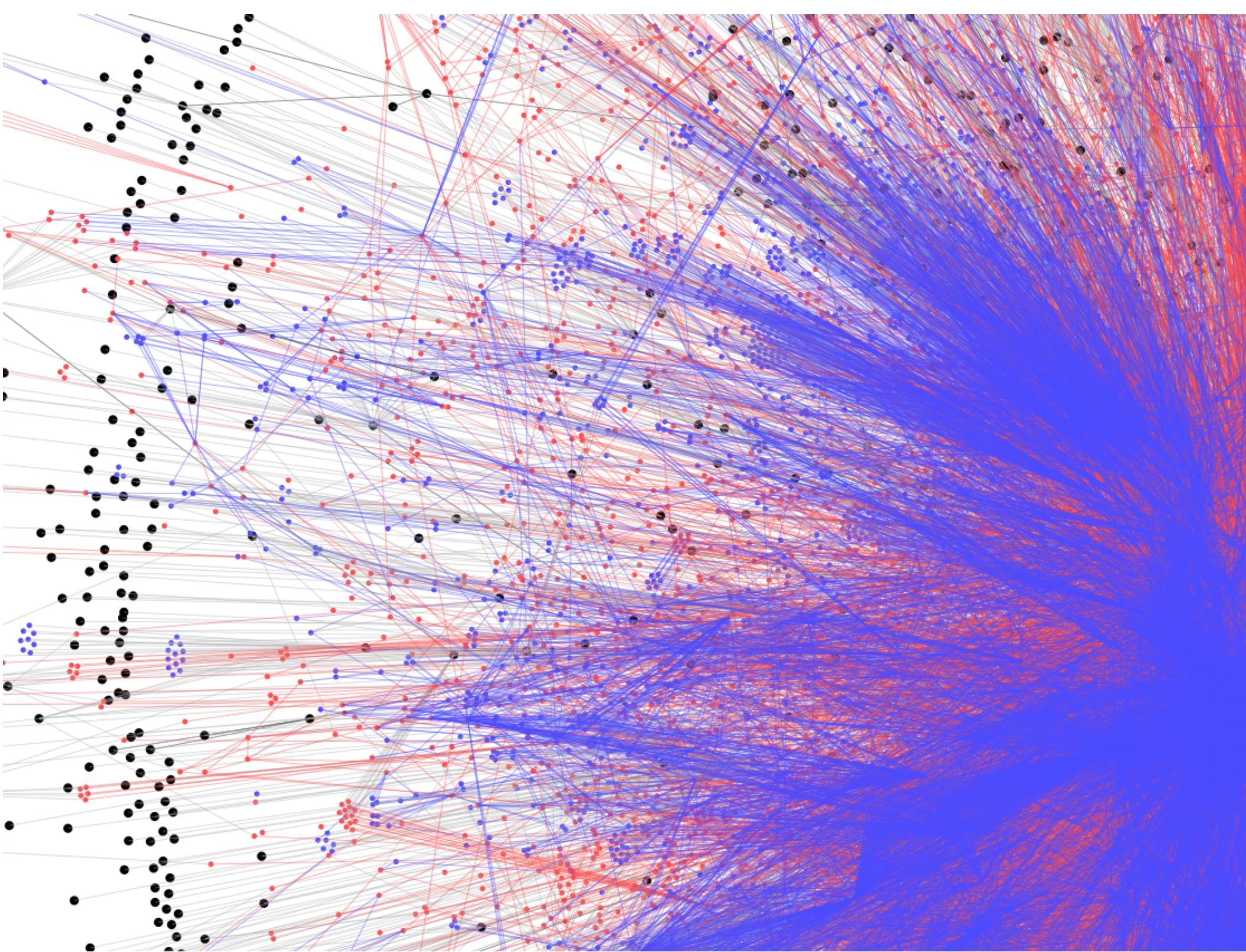
---

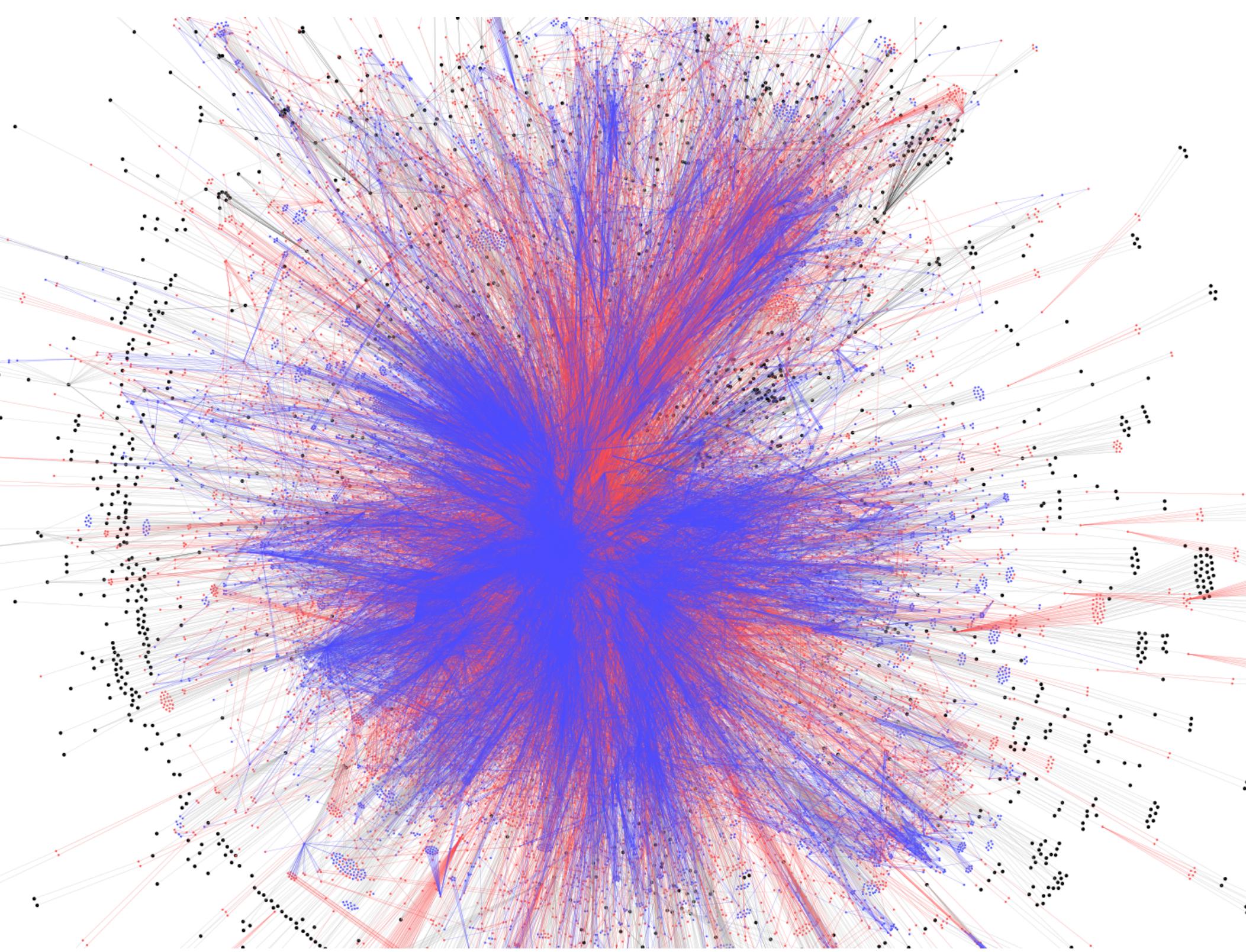


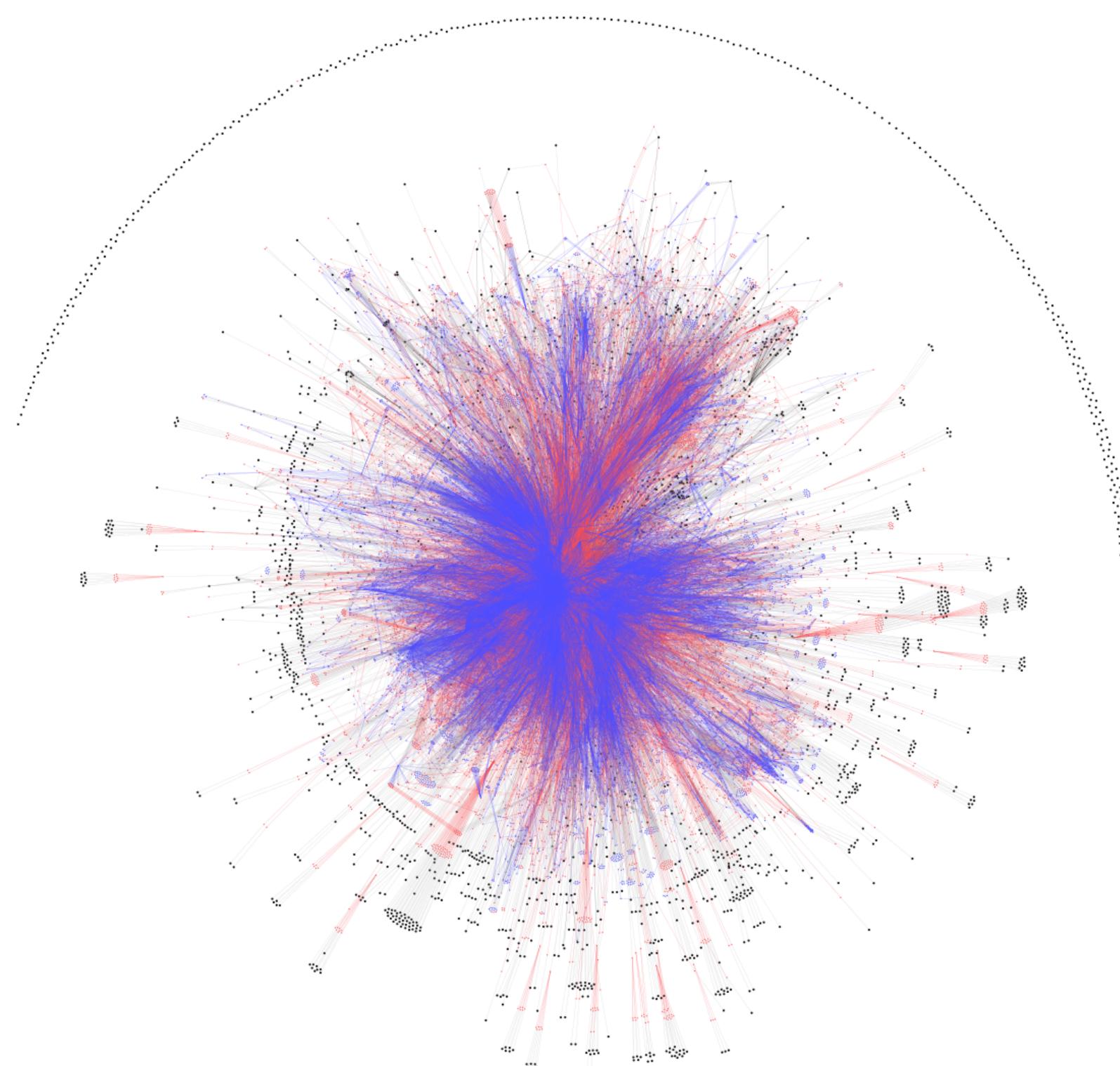
# Tareas of this semester

---

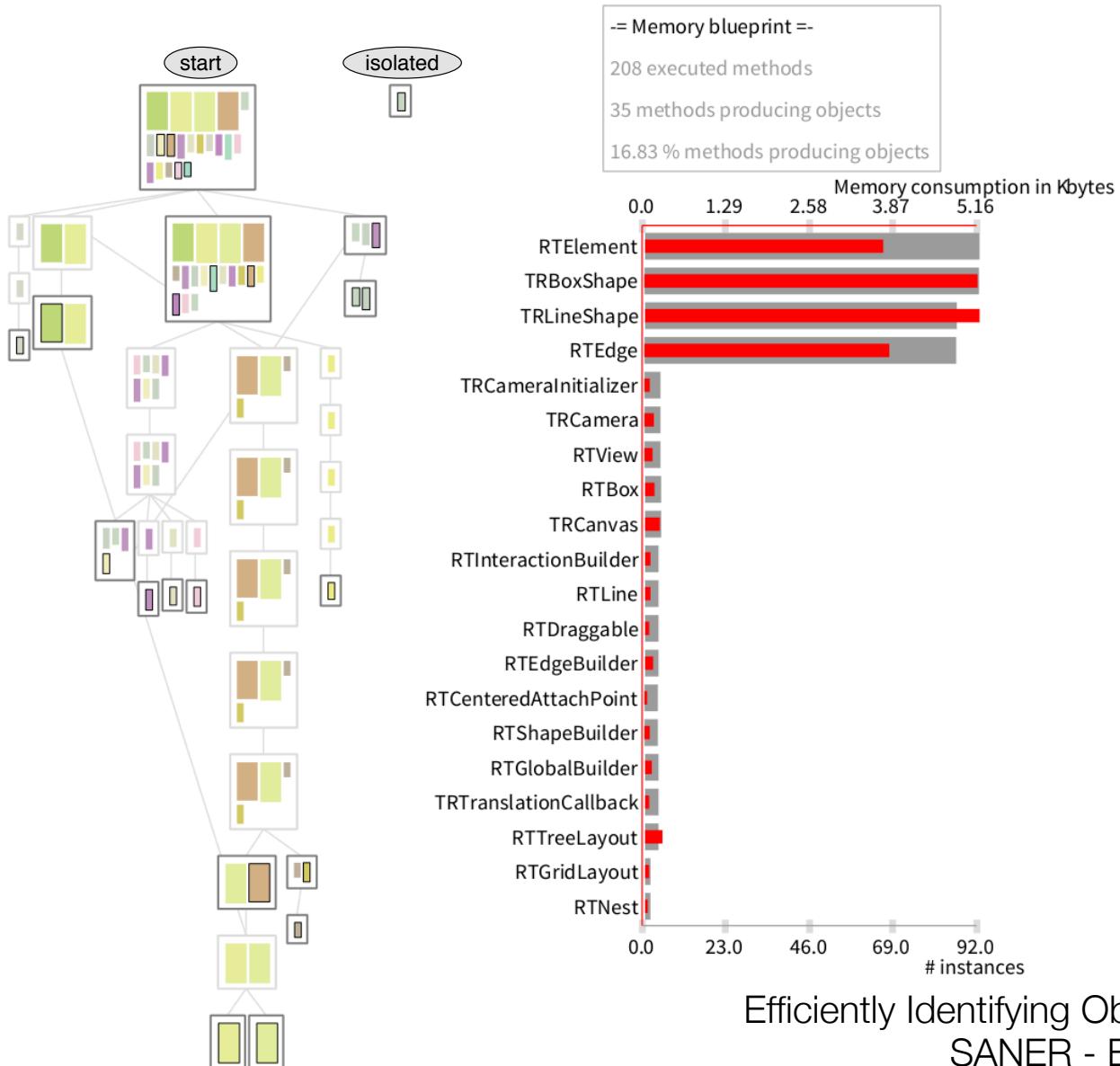


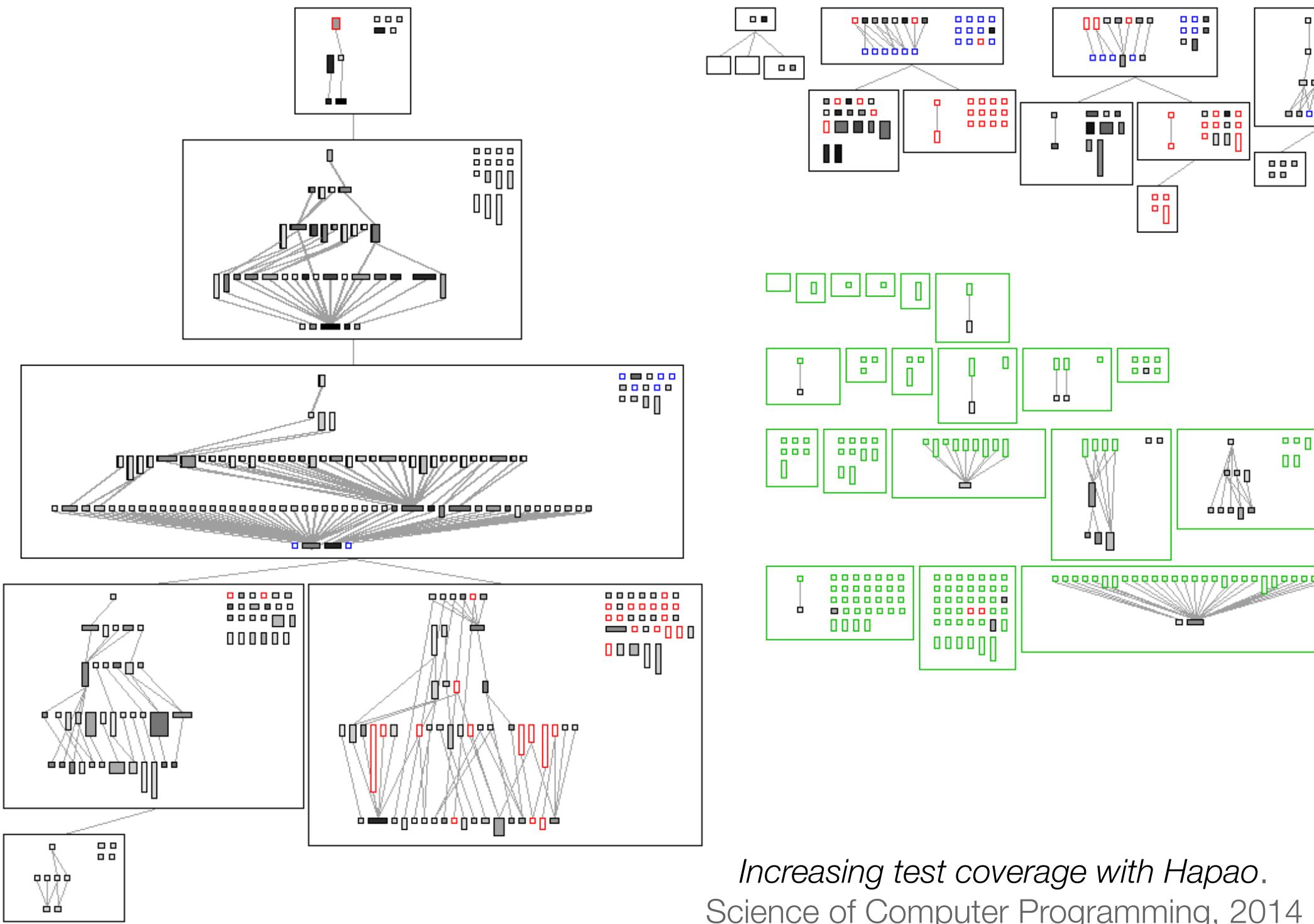




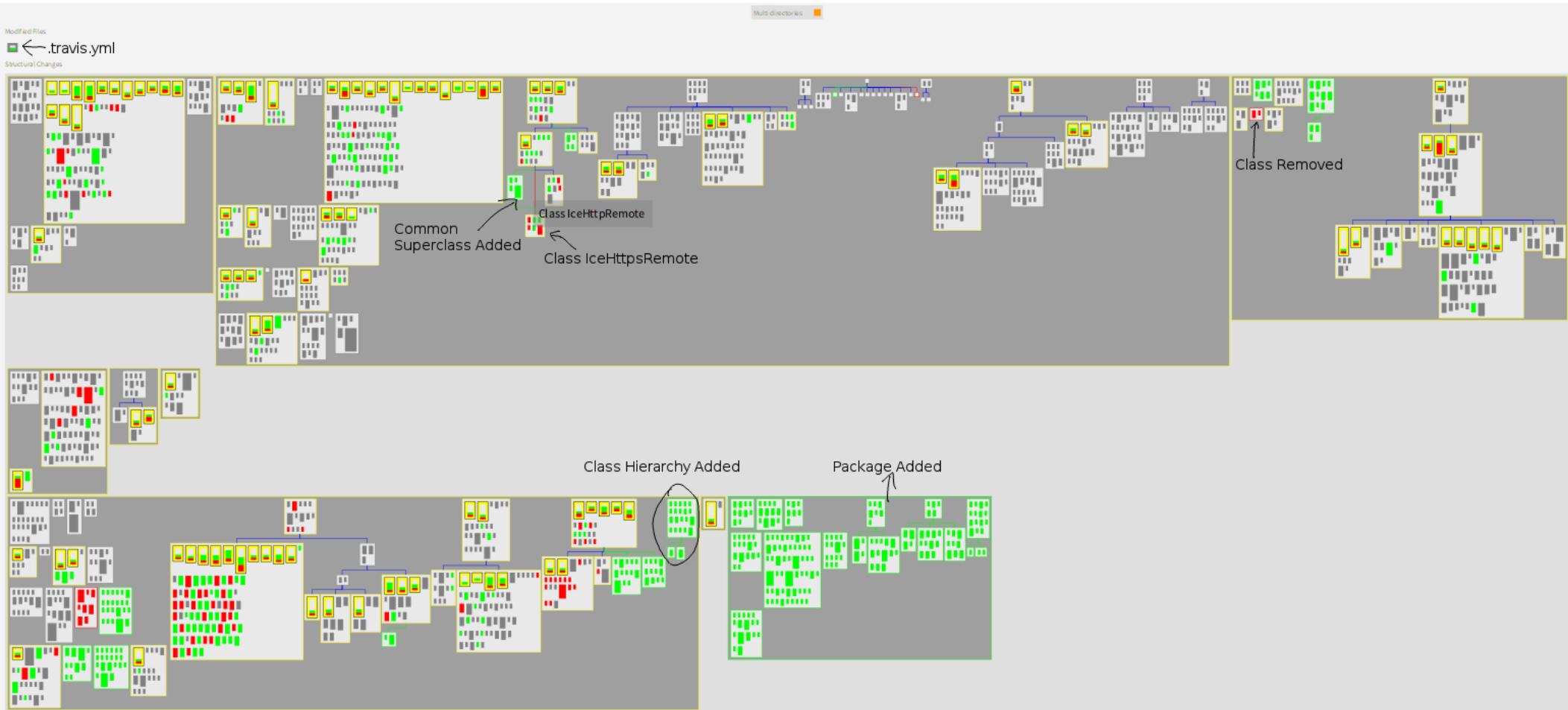


# Memory profiling

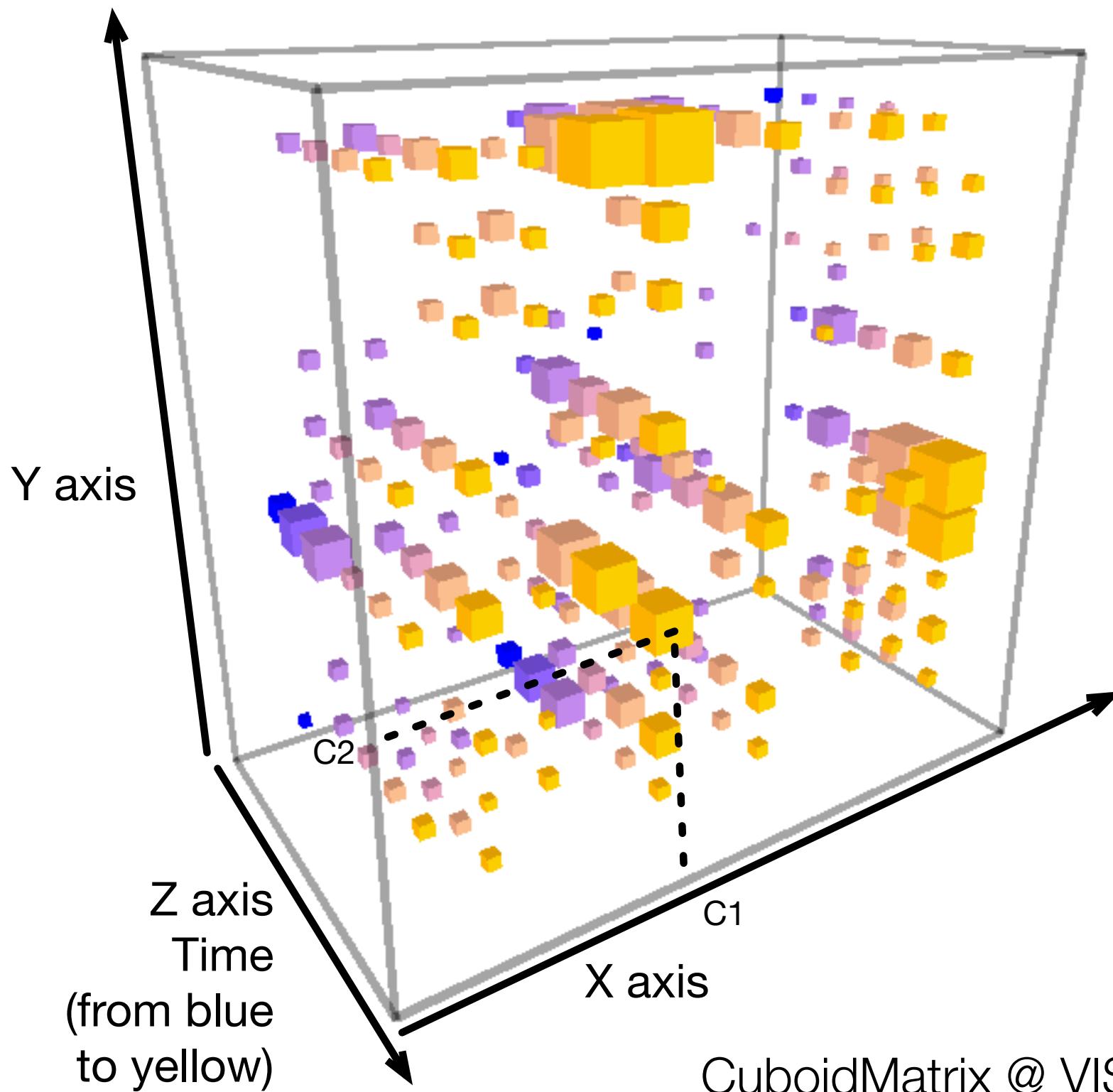




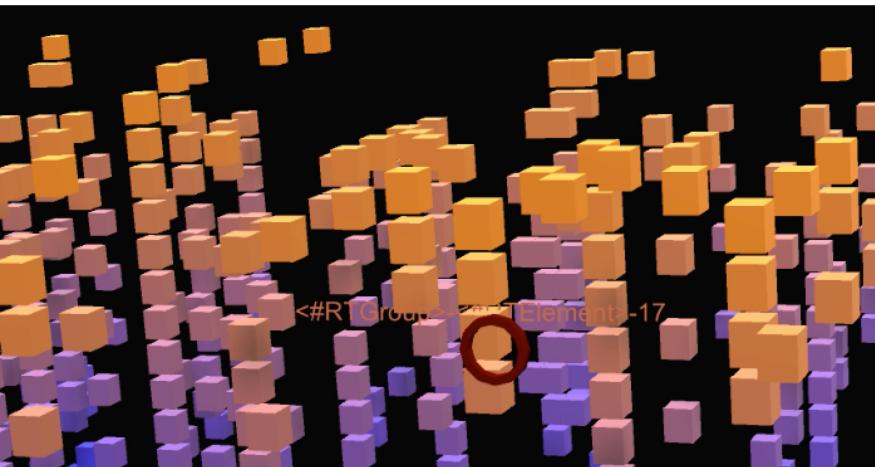
*Increasing test coverage with Hapao.*  
Science of Computer Programming, 2014



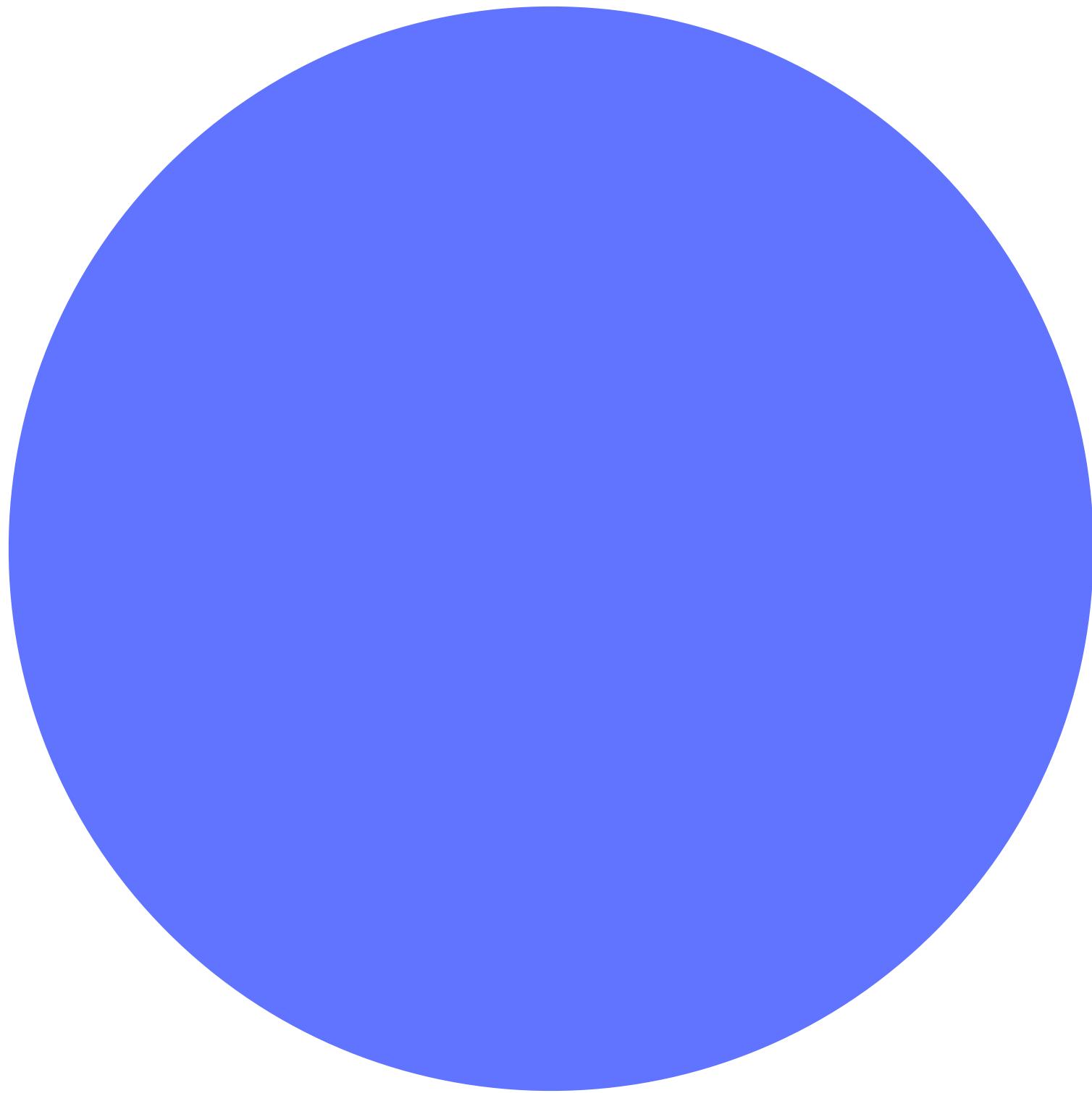
Visualizing GIT pull-request  
To be submitted

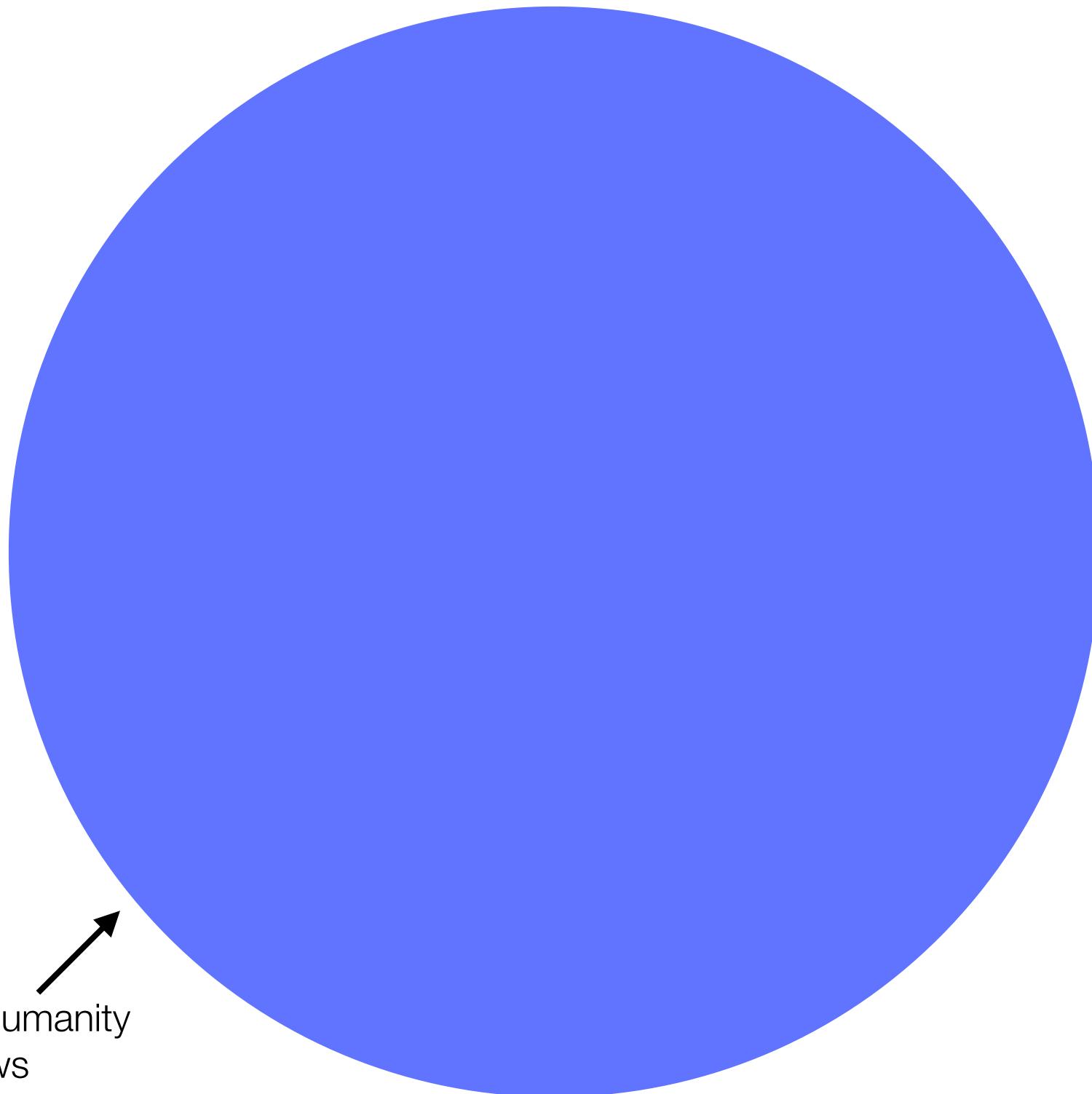


CuboidMatrix @ VISSOFT'16

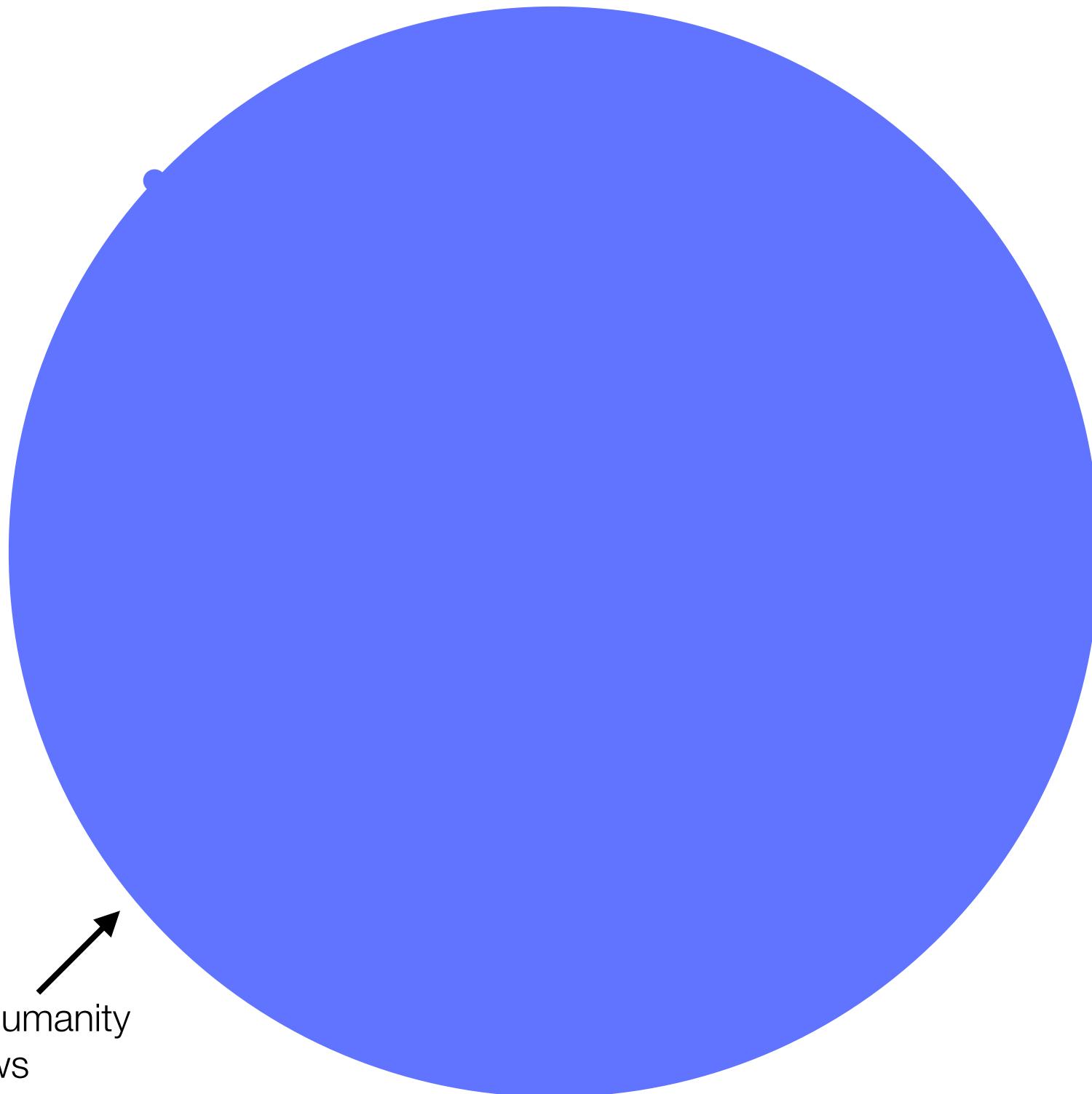


Augmented reality in a  
programming environment  
To be submitted



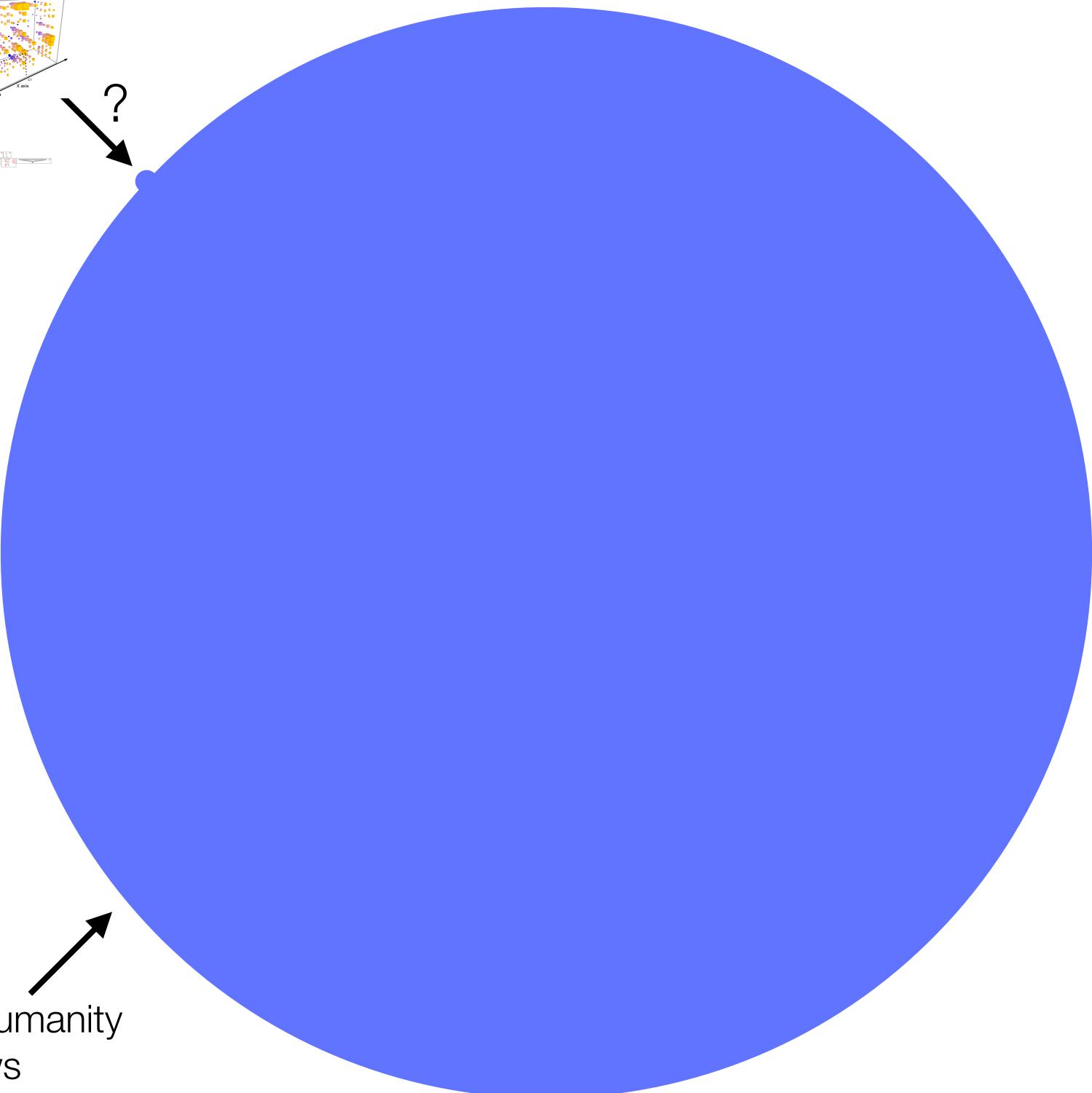
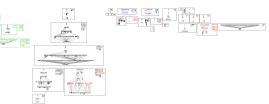
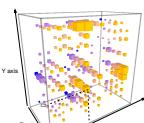
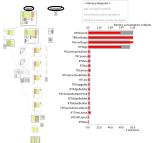


What the humanity  
knows



What the humanity  
knows





What the humanity  
knows

# How do we know we are improving the state of the Art?

---

Are the visualizations I just shown really useful?

How do programmers perceive our contributions?

Validation is complex and need to be done with care

We use

statistical models, psychological models

empirical methods

# Opportunities

---

When building a software, many decisions have to be taken

Important decisions are usually taken in group

Difficult decisions are often quickly taken, without much reflexion

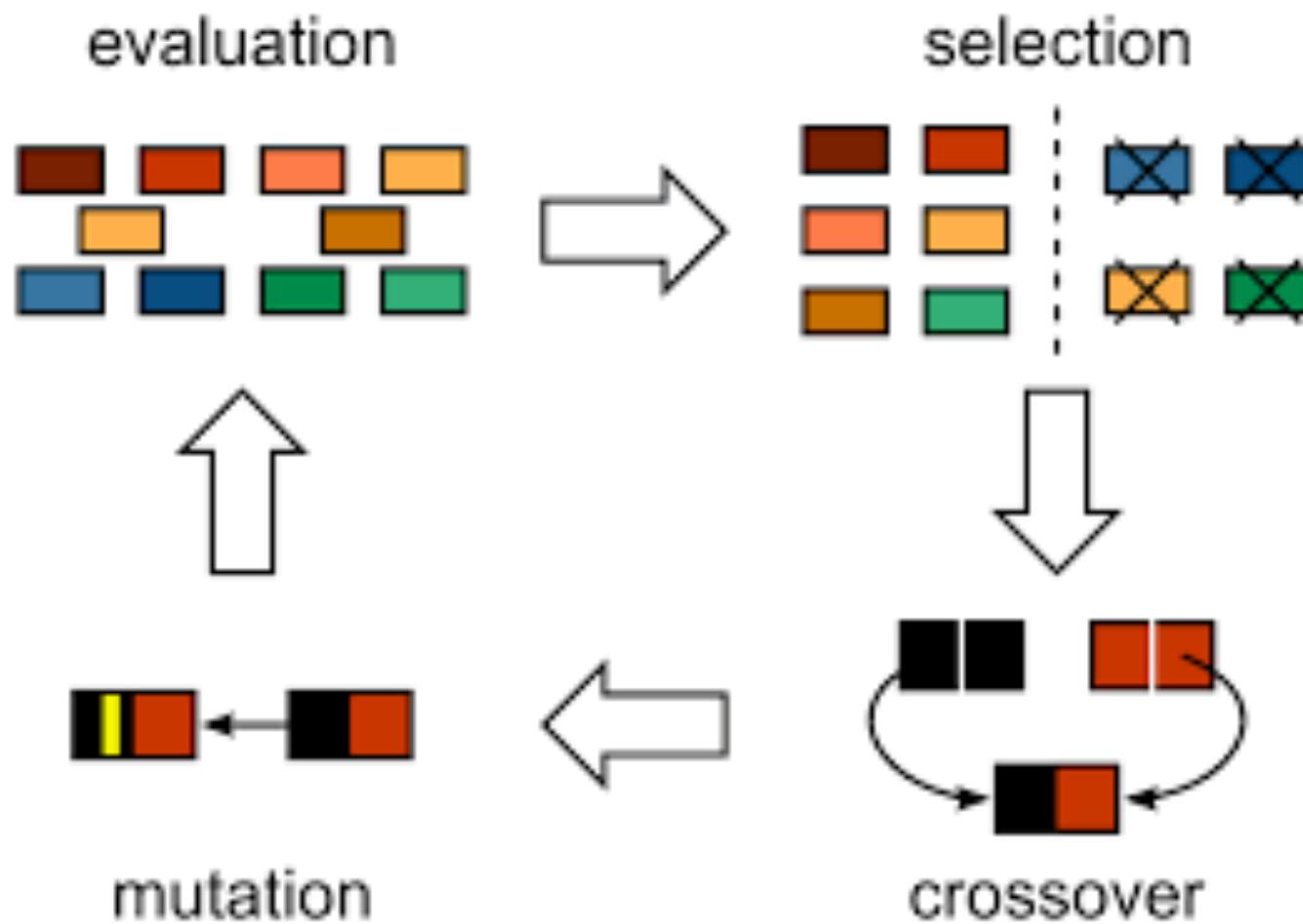
Why not letting the computer help us taking difficult decisions?

Artificial Intelligence algorithms are often used in data and images

... but rarely in software engineering

# Genetic Algorithm / Programming

---



# Research Idea 1: Applying genetic algorithm to software performance

---

A *benchmark* is a representative execution of a software system

Essential to measure performance evolution

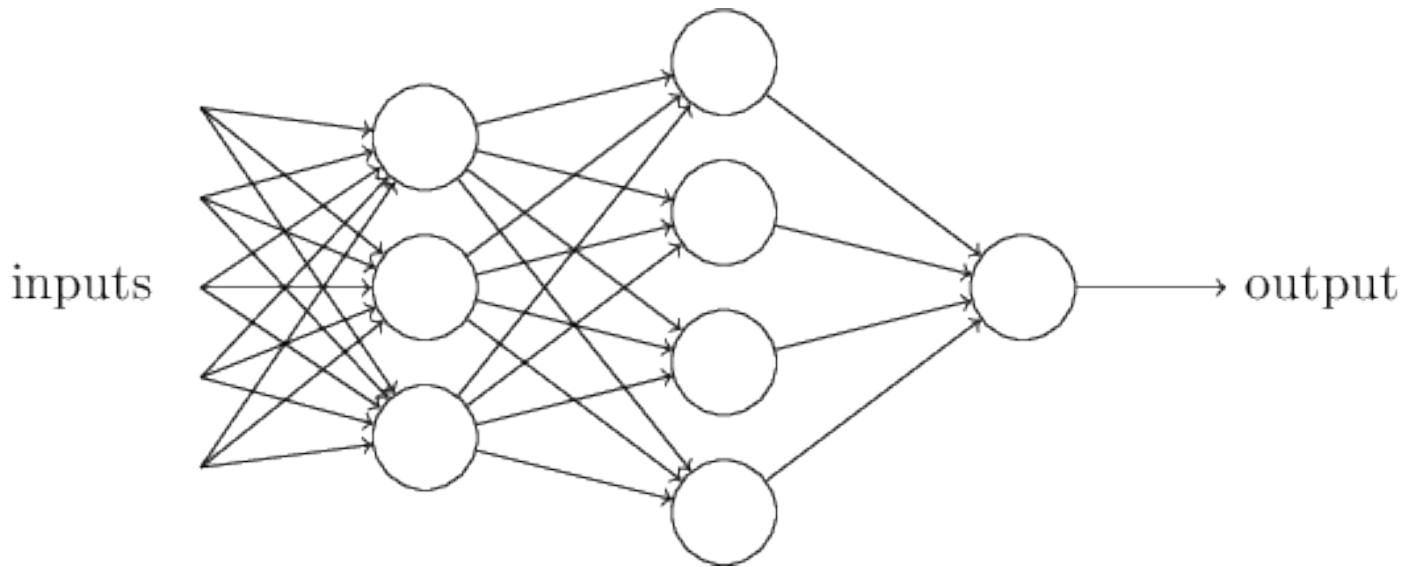
A benchmark requires a *workload*

However, defining such workload is not trivial

Genetic algorithm could be used to define software execution workload

# Neural Networks

---



A neural network is like a hashmap, but more flexible on its input

# Research Idea 2: Predicting software changes

---

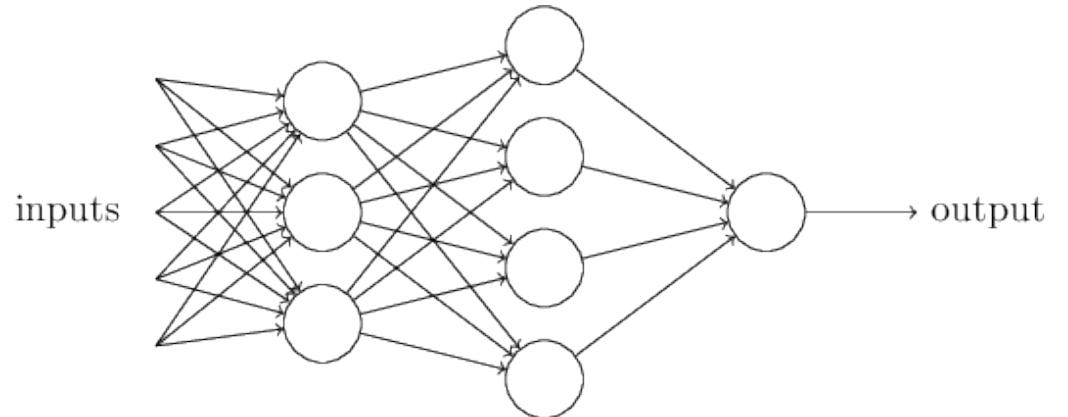
We are using Neural Networks to make prediction of future changes a software may have

Last month I changed my database and the UI of my application

If I change today my database, will I need to modify my UI tomorrow?

# Recommendation systems

---

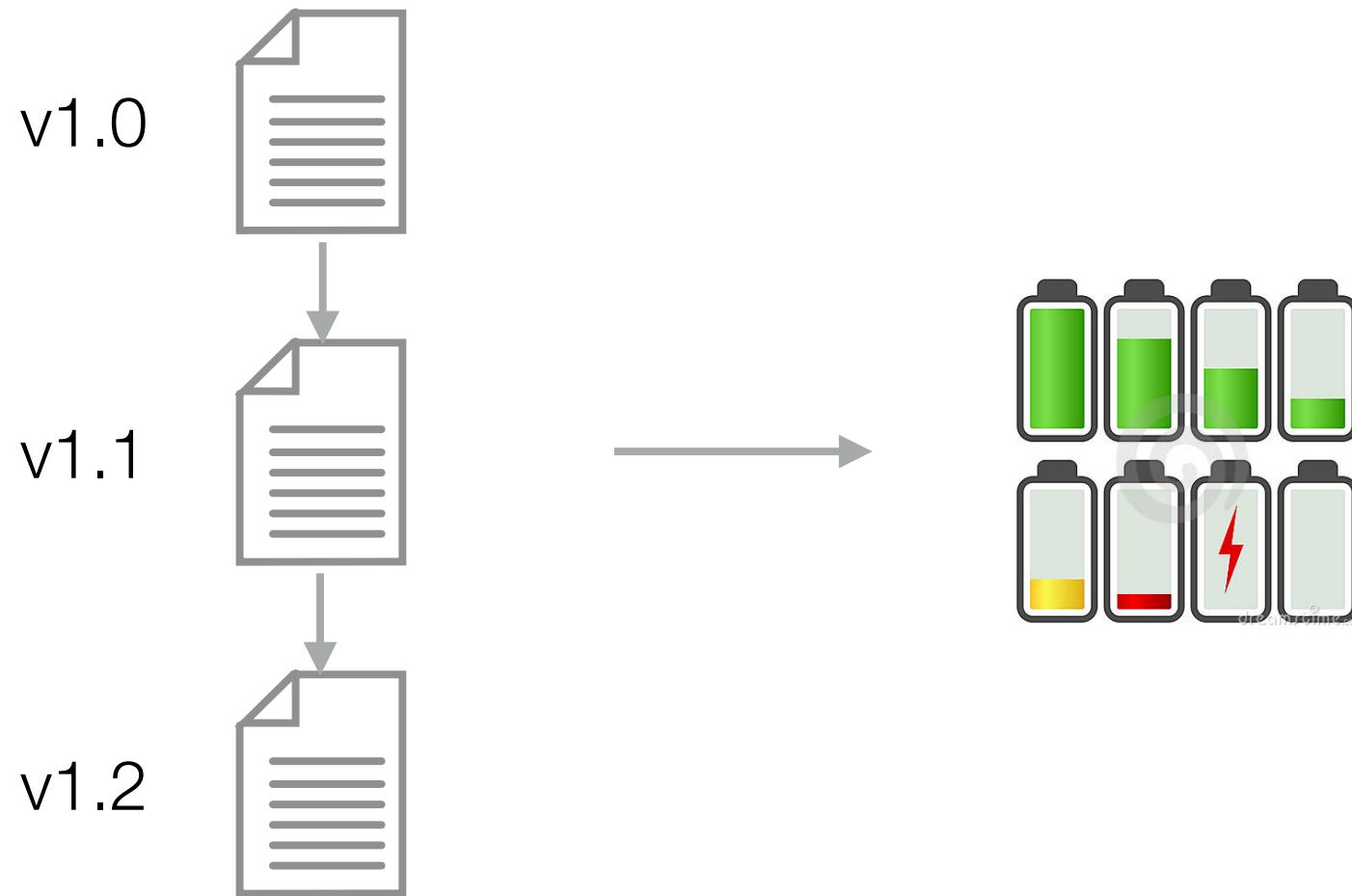


Using artificial intelligence and visualization techniques to help building software is a big topic

Can be applied to  
software testing,  
software performance,  
programming environments

# Battery consumption and source code evolution?

---



# What we have seen

---

Real software quality problems

A few innovative software engineering tools

Some research opportunities to intelligently build  
software

<http://bergel.eu>