Patterns of Organic Architecture

Madman's diary

By

Jarosław Pałka

... A word of warning ...

It is not may intention to make you feel offended,

this talk is

my view of the world,
my experience,
my life,
my twisted sens of humor

Respect it!!!

and remember

you can always leave the room

you can always talk to me

and whatever people say

I am just a human being

About me

work://chief_architect@lumesse

owner://symentis.pl

twitter://j_palka

blog://geekyprimitives.wordpress.com

scm:bitbucket://kcrimson

scm:github://kcrimson

What does society think I do?

What does my wife think I do?

What do I really do?


HUMAN BONE

~ 8 COMPANIES IN 16 YEARS

~ 26 PROJECTS

AND JUST

...ONE...

PROJECT BUILT FROM

...SCRATCH...

How do I feel about it?


I AM FEELING
LUCKY!

What I will not learn today
from this talk?

Which mixture of patterns,
xDD,
languages,
frameworks And paradigms
will lead me to success

But you will learn how to live with...

Monolithic, legacy code base,
which gets closer and closer with
each line to borders of human
capabilities,
and is about to colapse
into a black hole,
which is going to suck all living
developers within its reach

We are living in a

...Big Ball of Mud...

Autogenerated Stovepipe
Stovepipe Enterprise
Jumble
Stovepipe System
Cover Your Assets
Vendor Lock-In
Wolf Ticket
Architecture by Implication
Warm Bodies
Design by Committee
Swiss Army Knife
Reinvent the Wheel
The Grand Old Duke of York

What is common to all these cases?

Complexity

Do you need proof?

Let me give you the proof!!!

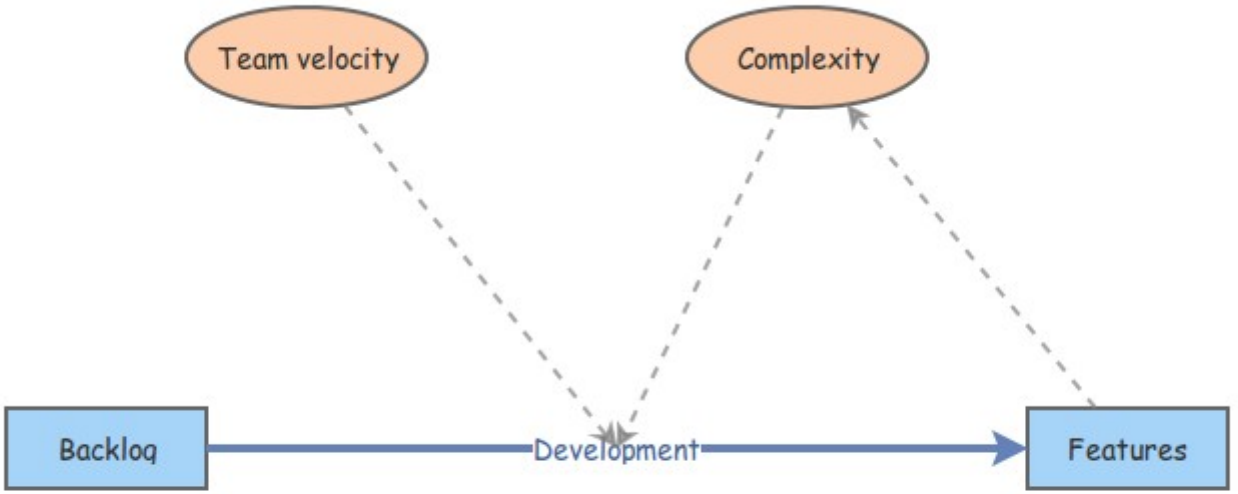
„I FUCKING LOVE SCIENCE”

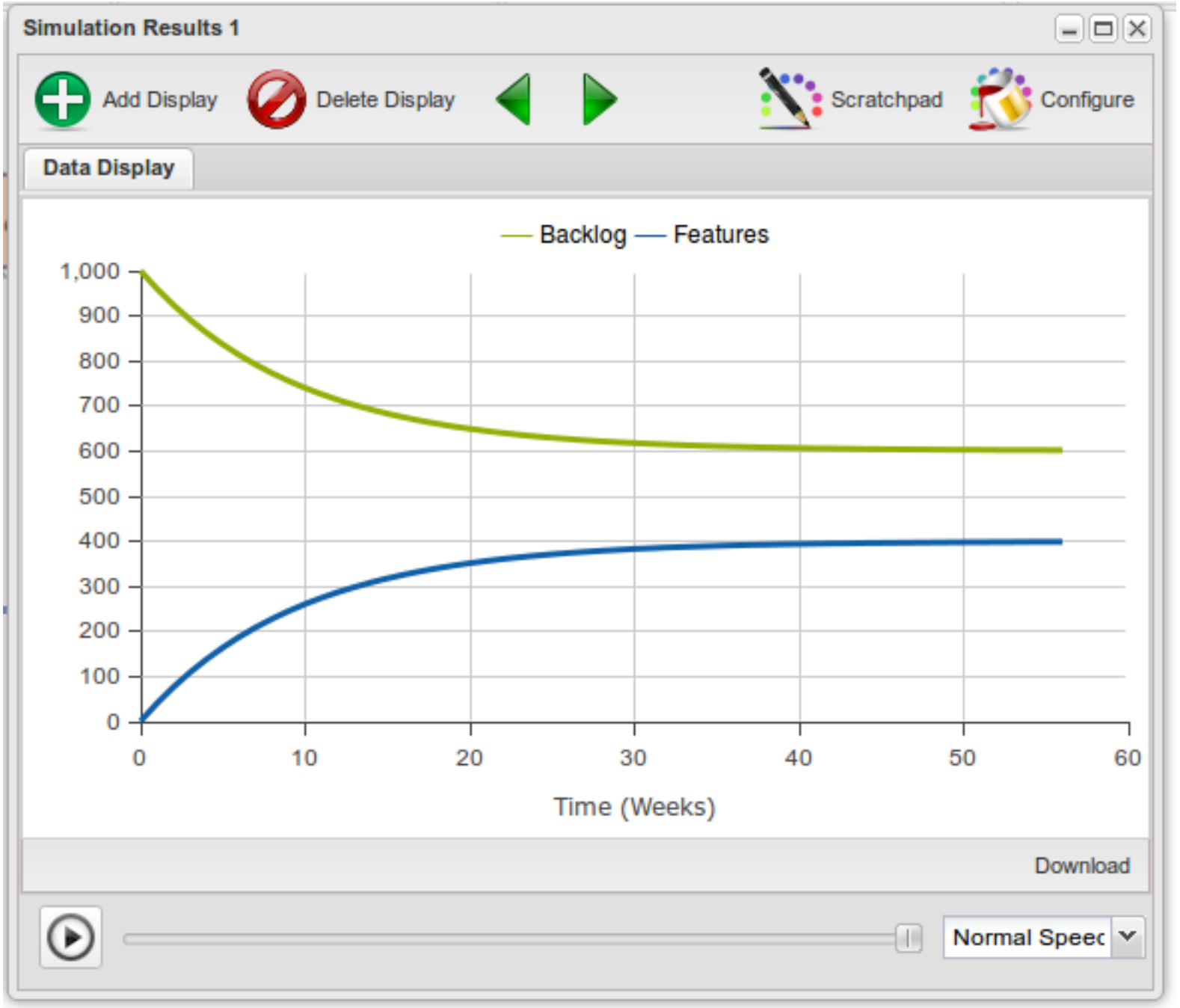
System's thinking

System dynamics

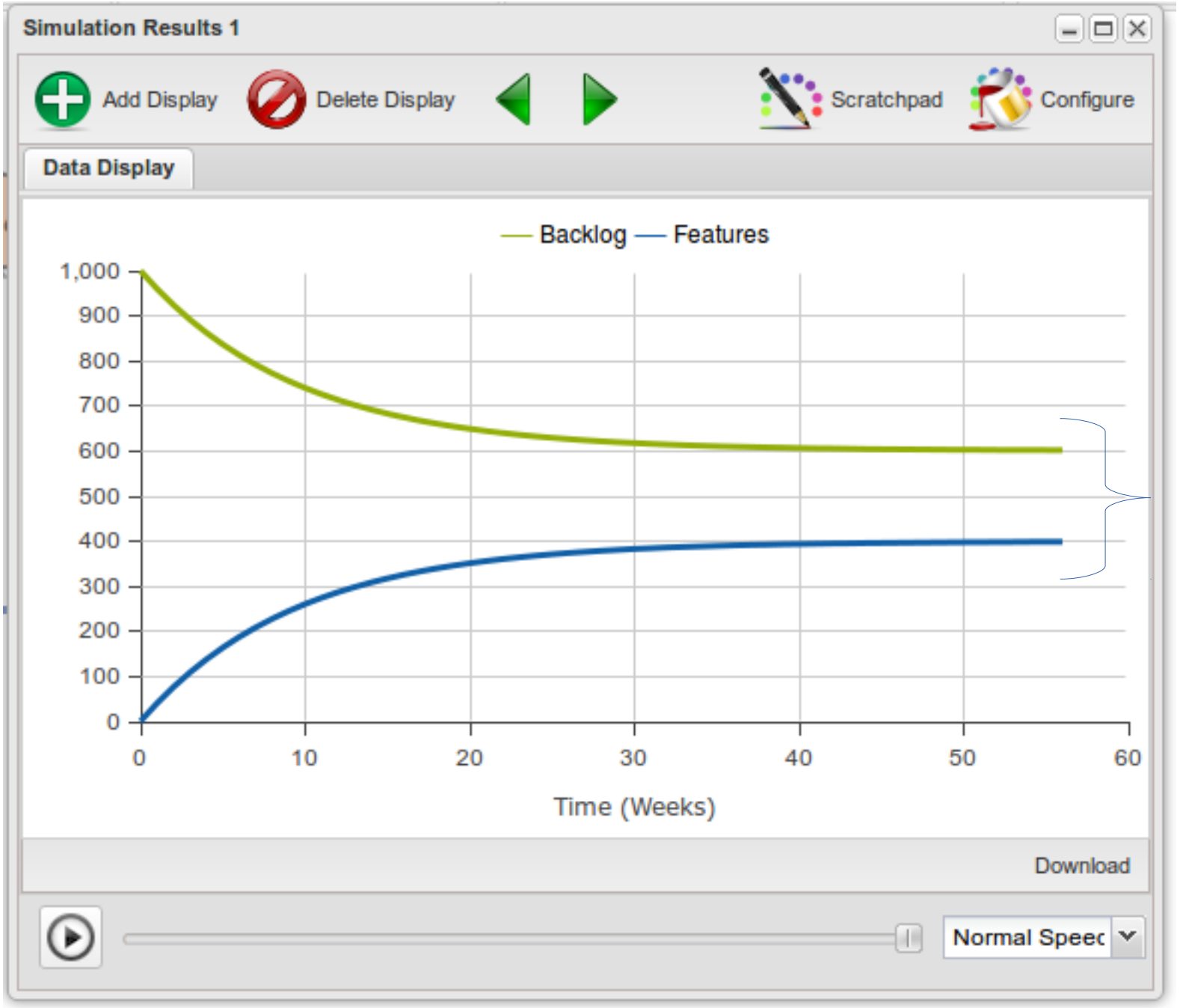
Complexity theory

Strange Attractor


Team velocity
Complexity
Backlog
Development
Features


Simulation Results 1
Add Display
Delete Display
Scratchpad
Configure
Data Display
Backlog
Features
1,000
900
800
700
600
500
400
300
200
100
0
0
10
20
30
40
50
60
Time (Weeks)
Download
Normal Speed


MIND THE GAP


Simulation Results 1
Add Display
Delete Display
Scratchpad
Configure
Data Display
Backlog
Features
1,000
900
800
700
600
500
400
300
200
100
0
0
10
20
30
40
50
60
Time (Weeks)
Download
Normal Speed

The Gap

How do organizations work around it?

Let's hire more students!!!

Let rewrite it from ...scratch...

(of course in newest, coolest technology we know shit about)

You may ask, why?

Time is not on our side

Few „extra features" everybody is waiting for?

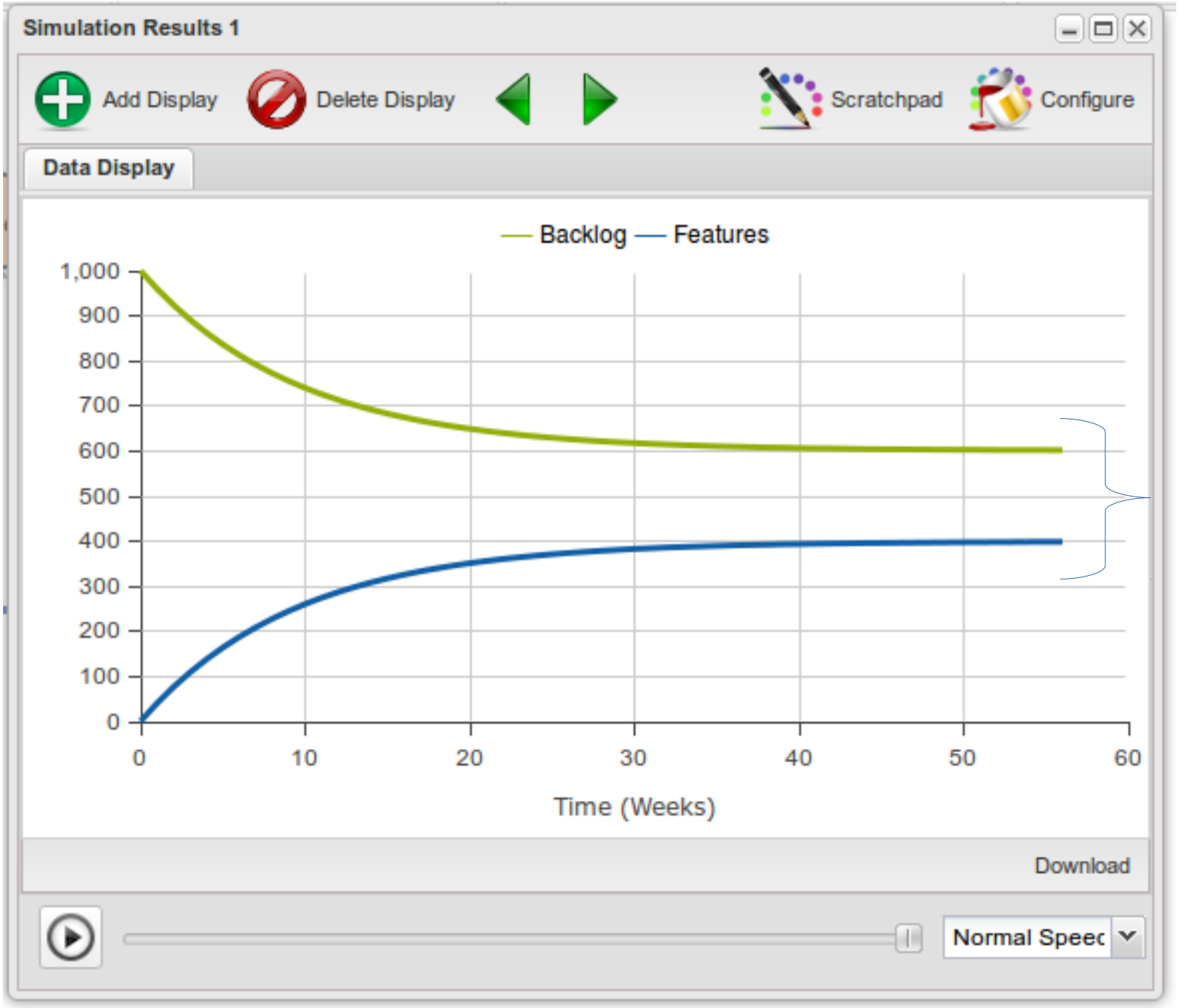
Too much faith in technology?

Too often these projects are seen as purely technical?

Ignorance?

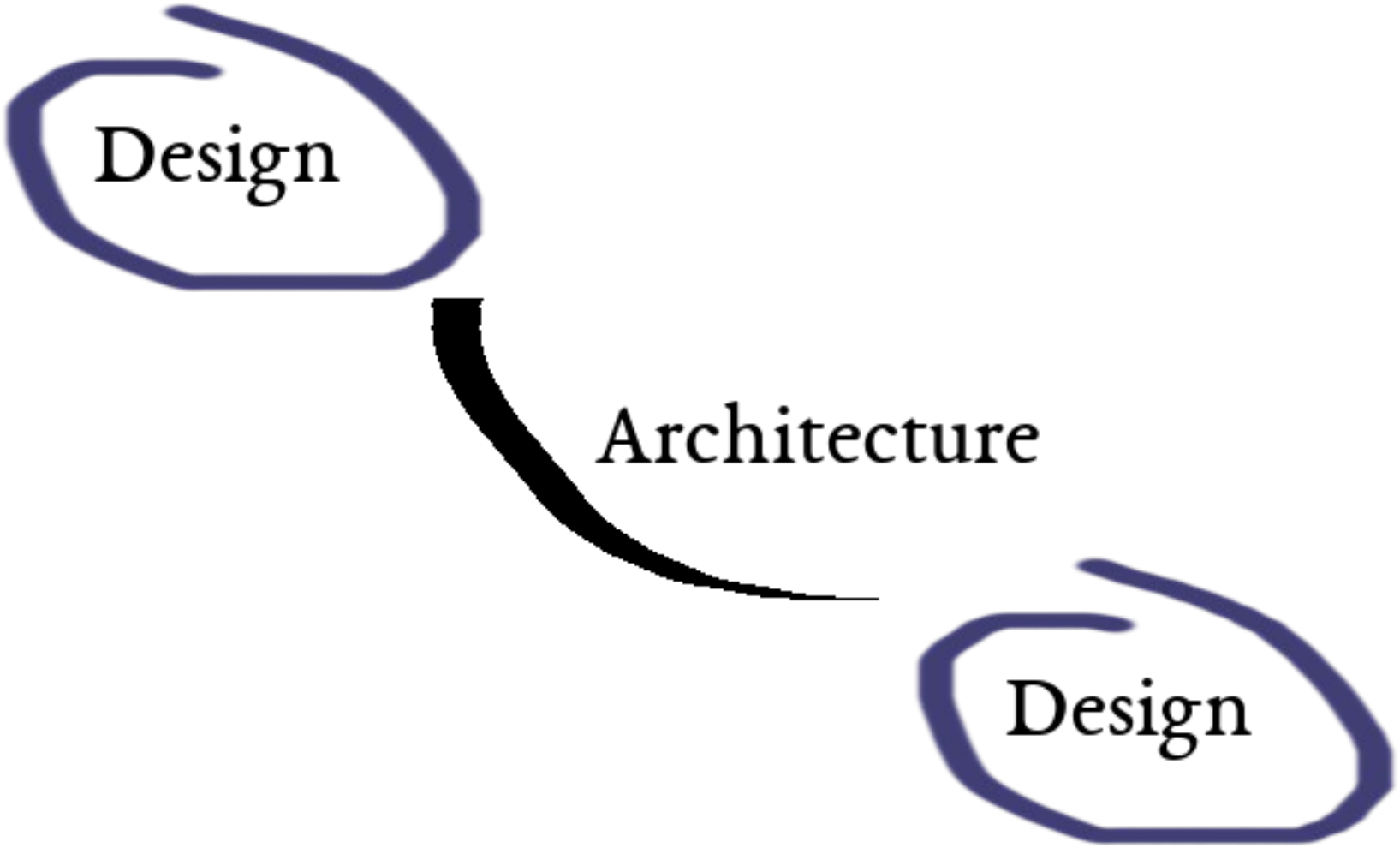
Arogance?

If it all doesn't work,
why not to try something
different?


Simulation Results 1
Add Display
Delete Display
Scratchpad
Configure
Data Display
Backlog
Features
1,000
900
800
700
600
500
400
300
200
100
0
0
10
20
30
40
50
60
Time (Weeks)
Download
Normal Speed

The Gap

Patterns of organic architecture


Design
Architecture
Design

Architecture is a process which goal is to transform your system from one design to another design

Architecture is a process
which goal is to
transform your system
from one design to
another design

Architecture is a process
which goal is to
transform your system
from one design to
another design

Understand the Gap

↓

Understand Context

↓

Identify Constraints

You can't control what you can't measure

Tom DeMarco,
Controlling Software Projects,

You can't reason about what you can't measure

@j_palka

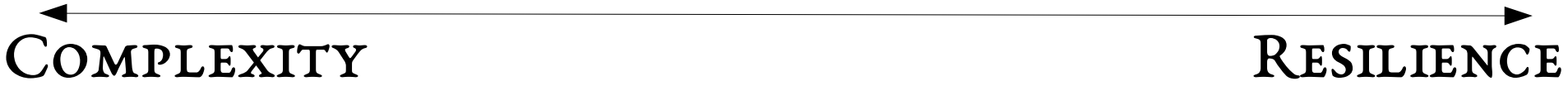
from a book which will never be written

You can't reason about what you can't measure

@j_palka

from a book which will never be written

So, how to measure your architecture?


Complexity
Resilience


Source code
the truth will
tell you


Listen to the system you must

SCM

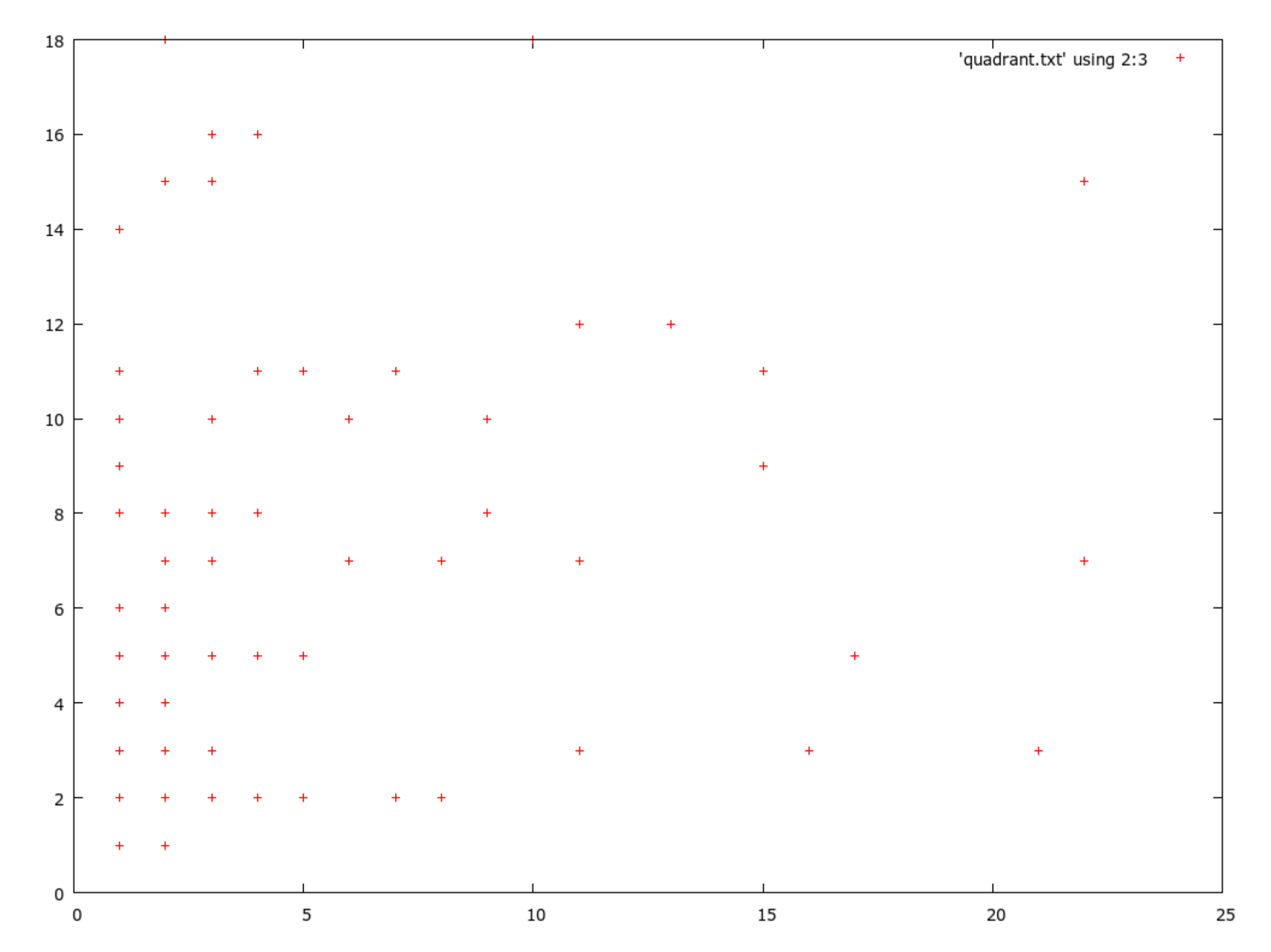
Bug tracker

Continous integration

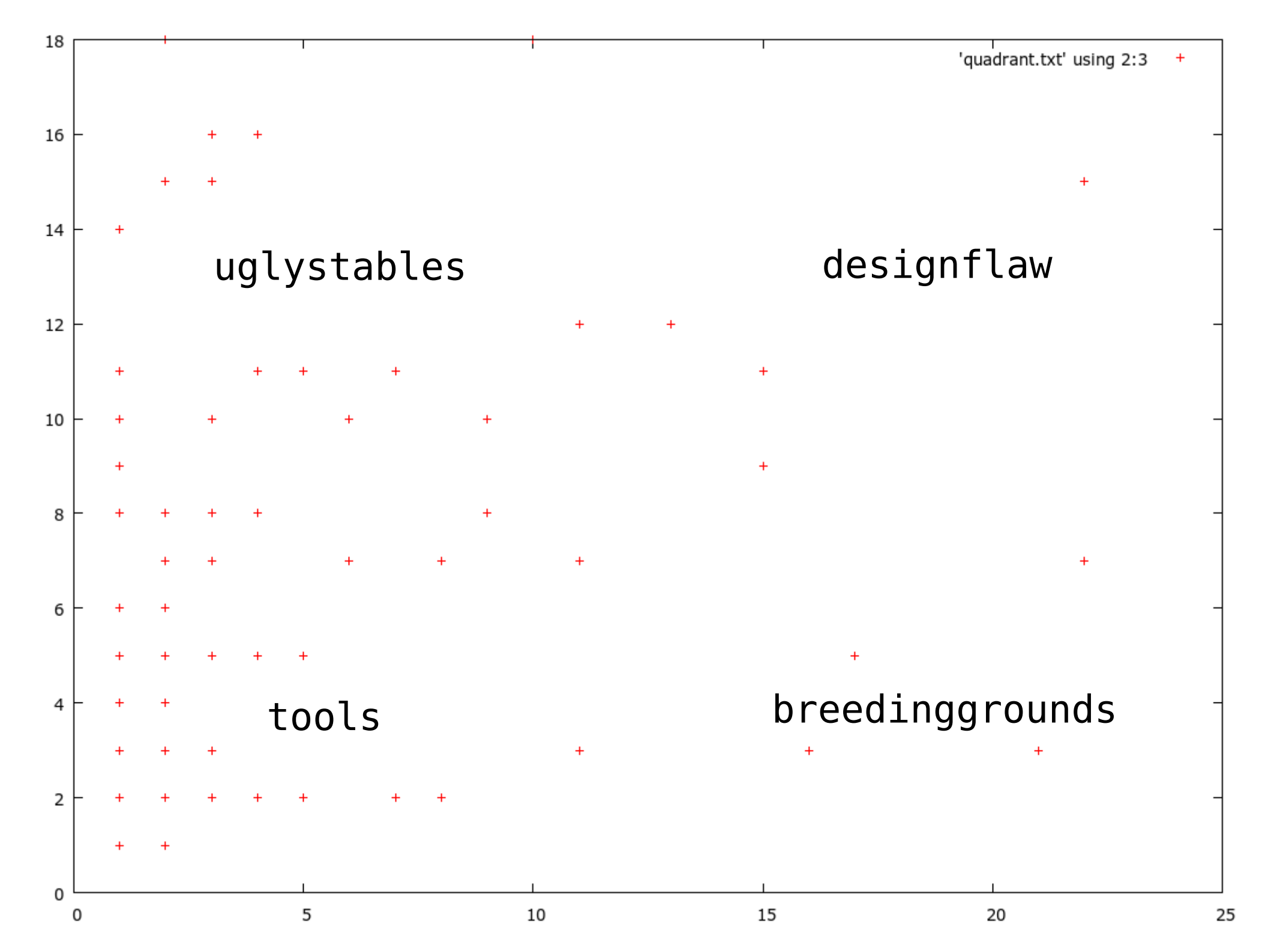
Static code analisys

Let's find stable parts of the system

```
# count complexity per each file
find . -iname jacoco.csv
| xargs tail -q -n +2
| awk -F , '{gsub("\.","/",$2);print ($1"/src/main/java/"$2"/"$3".java"),$10+
$11}'
| sort > coverage.txt

# count number of changes
echo  'changeset="{files}"' > files.style; echo 'file="\n{file}"' >> files.style

hg log --style files.style
| sort
| uniq -c
| awk '{print $2,$1}' > changes.txt

# merge changes
join coverage.txt changes.txt
```


'quadrant.txt' using 2:3
0
2
4
6
8
10
12
14
16
18
0
5
10
15
20
25

Michael Feathers Quadrant


'quadrant.txt' using 2:3
uglystables
designflaw
tools
breedinggrounds
0
2
4
6
8
10
12
14
16
18
0
5
10
15
20
25

Let's find fragile parts of the system

```
#fetch all jobs
jobs_rsp = requests.get(
"https://primitive.ci.cloudbees.com/job/roadrunner/api/python")
#all builds urls
build_urls = [x['url'] for x in eval(jobs_rsp.content)['builds']]

pairs = []
for build_url in build_urls:
    build = eval(requests.get("%sapi/python" % (build_url)).content)
    result = build['result']

    changeSetItems = build['changeSet']['items']
    if changeSetItems and not result == 'SUCCESS':
        affectedPaths = build['changeSet']['items'][0]['affectedPaths']

        for i in itertools.permutations(affectedPaths,2):
            pairs.append(i)

counter = collections.Counter(pairs).most_common(5)
for pair in counter:
    print pair
```

```
(('.../cli/CliConfigurationBuilderTest.java',
  '.../cli/RunTest.java'), 3)

(('.../cli/RunTest.java', '
   .../cli/CliConfigurationBuilderTest.java'), 3)

(('.../cli/CliConfigurationBuilderTest.java',
  '.../cli/BenchTest.java'), 3)

(('.../cli/BenchTest.java',
  '.../cli/RunTest.java'), 3)

(('.../cli/RunTest.java',
  '.../cli/BenchTest.java'), 3)
```

Package principles

aka

Are my classes in a right place?

```
(echo "<changes>" && hg log --template
"<changeset>
{files % '<file>{file}</file>\n'}
</changeset>\n"
&& echo "</changes>") > out.xml
```

+

Neo4j (jQAssistant)

```
neo4j-sh (?)$ MATCH (a)-[c:`changeset`]->(b) RETURN labels(a),c.times,labels(b) order by c.times desc limit
5;
+-----------------------------------------------------------------------------------------------------------+
| labels(a)                                          | c.times | labels(b)                                  |
+-----------------------------------------------------------------------------------------------------------+
| [".../listeners/SummaryScenarioListener.java"] | 13      | [".../listeners/LoggingScenarioListener.java"] |
| [".../listeners/LoggingScenarioListener.java"] | 13      | [".../listeners/SummaryScenarioListener.java"] |
| ["pom.xml"]                                        | 12      | ["roadrunner-core/pom.xml]                 |
| [".../cli/BenchTest.java"]                         | 12      | [".../cli/RunTest.java"]                   |
| [".../cli/RunTest.java"]                           | 12      | [".../cli/BenchTest.java"]                 |
+-----------------------------------------------------------------------------------------------------------+
```

Can you please show me these patterns of organic architecture?


Sow and Grow

„aka” refactoring

compulsive „refactoring” is evil

no user stories „refactor X”

before you start, think, is it worth it?

don't ask for permissions, it is better to ask for forgivness

give „technical debt” meaning

„VISUAL MANAGEMENT"

DEFINE LIMITED NUMBER OF METRICS

USE ONLY THESE METRICS WHICH SUPPORT YOUR GOALS

BECAUSE

„YOU GET WHAT YOU MEASURE"


Sow and harvest

„AKA” MODULARIZATION

MODULARIZE TO STABLE PARTS OF THE SYSTEM

OTHERWISE YOU WILL MAKE YOUR
SYSTEM EVEN MORE UNSTABLE

BUT BEFORE YOU START ...

LET'S BUILD SOME ...„FRAMEWORK" ...


YOU MUST BE JOKING

Architecture is a process which goal is to transform your system from one design to another design

... clearly define goals ...

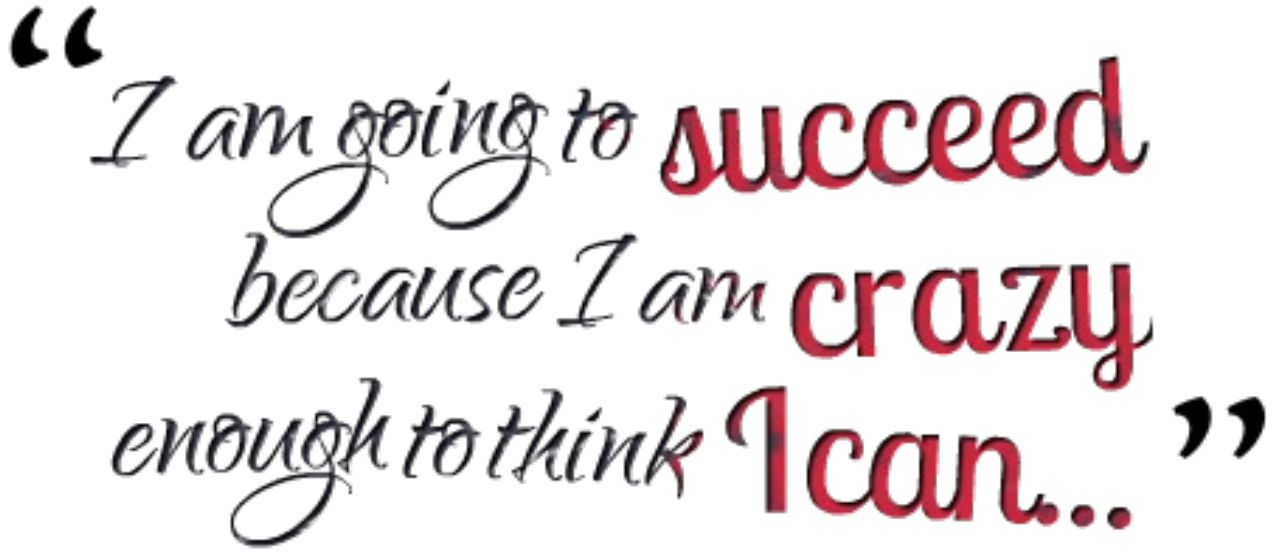
... strategy adjusted to needs and capabilties ...

... give yourself some design space ...

Don't get paralized by „big design (tm)"

You don't need to know answers to all questions

... because your goal is a moving target ...



Anonymous

Saturday - Nov 3, 2012(5:04 pm)

What if your architecture would look like this?

Batch processing separated from online

Modules communicate asynchronously

Users see system as one

And communicate with system synchronously

The only thing we share is a model of our system

mvn clean install < 60 sec

Every module has to inherit these principles

And can introduce new one which makes these system wide principles more specific

Composting

But sometimes,
despite our
hard work,
knowledge and
experience


Complexity

Do you know how your users use your application?

Do you know that your biggest customer is no longer using your system?

Do you know that some „killer feature” is not sooo „killer”?

How can I know it all?

/var/log/httpd/access.log

Instrument your code?

Aspects?

Byteman?

Bug tracker?

People from support?

Don't buy expensive tools

Invest in your creativity

You know your system better,
then some third party provider

But please don't comment out code

Use your SCM

... throw this shit out...

What's next?

Hierarchy

↓

Self-Organization

↓

Resilience

System's resilience is often sacrificed for purposes of short-term productivity and stability.

Productivity and stability
are the
usual excuses for turning
creative human beings into
mechanical adjuncts
to production processes.

Or for establishing bureaucracies and theories of knowledge that treat people as if they were only numbers.

Donella Meadows, thinking in systems a primer

Thank you!!!