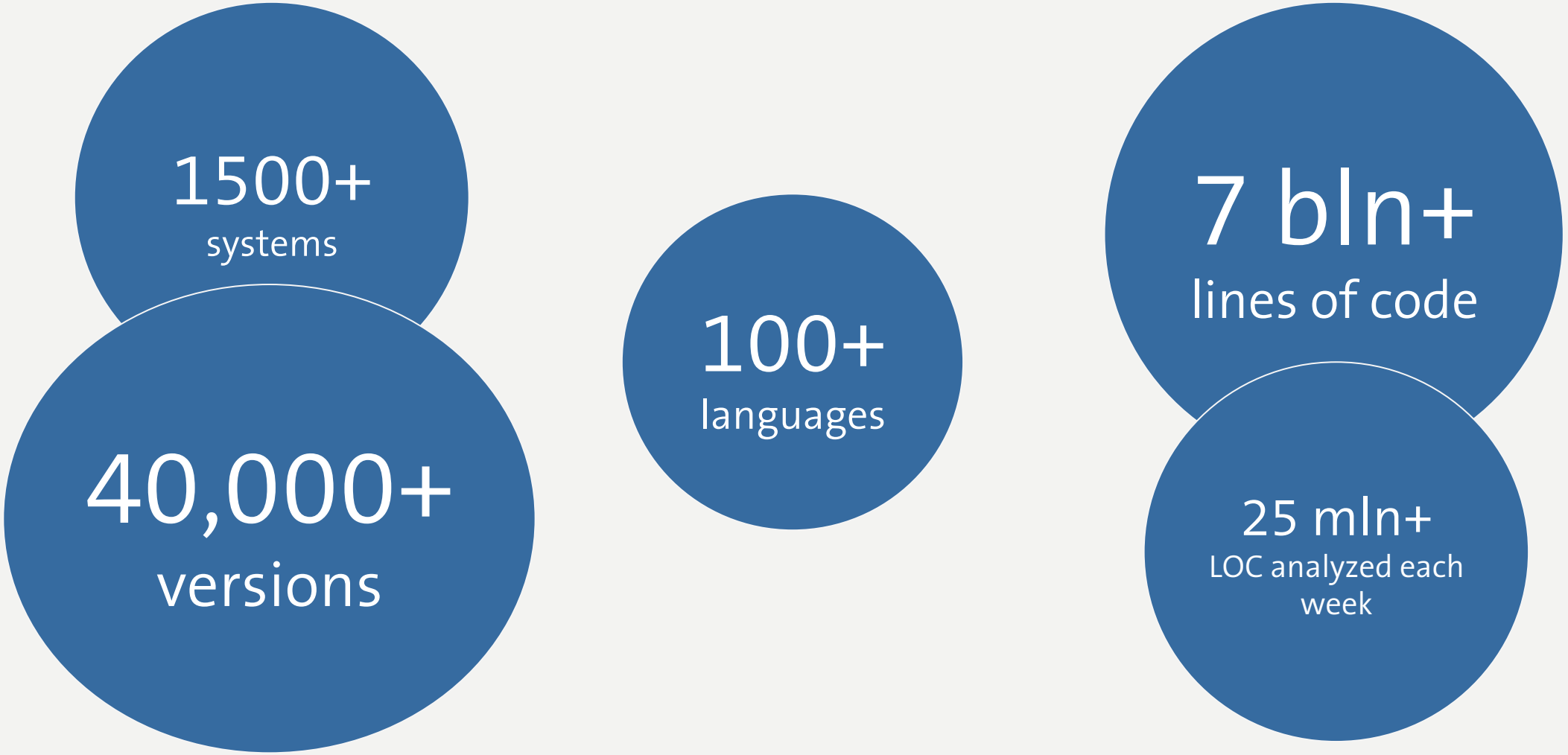
Software Improvement Group

System Architecture Metrics at SIG: Lessons Learned

Zeljko Obrenovic, http://obren.info

February 2, 2016

Main-tainability TÜViT 2015 Trusted Product
Voluntary Validation



Contents

100+ languages (200 dialects)

abap, abapsmartforms, abl, accell, acl, actionscript, actionscript3, **ada**, adabasnatural, adfxml, ads, angularjstemplate, applicationmaster, aps, aquima, asp, aspx, batch, bea, bixml, biztalk, bpel, brail, bsp, **c**, cache, cacheobjectscript, ccl, cgdc, cl, clearbasic, clojure, **cobol**, coffeescript, coldfusion, common, configuration, coolgen, coolgenc, coolgencobol, cordysbpm, **cpp**, csharp, csharppreprocessed, csp, css, cucumber, datastage, datastageetl, datastageworkflow, db2, dcl, delphi, delphiforms, docker, drools, easytrieve, egl, ejs, embeddedsql, erb, esql, filetab, finacle, flow, **fortran**, freemarker, gensym, groovy, gsp, gupta, hql, html, ibmbpm, ibmbpmbpd, ibmbpmprocess, ideal, idms, importexpander, informix, informix4gl, informixsql, ingres, intershoppipeline, jade, jasperreports, **java**, javafx, **javascript**, jbc, jcl, jcs, jsf, json, jsp, less, lodestar, logicnets, lotusscript, lua, magic, magik, magnum, matlab, **mendix**, messagebuilder, models, mustache, mysql, naviscript, nodejs, nonstopsql, normalizedsystemsjava, **objectivec**, odi, opa, opc, openroad, oracle, pascal, pascalprecompiled, performance, perl, php, pli, plsql, plsqlforms, plsqlreports, postgresql, powerbuilder, powershell, progress, progressstrict, pronto, prt, puppet, python, razor, rexx, rpg, ruby, sas, sass, **scala**, scl, scr, script, shared, shellscript, siebeldeclarative, siebeljs, siebelscripted, siebelworkflow, smalltalk, sourcegraph, sql, sqlite, sqlj, sqr, ssis, starlimssql, streamserve, swift, synon, t4, tacl, tal, tandem, tapestry, tibco, tibcobe, tibcobejava, tibcobestatemachine, tripleforms, tsql, turtle, typescript, uil, uniface, unknown, until, vag, vagrecord, vb, vbnet, velocity, visualobjects, visualrpg, webfocus, webmethods, wsdl, wtx, xaml, xml, xpdl, xpp, xquery, xsd, xslt, yaml

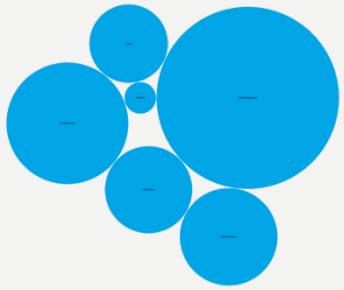
Dependencies among modules

31 modules, 347 connections, 79 places with cyclic dependencies

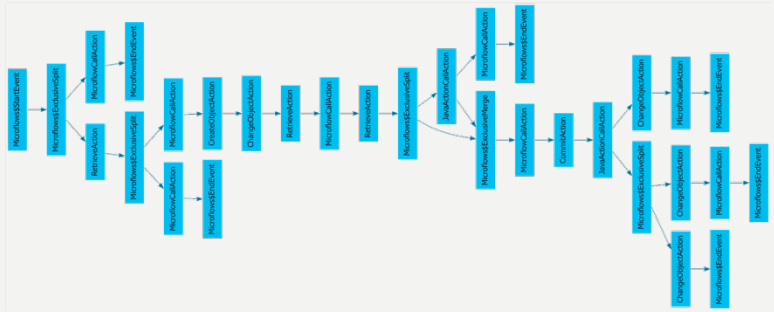
Basic Types of Metrics

SIG is a “boring” company

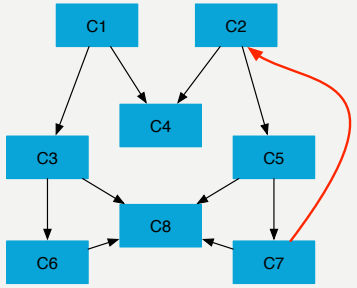
- **Size** – e.g. lines of code



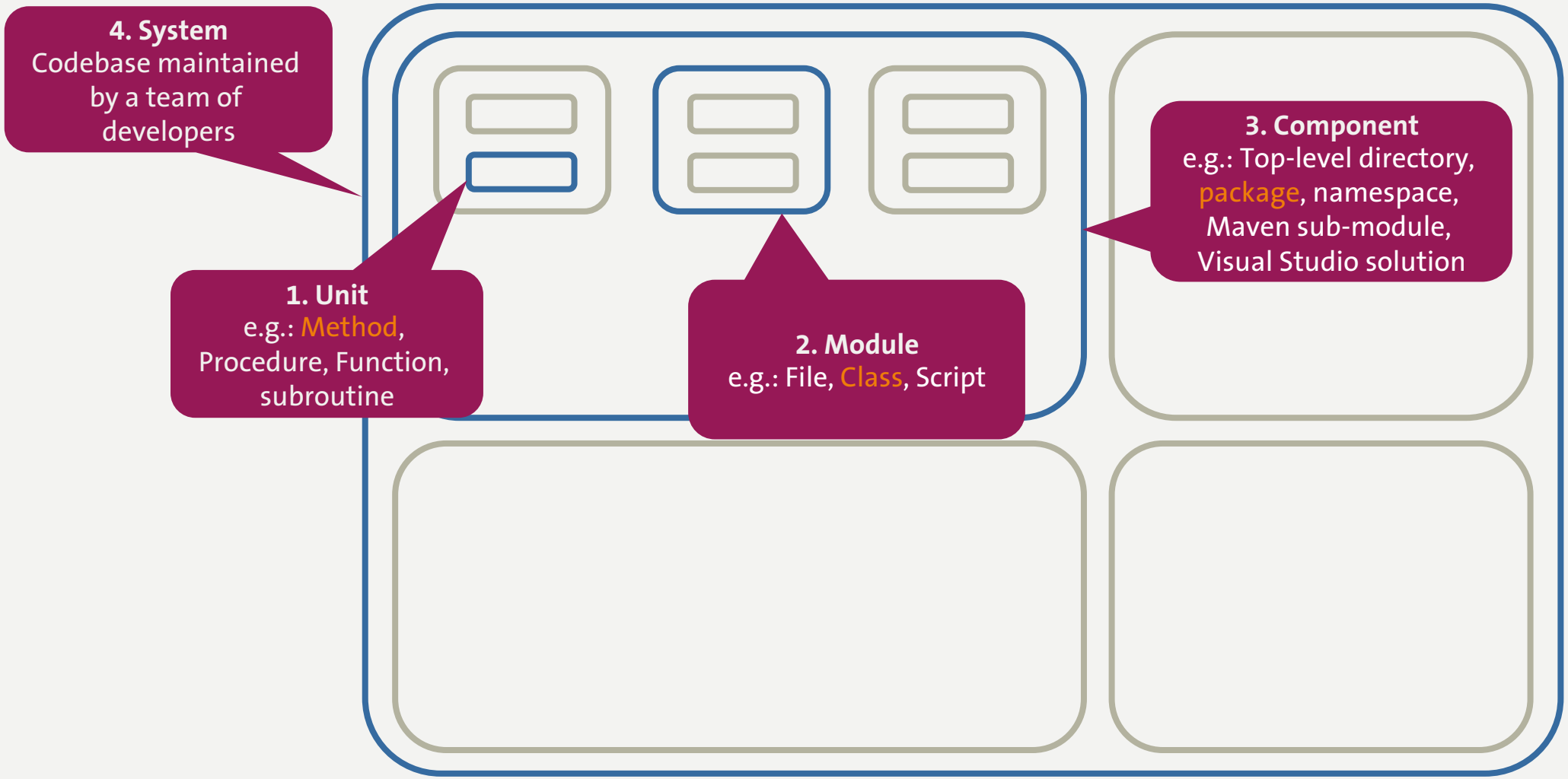
- **Complexity** – e.g. control flow



- **Coupling** – degree of interdependence



Mapping technologies to basic metrics

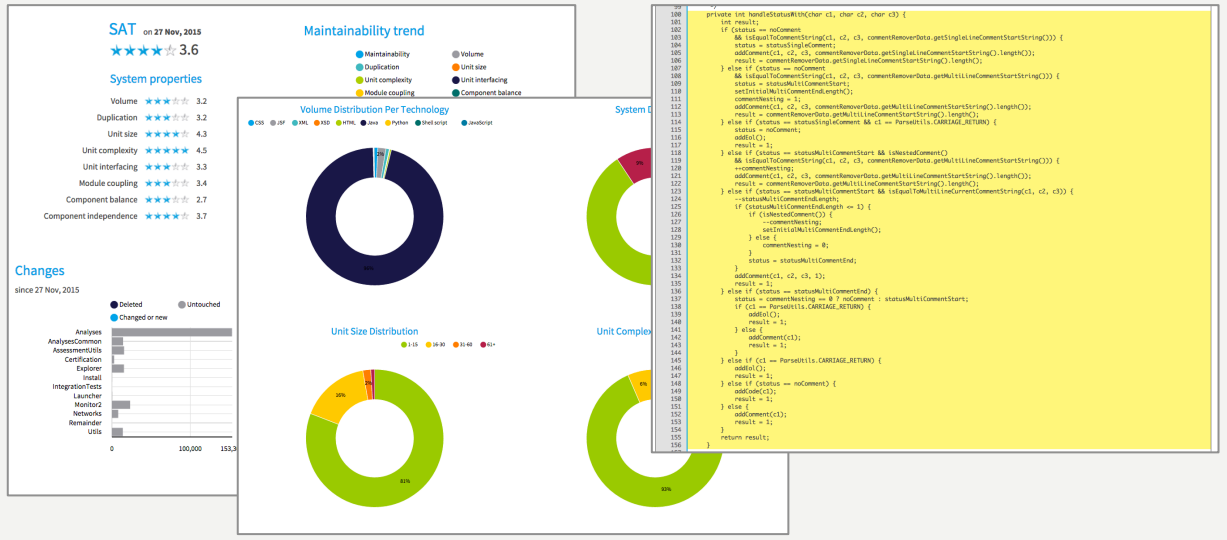


Contents

Getting what you measure

"Metric is a good servant but a bad master"

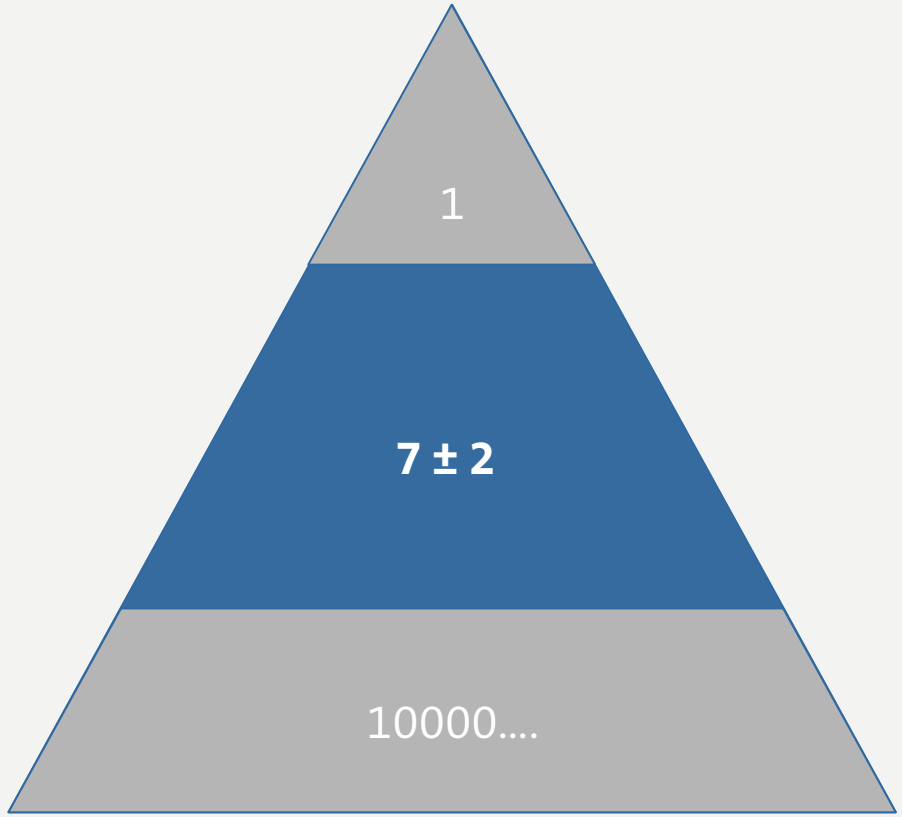
- Making alterations just to improve the value of a metric
  - Difficult to avoid using metrics as a goal rather than as a feedback mechanism
- Recognized when changes made to the software are purely cosmetic.
- Examples: order of getters and setter, mechanical splitting of methods (e.g. step 1, step 2…)
- In the end you need to be able to zoom in the code
  - Need for tools that enable rapid metrics zoom in / zoom out to the source code



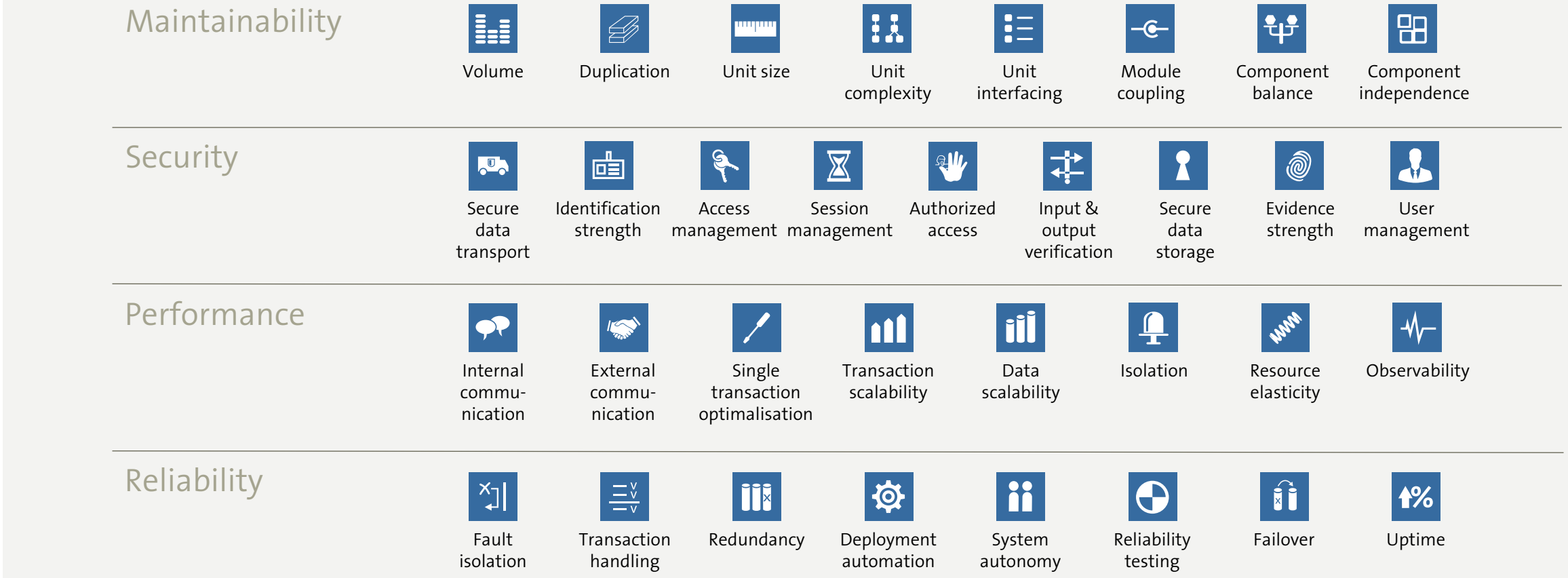
Getting what you measure

One-track metric or a metric galore

- Focusing on only a single metric:
  - E.g. duplication vs. coupling
- “Focusing” on too many metrics:
  - E.g. more metrics than lines of code
- 7 ± 2 (complementary) metrics
  - Applying usability principles when designing metrics



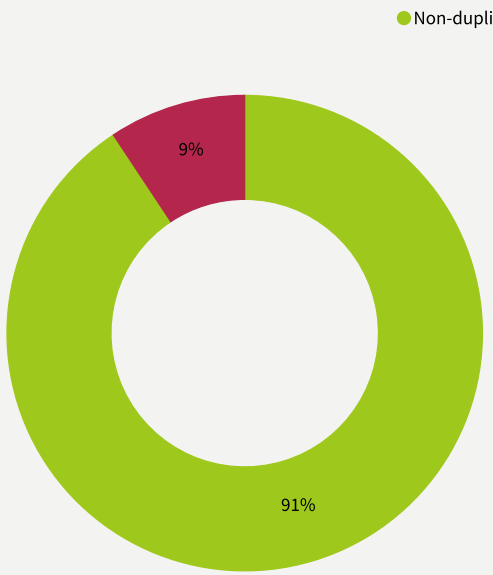
SIG Metrics Models



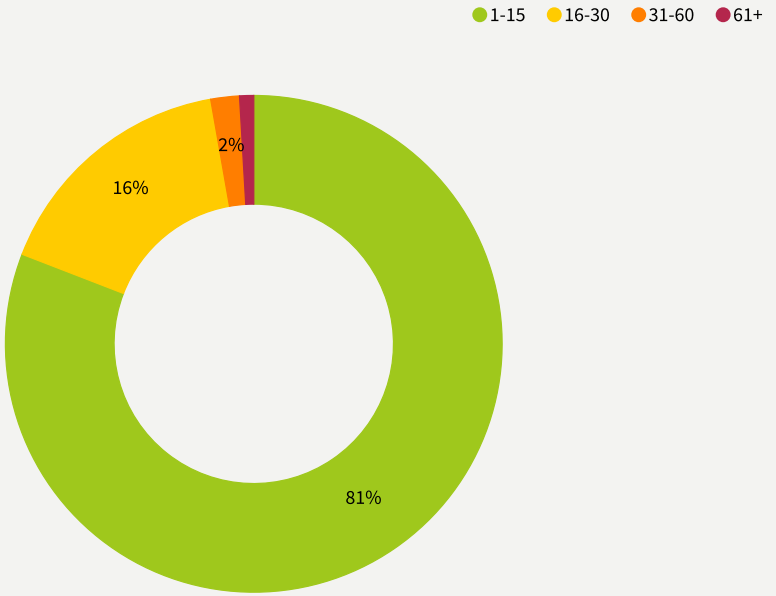
Getting what you measure

In the land of details aggregation rules

- 2,463 duplicates vs. 9% duplication



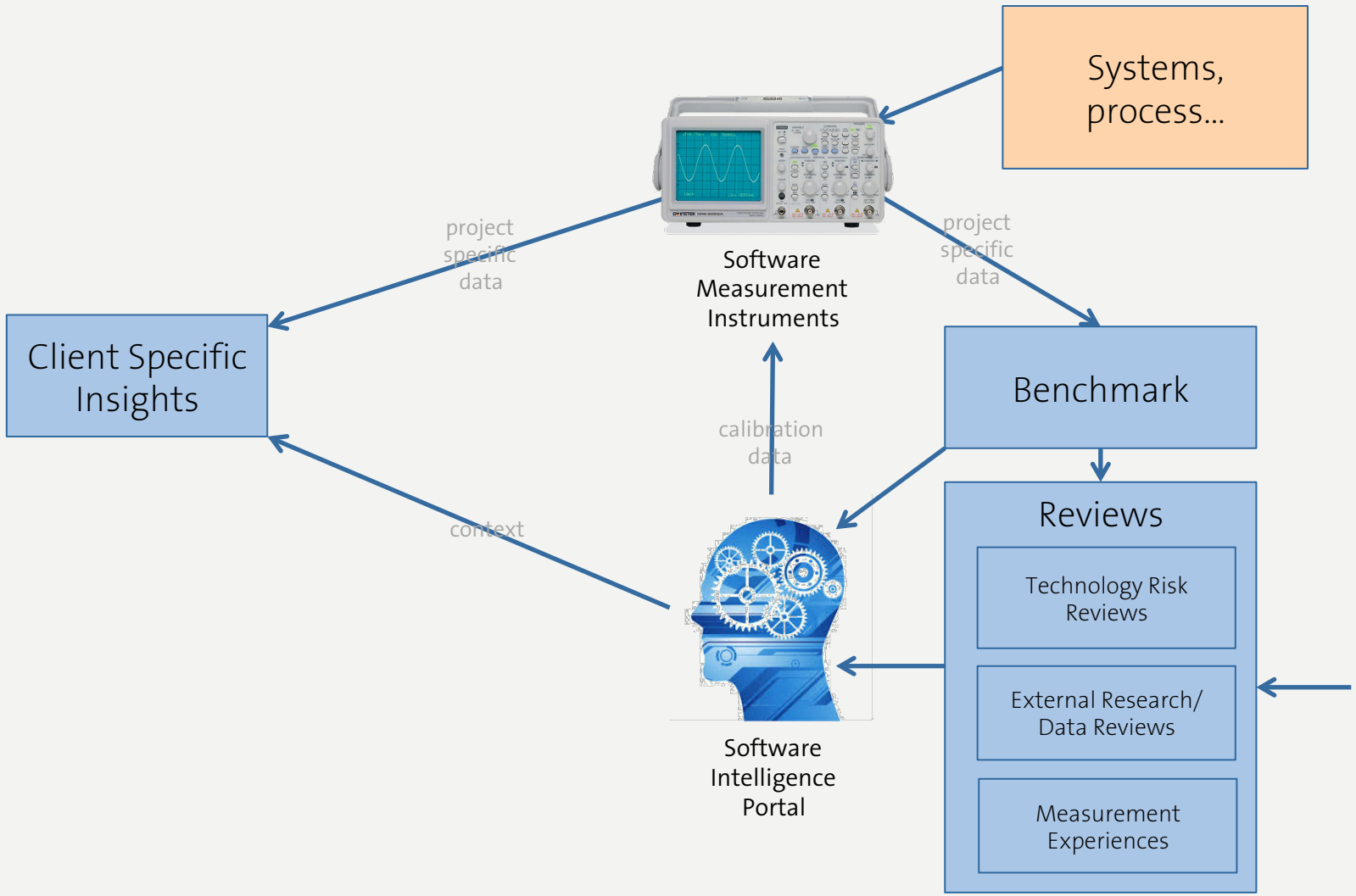
- 28,882 methods,
  1,326 are longer than 15 lines of code vs. risk profiles



Getting what you measure

Giving meaning to numbers

- Comparison with other systems – benchmark
- Comparison in time (trend analysis)
- Research


Systems, process...
project specific data
project specific data
Software Measurement Instruments
Client Specific Insights
Benchmark
calibration data
context
Reviews
Technology Risk Reviews
External Research/ Data Reviews
Measurement Experiences
Software Intelligence Portal

Getting what you measure

"Better wrong than vague"

- About concrete numbers with transparent model
  and connected to the concrete examples source code you can have concrete discussion

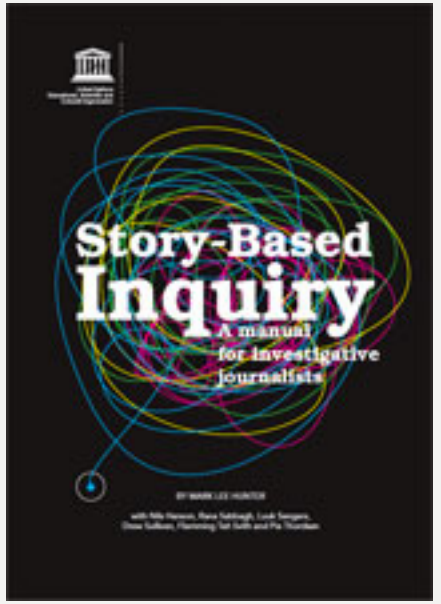
- Sometimes the biggest value of metrics is not the measurement
  but making such discussions possible

Contents

Story

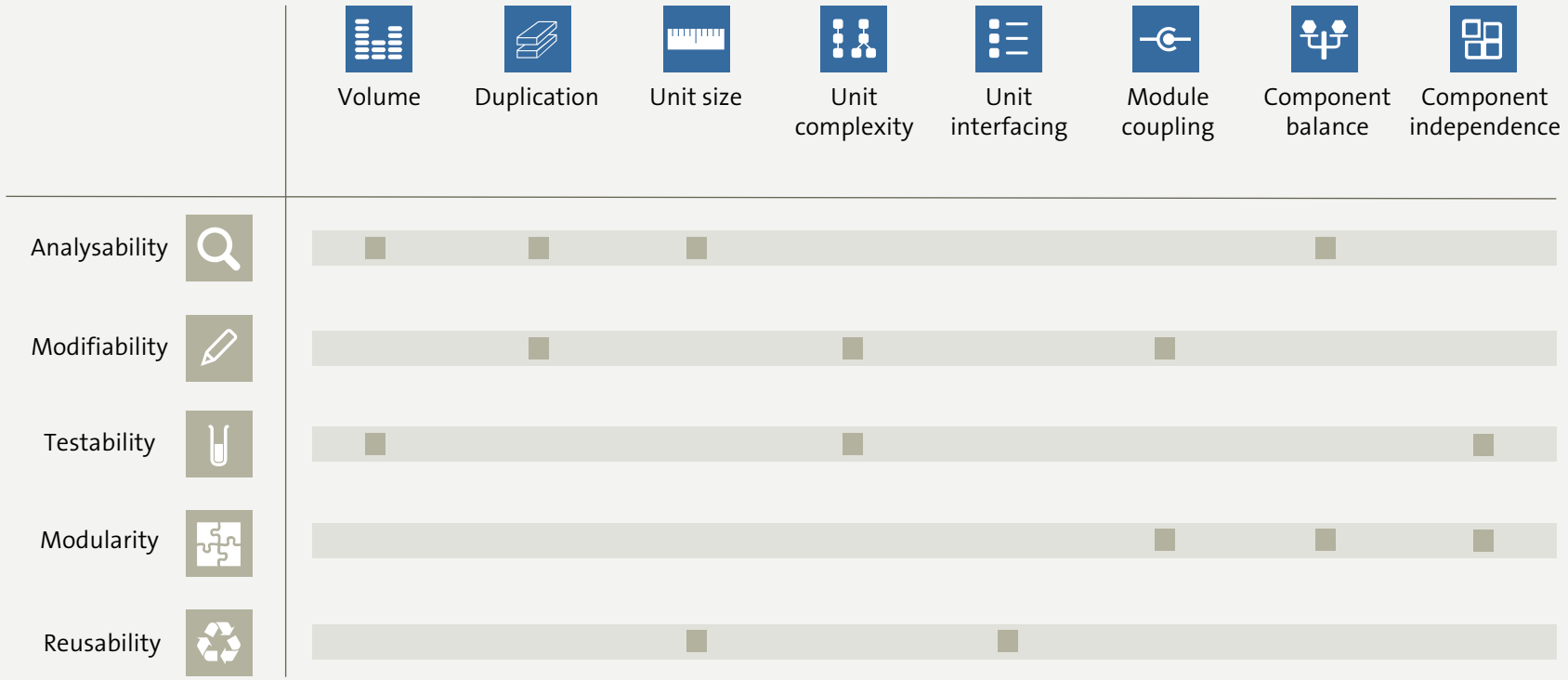
Unexpected lessons from investigative journalism

- *“The facts [metrics] do not tell the story. The story tells the facts [metrics].”*
- We discover a subject.
- We create a hypothesis to verify.
- We seek open source data to verify the hypothesis.
- We seek human sources.
- As we collect the data, we organise it –
  so that it is easier to examine, compose into a story, and check.
- We put the data in a narrative order and compose the story.
- We do quality control to make sure the story is right.
- We publish the story, promote and defend it.

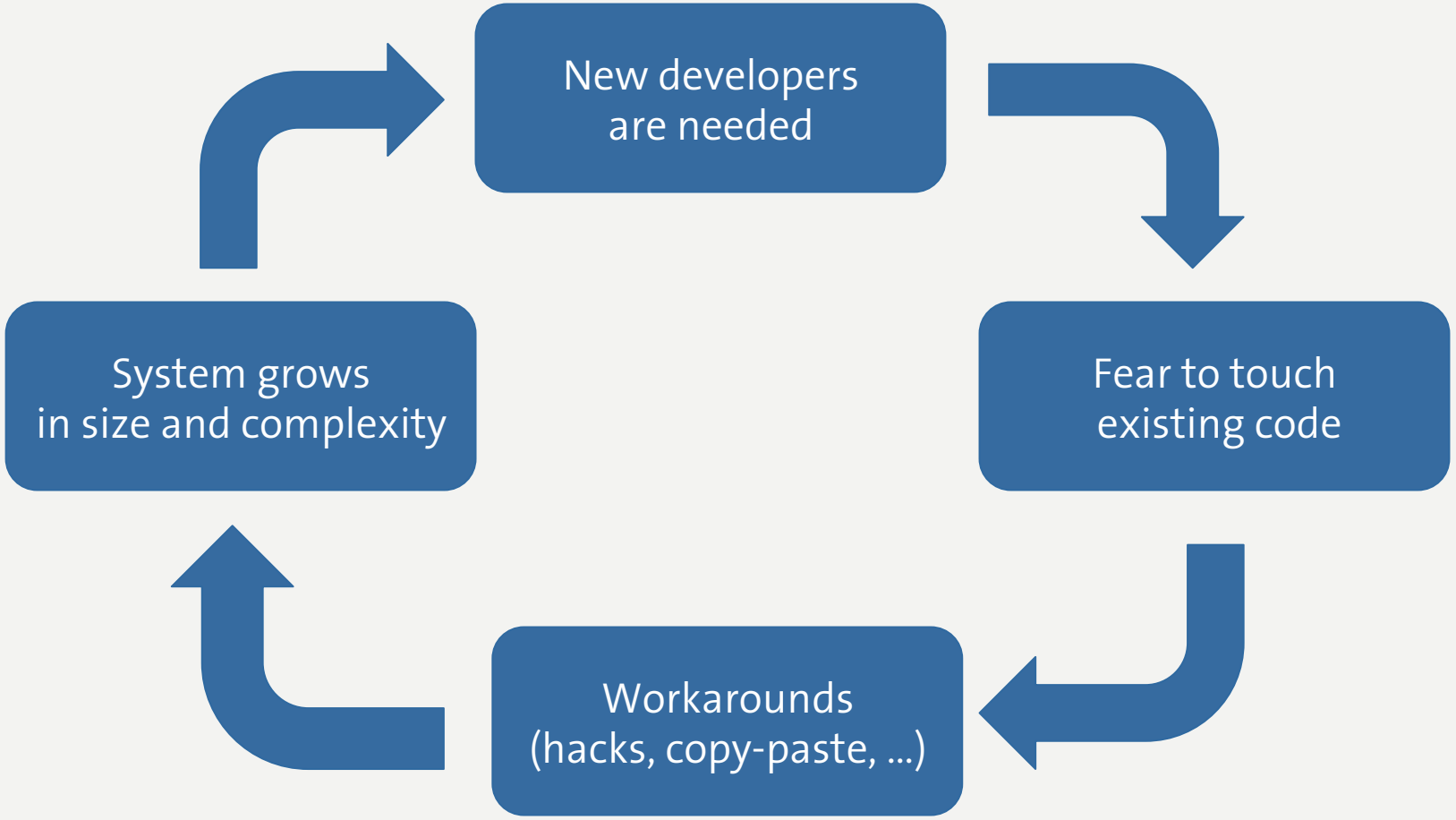


SIG Quality Model Maintainability

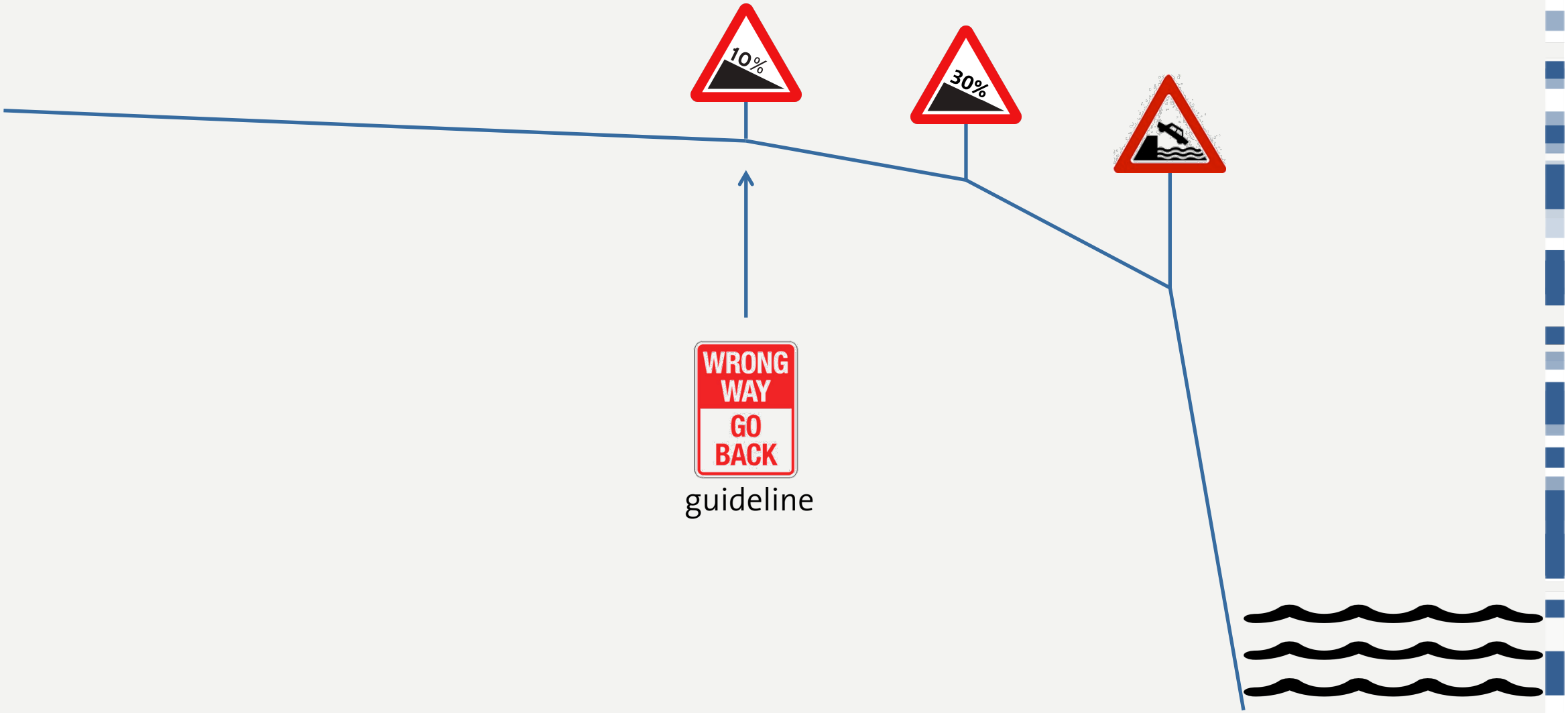
Metrics and their story, takes time to build narrative



The Cycle of Unsustainable Software Development



From Measuring to Guidelines – Identifying Tipping Points



Contents

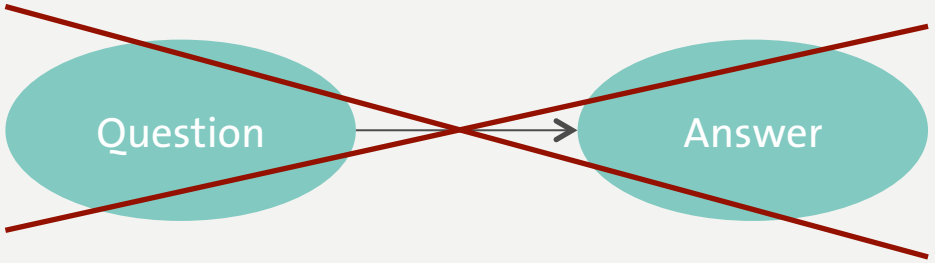
Analysis with metrics is a creative process

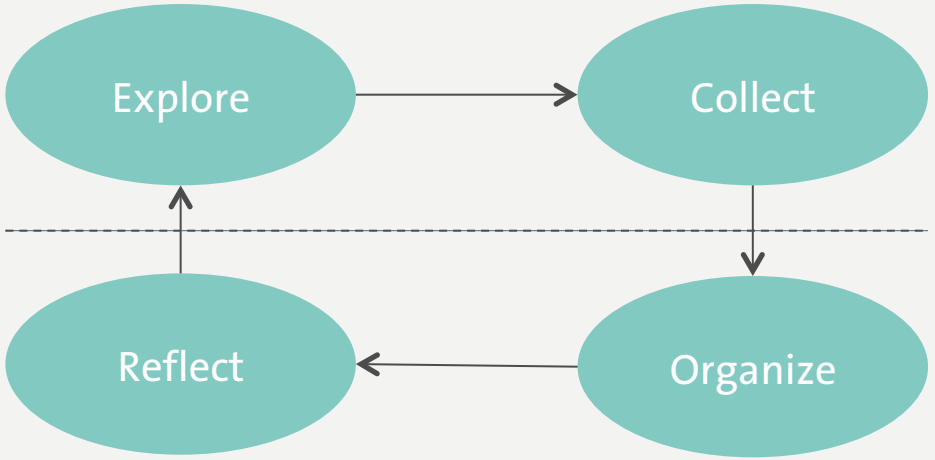
Principles of creativity support tools

- **Exploration**
  - Explore metrics, use metrics to guide exploration
  - Speed and ease, safe exploration space
- **Many paths and many styles**
  - GUI, scripts...
- **Open Interchange**
  - E.g. database, XML, CSV, Excel, PowerPoint...

→ Enable innovative usage of tools

→ e.g. quality workshops





Metrics cannot replace exploration of source code

Pattern of metrics driven explorations

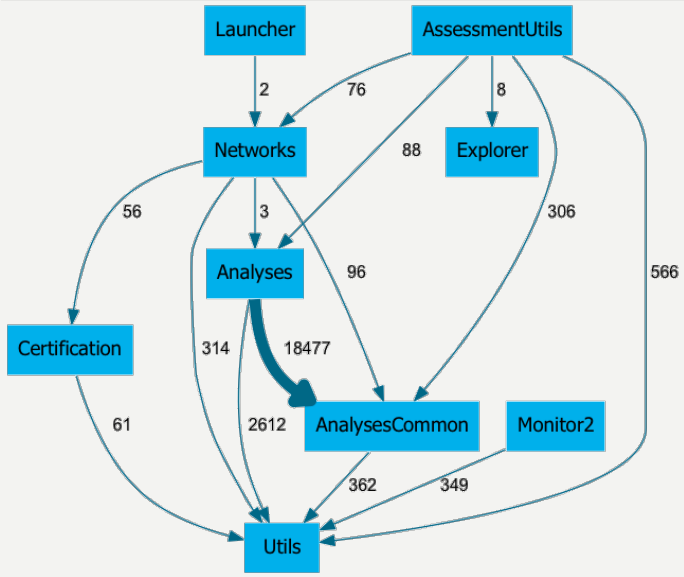
- Biggest / most complex constructs first
- Elephant in the room, fast forward through code
  - Not missing huge atypical concepts
- Simple query + false positives elimination
  - Simple search with more false positives
    + quick exploration and preview

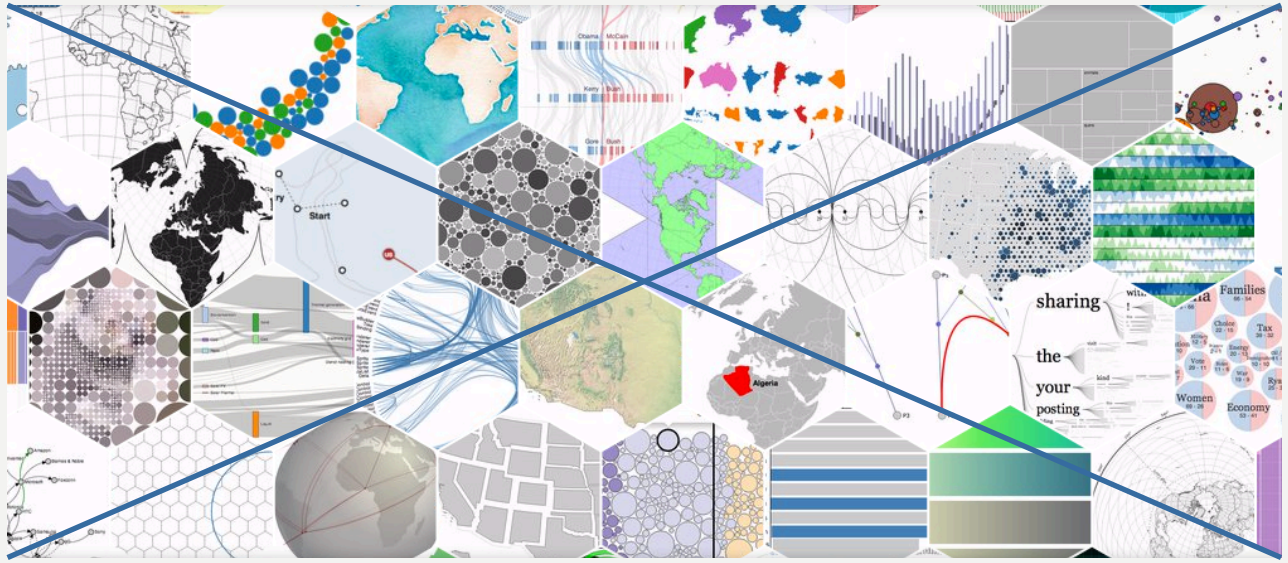
Contents

It is difficult to beat Graphviz and barcharts

For automatically generated graphs

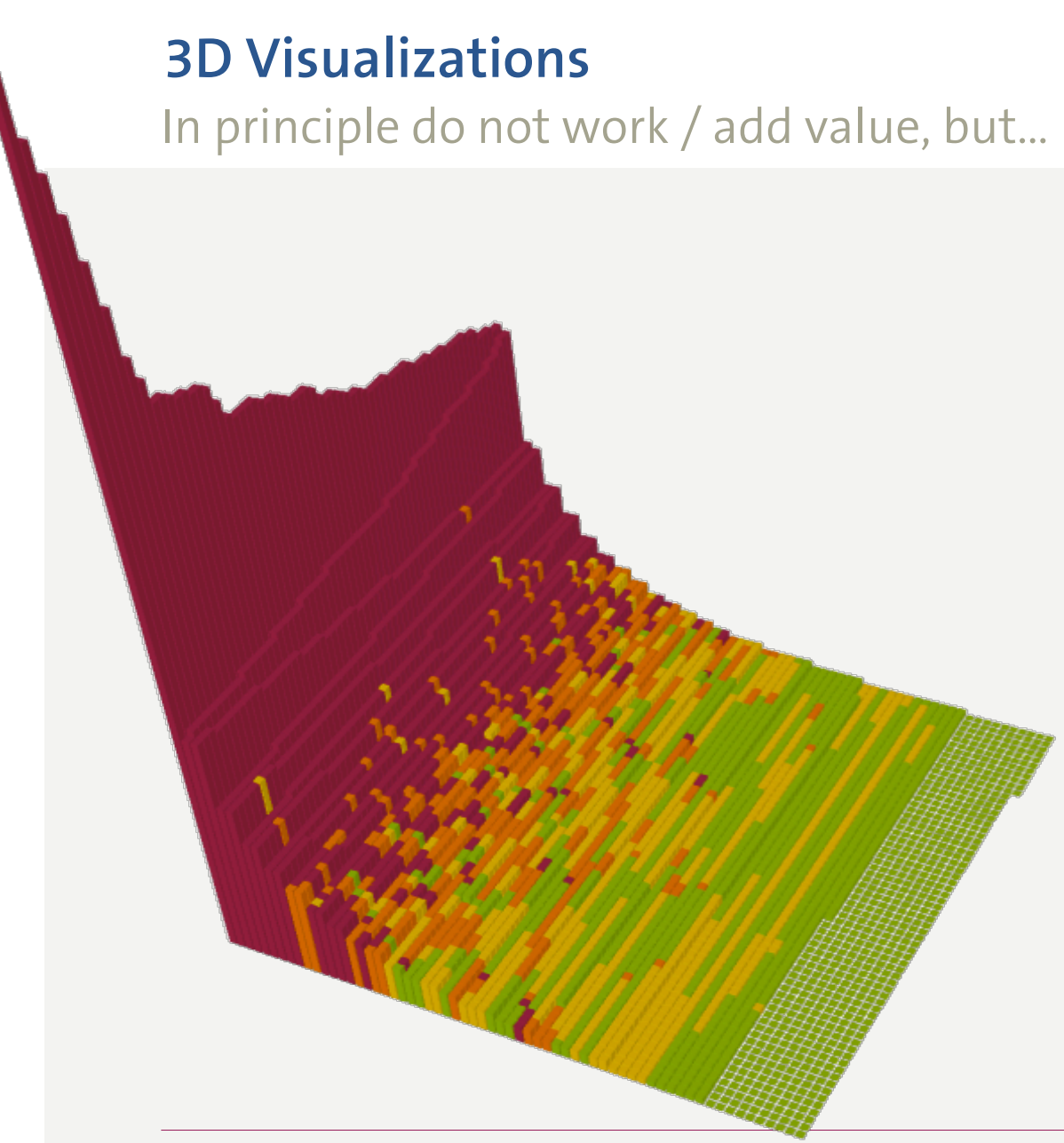
- The only visualizations working well for very diverse data sets sizes
- “Advanced” visualizations work very nicely for concrete data sets shown in the examples, but fail miserably if a data set is different

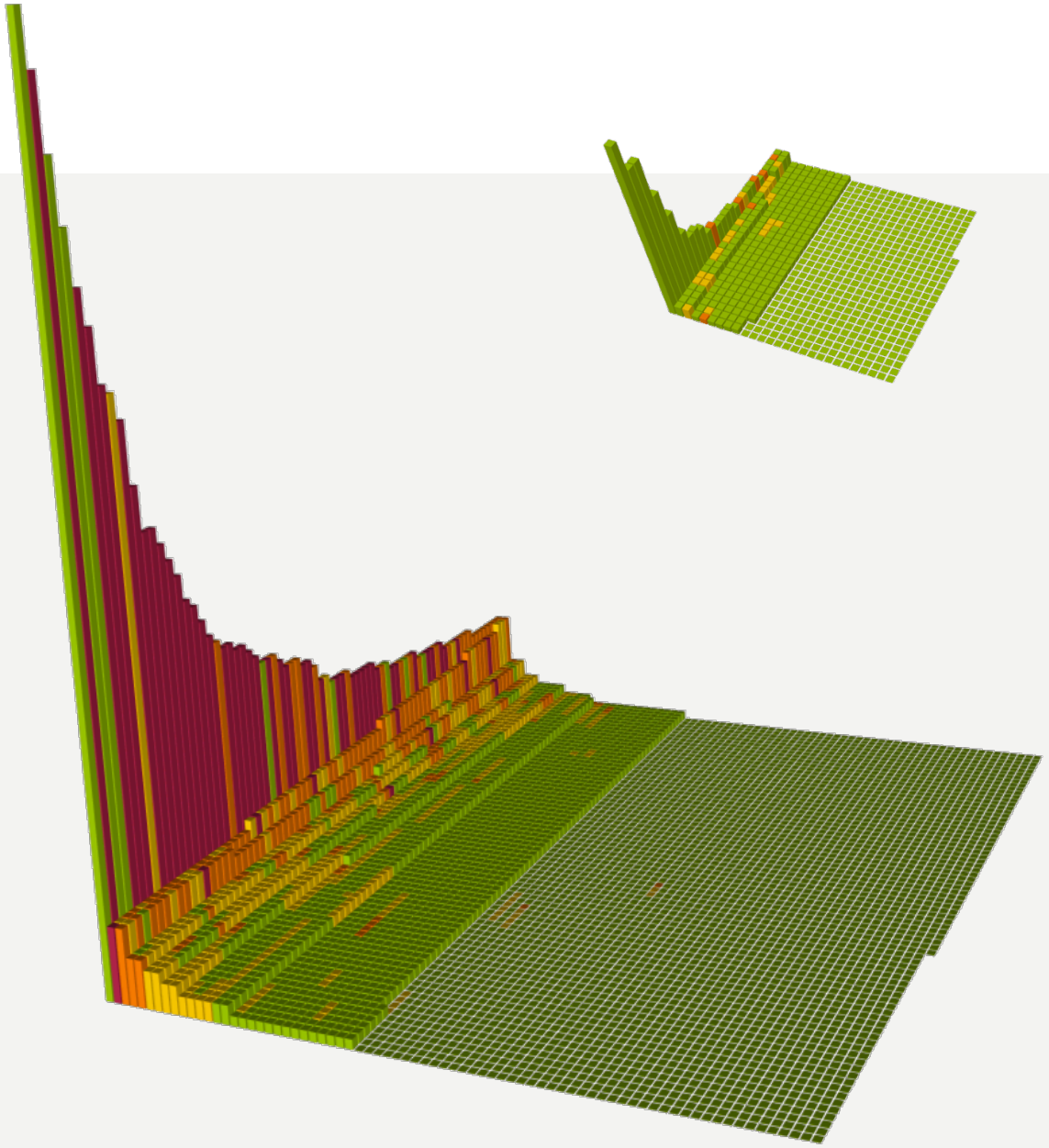




3D Visualizations

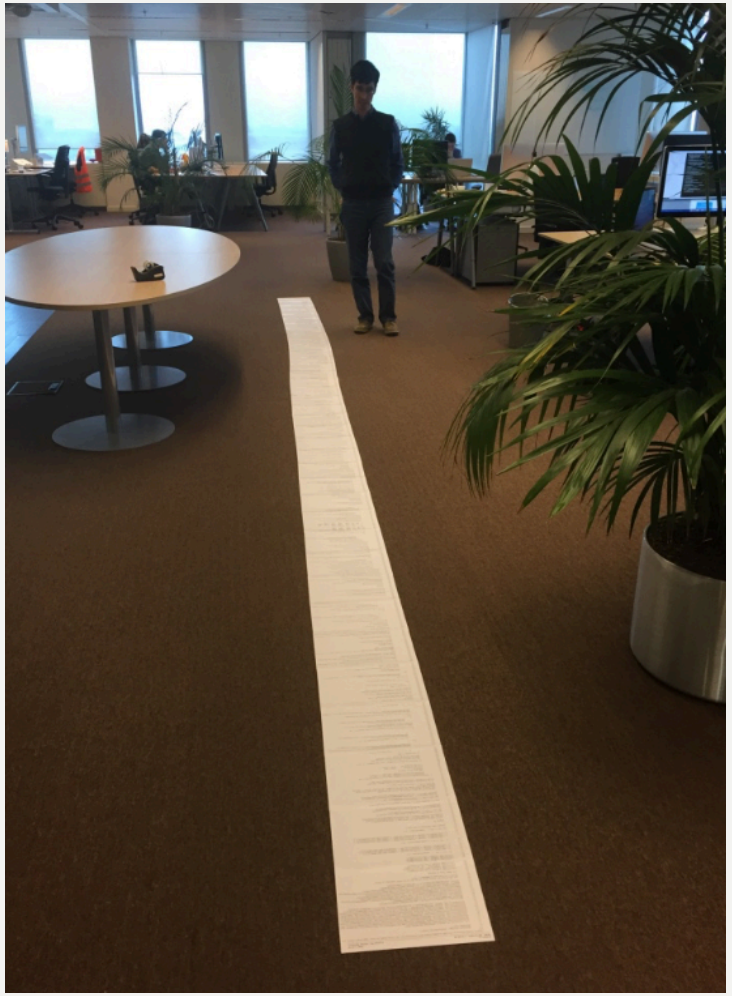
In principle do not work / add value, but...





Do not underestimate paper and glue

Making metrics tangible



Conclusions

- No silver bullets: basic principles apply everywhere, all the time
- Getting what you measure: “metric is a good servant but a bad master”
- The importance of context and structure: “story tells that the facts, facts do not tell the story”
- Human in the loop: analysis with metrics is a creative process
- Visualizing the metrics: again “no silver bullets”, but it pays of to keep experimenting