Architecting for Maintainability

Marco di Biase

@mardibiase

Who am I?

- PhD at TU Delft in the Software Engineering Research Group
- Technical Consultant/Researcher at Software Improvement Group in Amsterdam

// INTRODUCTION

About Software Improvement Group

GETTING SOFTWARE RIGHT
FOR A HEALTHIER DIGITAL WORLD

Benchmarking

Our software analysis database is the largest in the world, containing more than 25 billion lines of code.

Scientific research

Our dedicated research department develops our measurement models and contributes to advancements in the field.

Monitoring

Our proprietary technology and online platform provide continuous insight into software quality, cost and risk.

Management consultancy

We provide fact-based, actionable advice to help organizations turn their software into an enabler for growth.

TÜViT® Certification

SIG is the only company in the world with a laboratory accredited by TüViT to certify software for ISO 25010.

If you want to meet SIG @ De Delftse Bedrijvendagen sign up at ddb.tudelft.nl

Take a card here later for a ‘Coffee Date’

4 yrs ago
I was there

https://delftswa.gitbooks.io/desosa2016/content/guava/chapter.html

Type to search

Delft Students on Software Architectur…

Guava - As it Currently is

Bastiaan Reijm, Marco di Biase, Qianqian Zhu, Luca Pascarella

Delft University of Technology

Abstract

Guava is a Collections and Utility library, focusing on compleme
present the current state of Guava for newcomers and interested
and elucidate the architecture by analysing several key viewpoin
present the current plans for Guava so that contributors and use
is going. We briefly discuss what to do when contributing. Finall
experiences at we gathered while analysing this project. Our aim
of Guava.

Table of Contents

Why am I giving this lecture?

- What are Software Metrics and what is their usefulness?
- What can these metrics do for you
- How metrics can measure architectural aspects

Measures

A measure is the **number** or **symbol** assigned to an **entity** by mapping the empirical world to the formal, relational world in order to **characterize an attribute**

Fenton - Software Metrics: a rigorous and practical approach

https://shop.imetec.com/it/5467q-pesapersone-meccanica-imetec-medical-pro.html

A measure is the **number** or **symbol** assigned to an **entity** by mapping the empirical world to the formal, relational world in order to **characterize an attribute**

- What is the number/measure?

86kg

- What is the entity?

- What is the attribute?

weight

How to measure software?

Let’s try our hand at code

How would you measure software “size”?

- Lines of code
- Man-months / man-years
- Function Points (yeah that’s old)

Example

```
⇒  cloc pandas
     929 text files.
     922 unique files.
     138 files ignored.

github.com/AlDanial/cloc v 1.80  T=11.04 s (77.5 files/s, 39769.0 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
Python                         685          72589          76047         230106
Cython                          39           5439           5904          17127
HTML                            96           1799            176          15074
C                               10           1368            923           7588
C/C++ Header                    23            739            930           3041
JSON                             2              0              0              2
-------------------------------------------------------------------------------
SUM:                           855          81934          83980         272938
-------------------------------------------------------------------------------
```

github.com/AlDanial/cloc

Software size

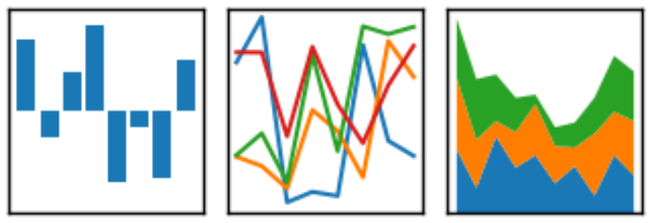
- What is the number/measure?

 # LOC

- What is the entity?

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



- What is the attribute?

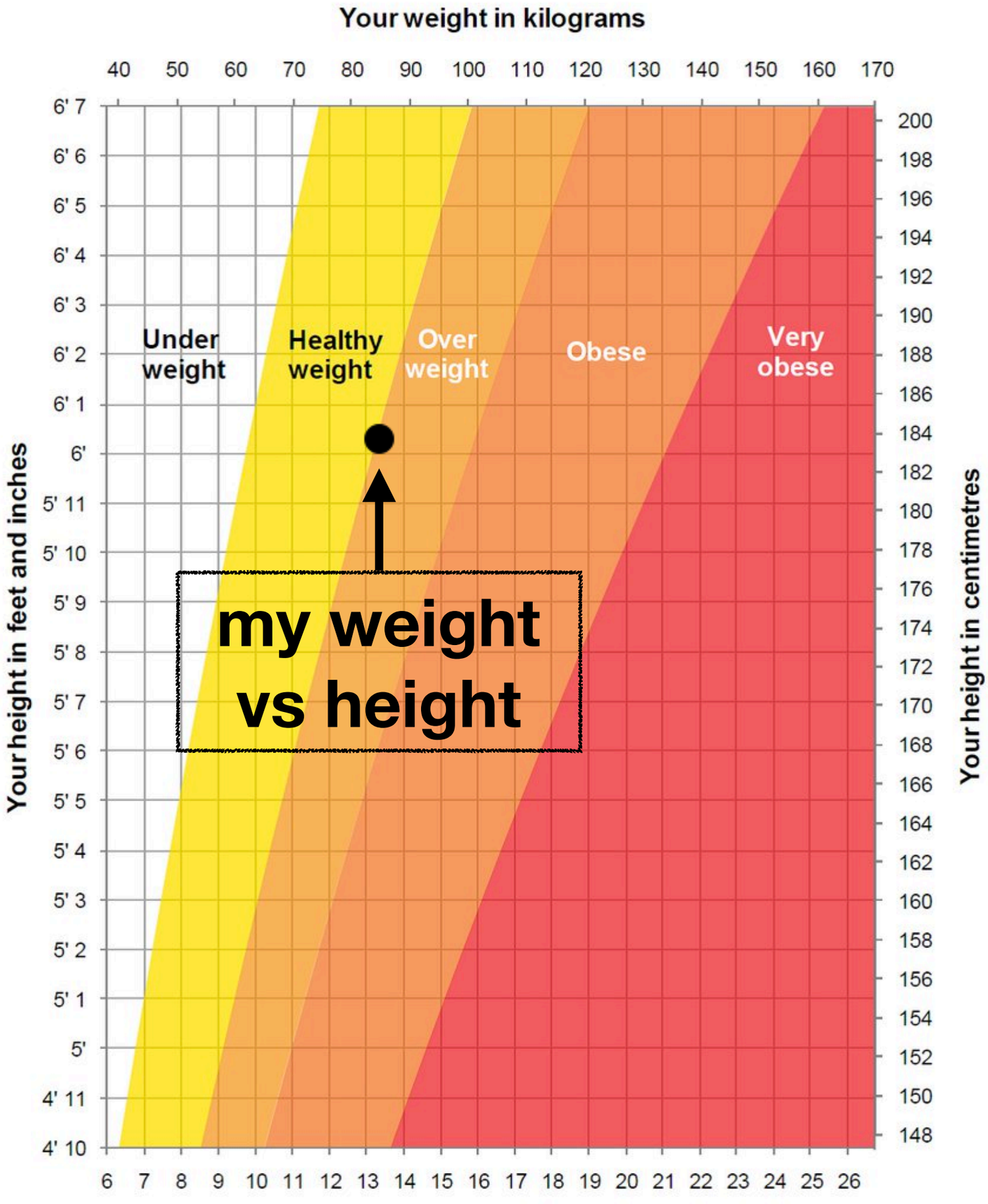
Code size / Volume

How to know if a project has a “normal” measure?

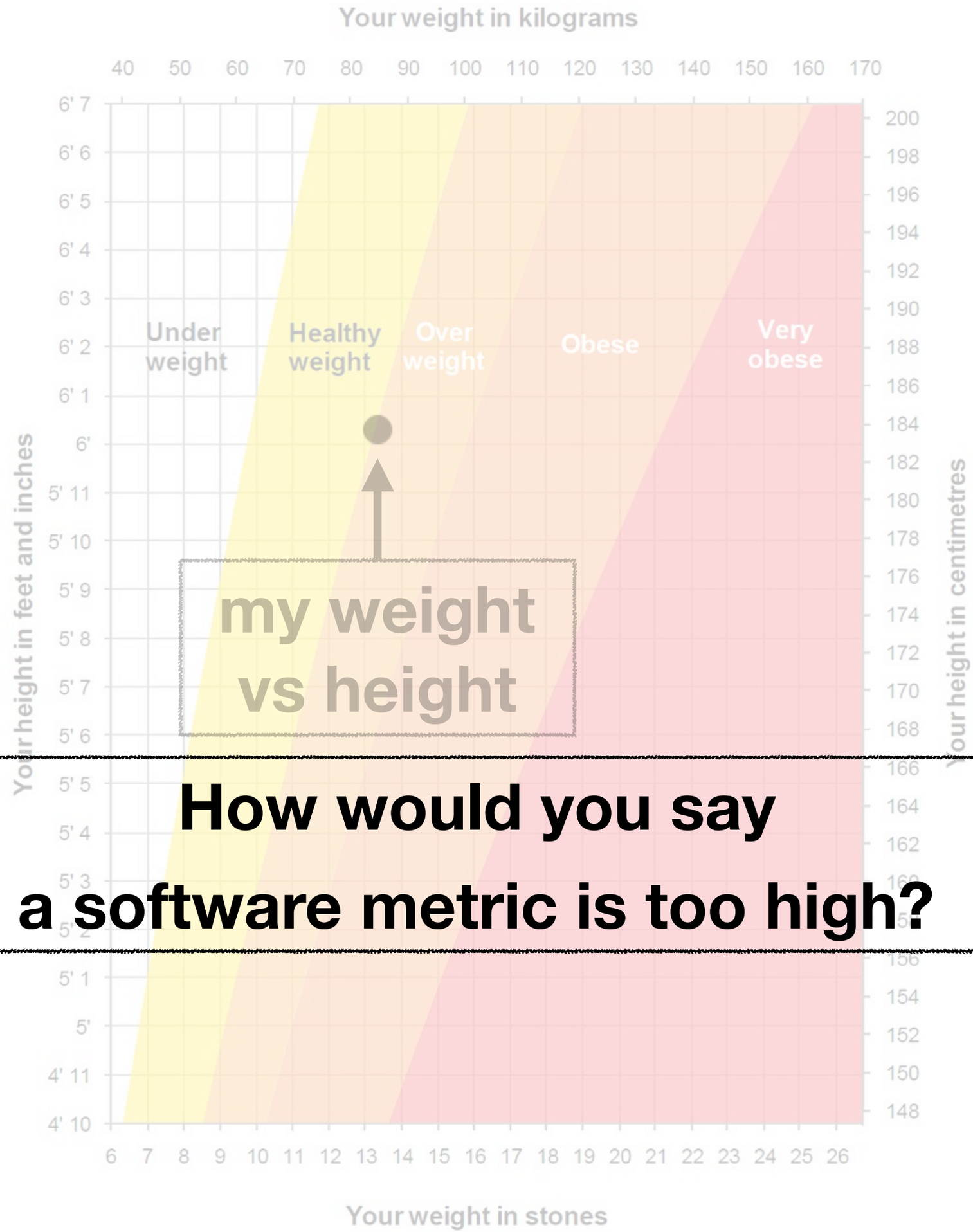
Definition of “normal”

- Only relatively to your project
- Your application
- Your context

How would you say
my weight is too much/too little?



https://www.nhs.uk/live-well/healthy-weight/height-weight-chart/



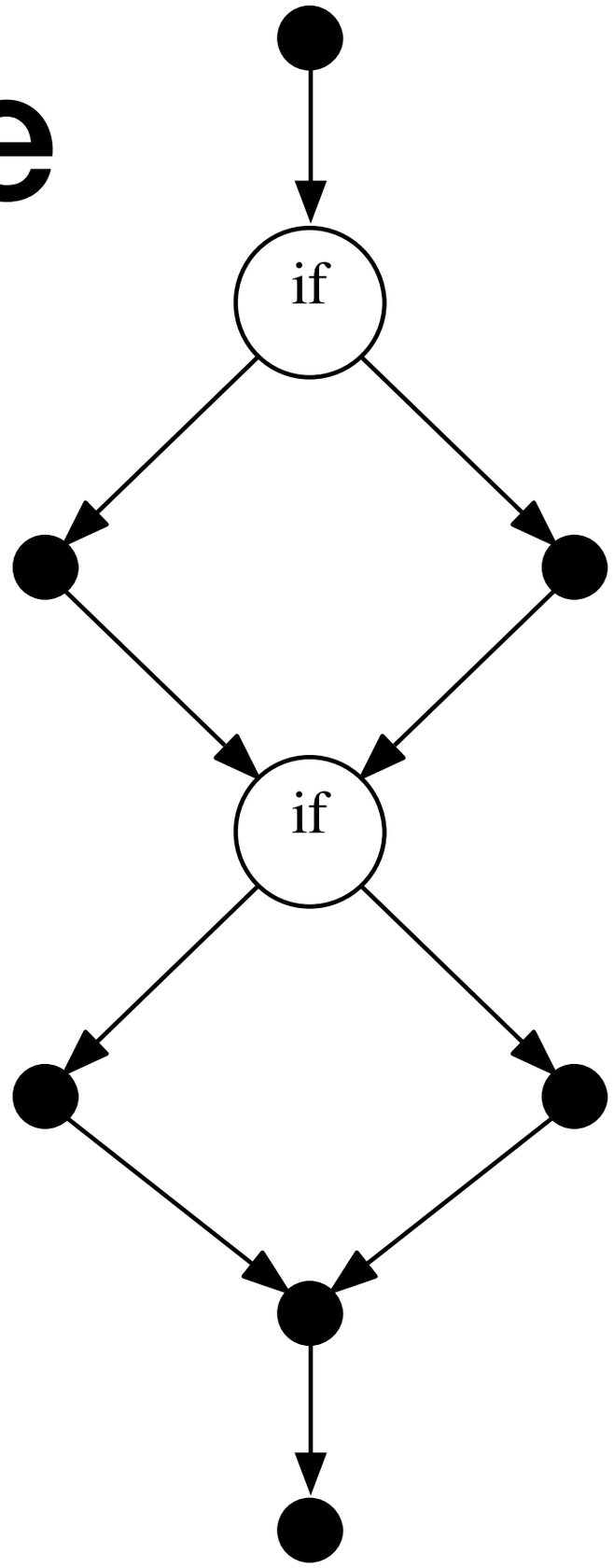
How would you say

a software metric is too high?

Example: code complexity

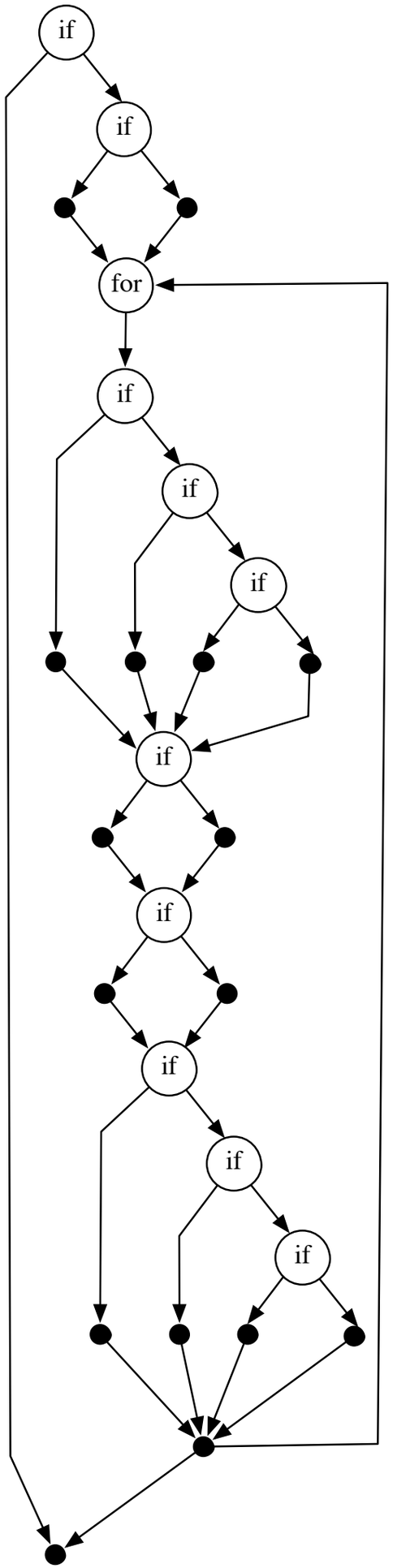
- McCabe Cyclomatic Complexity (McCabe, 1976)
- Measures the number of independent paths in source code
- Basically, it counts if-statements, for, while, etc.
- Indicates how difficult a program or module will be to test and maintain

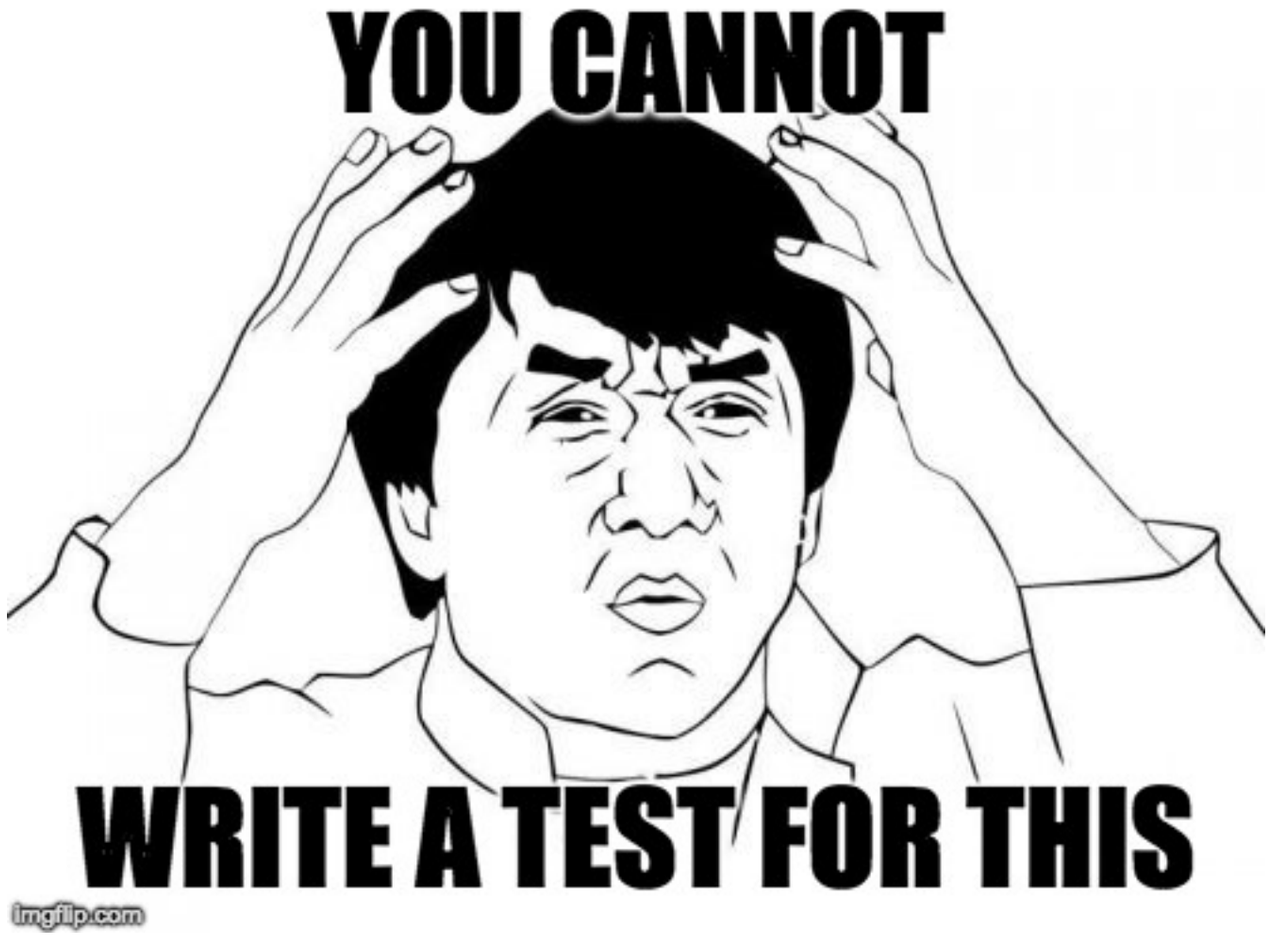
Example



When and why McCabe complexity is too much?

- McCabe suggested that when complexity is greater than **10** in a module (i.e. a method, function, etc.), it may be problematic
- The minimum number of tests needed to cover all independent execution paths is the number of branch points plus one


if
if
for
if
if
if
if
if
if
if
if
if
if


YOU CANNOT
WRITE A TEST FOR THIS
imgflip.com

Pitfalls

Developers start to “treat" metrics

Bouwers, Visser, & Van Deursen - Getting what you measure - ACM Queue May 2012

GOODHART'S LAW

WHEN A MEASURE BECOMES A TARGET,
IT CEASES TO BE A GOOD MEASURE

IF YOU MEASURE PEOPLE ON...	NUMBER OF NAILS MADE	WEIGHT OF NAILS MADE
THEN YOU MIGHT GET	1000's OF TINY NAILS	A FEW GIANT, HEAVY NAILS

http://www.sketchplanations.com/post/167369765942/goodharts-law-when-a-measure-becomes-a-target

Responsible use of the metrics is just as important as collecting them in the first place

https://blog.codinghorror.com/a-visit-from-the-metrics-maid/

How to use metrics with a logic?

Goal/Question/Metric

- Developed by Victor Basili to answer questions associated with any software process

- Measurement, to be effective, must be:

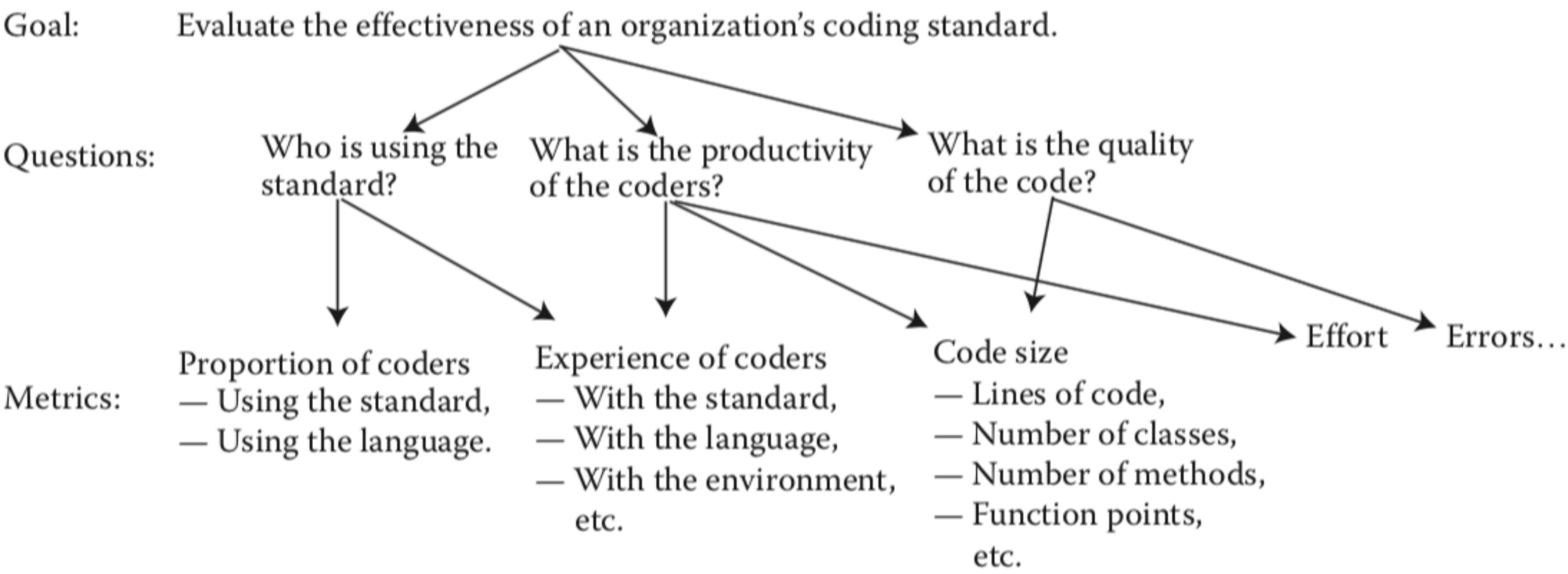
- Focused on specific goals;

- Applied to all life-cycle products, processes and resources;

- Interpreted based on characterization and understanding of the organizational context, environment and goals

Basili, Caldiera, Dieter Rombach - The Goal/Question/Metric Approach (1994)

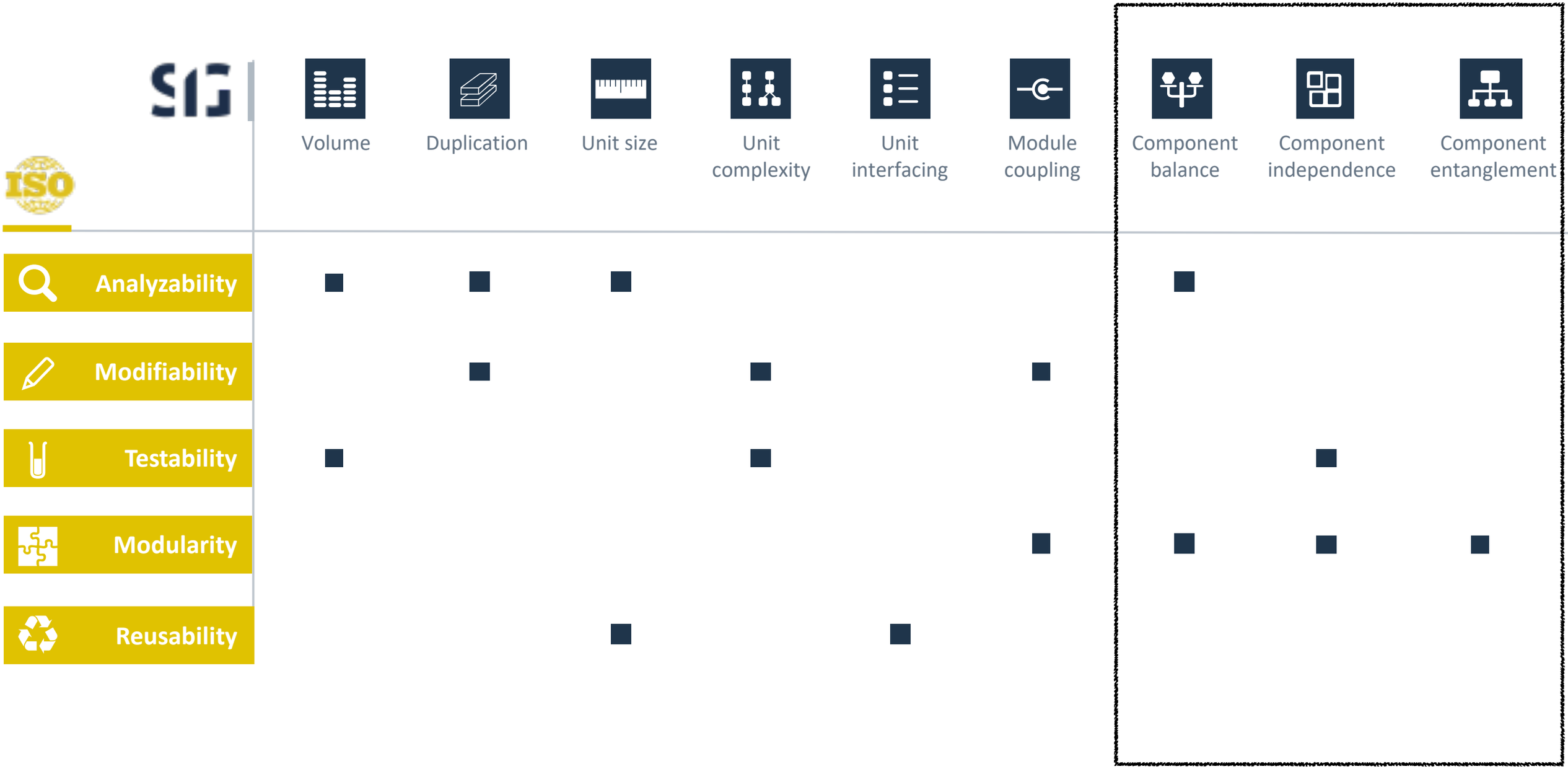
Goal/Question/Metric



Fenton - Software Metrics: a rigorous and practical approach

How can we apply this approach to Software Architecture?

Architectural metrics

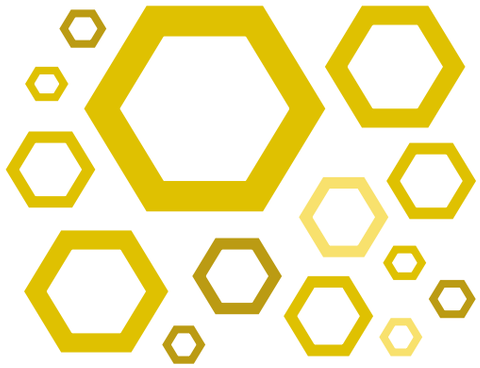


Component Balance

Example

All changes in a single large component

Most changes in a single large component



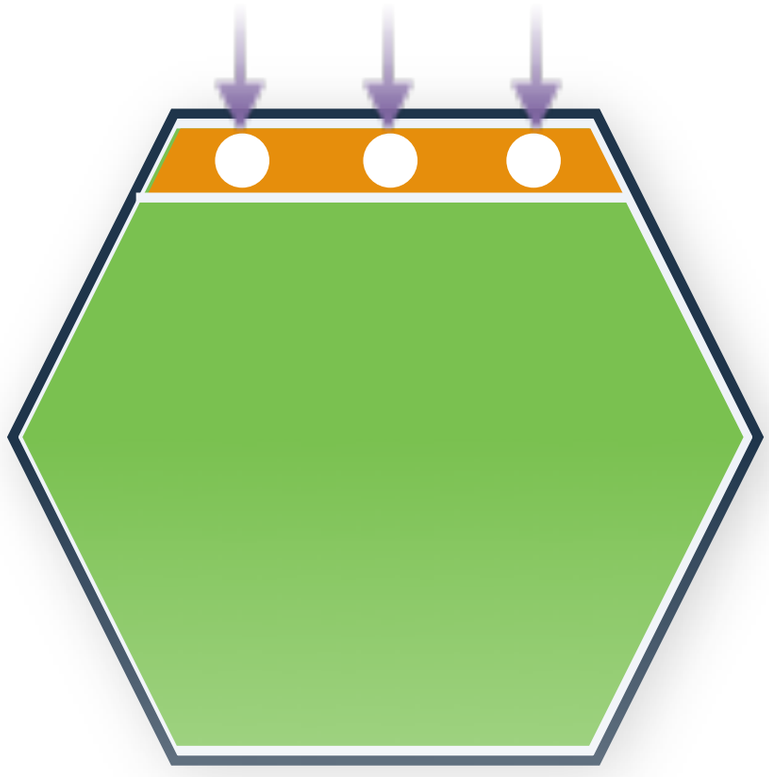
Many changes scattered across multiple components

Changes isolated to one or two components of limited scope

Component Independence

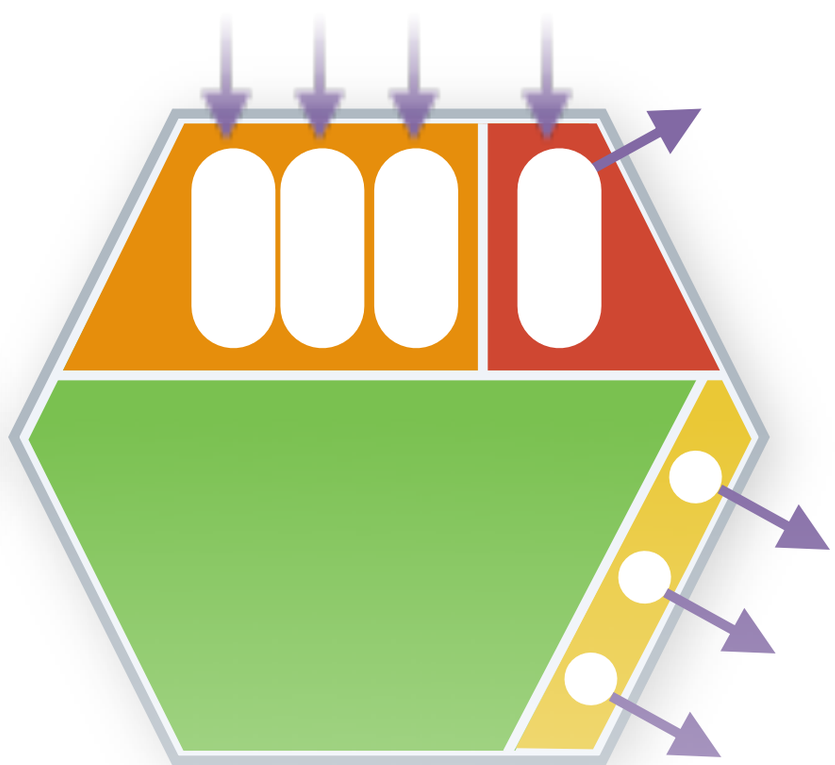
Example

Recommended practice



Component

Inadvisable practice

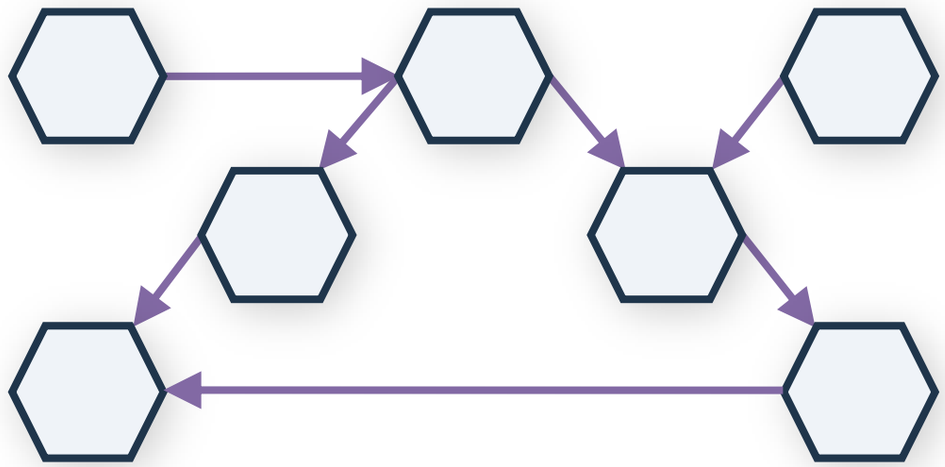


Component

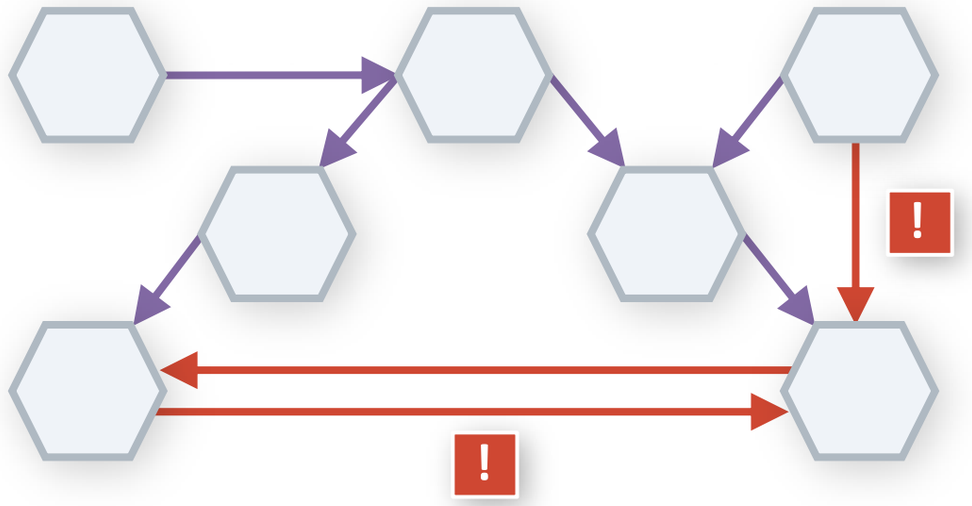
Component Entanglement

Example

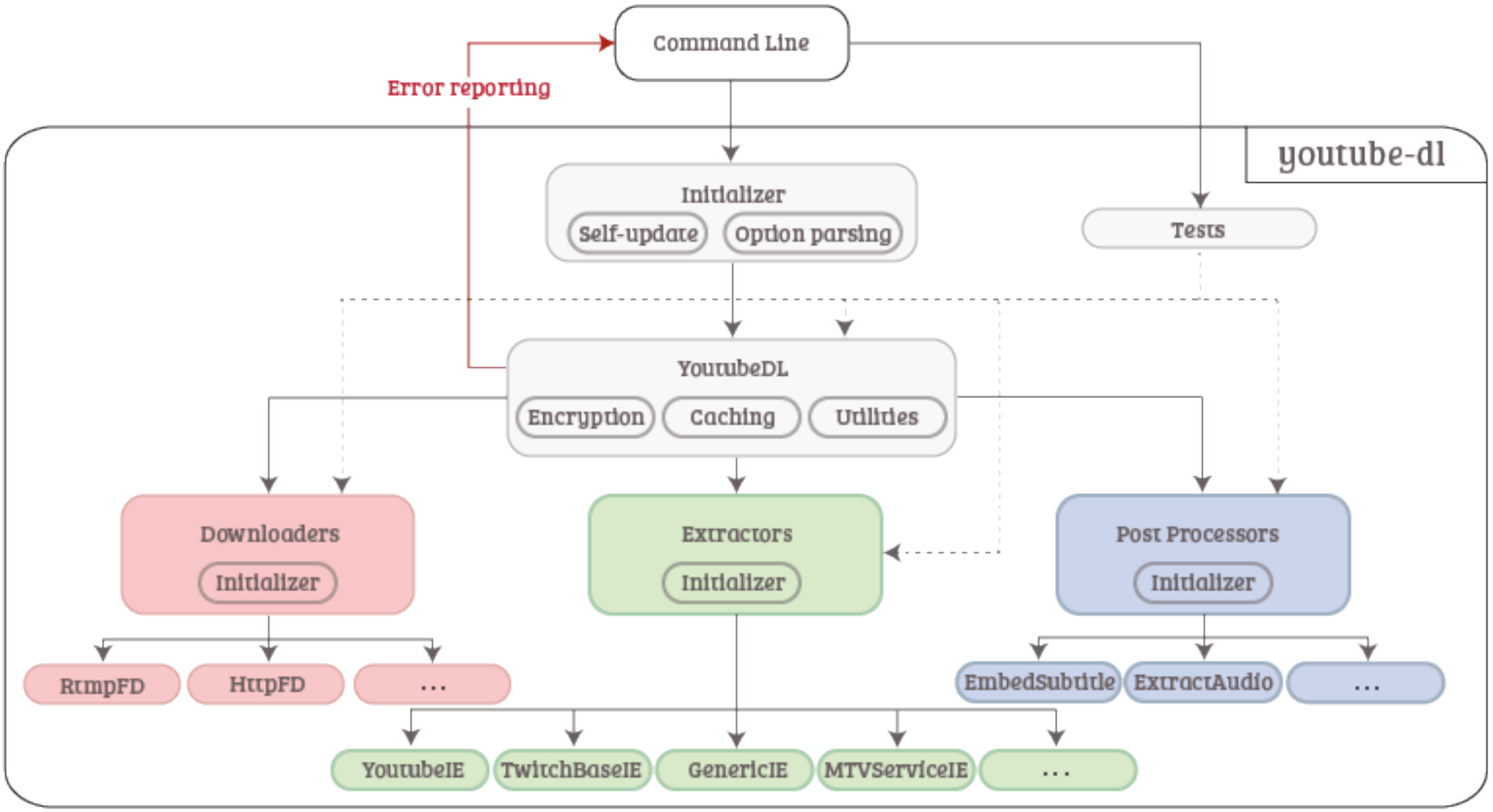
Recommended practice



Inadvisable practice



YouTube-DL Architecture



https://delftswa.gitbooks.io/desosa2016/content/youtube-dl/chapter.html

Architectural metrics

- Q: How can we measure if the architecture is balanced?
- A: LOC per architectural component

Architectural Component	#LOC Python
Downloader	2684
Extractors	118220
Postprocessor	1006
Core	14283

Architectural metrics

- Q: How can we measure if the architecture is balanced?
- A: LOC per architectural component

Architectural Component	#LOC Python
Downloader	2684
Extractors	118220
Postprocessor	1006
Core	14283

86% of the code is in Extractors

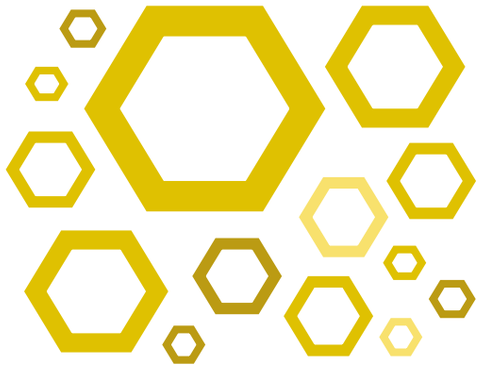
How do you gather this in a single metrics?

Component Balance

Example

All changes in a single large component

Most changes in a single large component



Many changes scattered across multiple components

Changes isolated to one or two components of limited scope

Component Balance

Example

For the example of youtube-dl, architectural component balance is 0.07 (scale is from 0 to 5)

Now you probably also understand why it's relevant that you define components properly!

How do you solve unbalanced architectural components?

Possible solutions to unbalanced architectural components

- The biggest architectural component could become a separate project

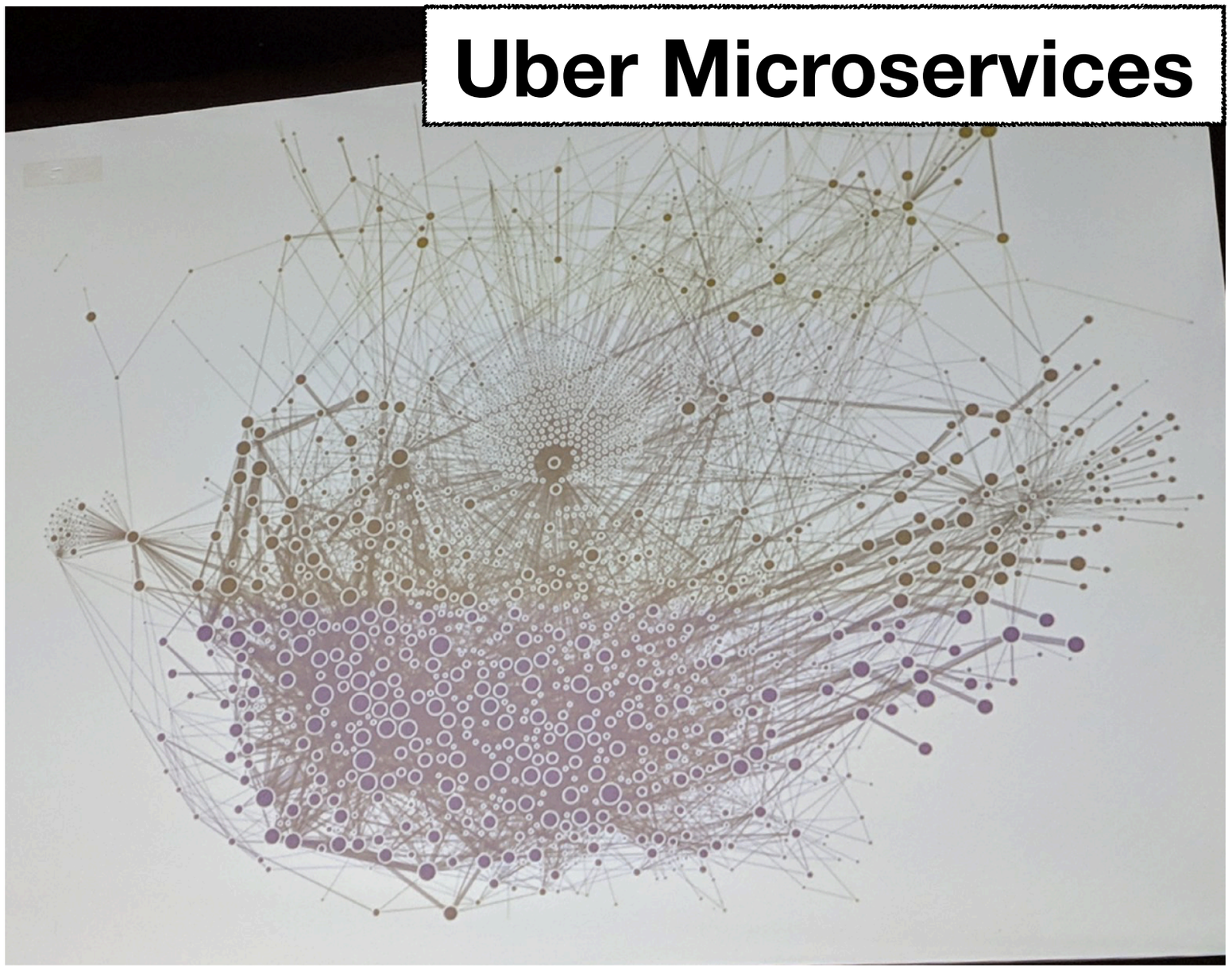
- The biggest architectural component could be refactored to generate a hierarchy that allows for abstraction

- split up in sub-components focused on specific goals

- in practice: use Abstract Factory Design Pattern

Possible architecture alternative: microservices

Let's always use a grain of salt and not end up with this though



https://twitter.com/msuriar/status/1110244877424578560

Possible architecture alternative: microservices

- Q: What made you change your mind about a paradigm?

- A: **Microservices**. They seemed really cool until I worked on a few large projects using them. Disaster so epic I watched most of engineering Management walk the plank.

- The biggest cause lies in inter-service communication

- Another big issue is the service explosion itself. Keeping 30 backend applications up to date and playing nice with each other is a full time job

https://news.ycombinator.com/item?id=19298557

Not everything
that can be counted
counts, and not
everything that
counts can be
counted.

–Albert Einstein

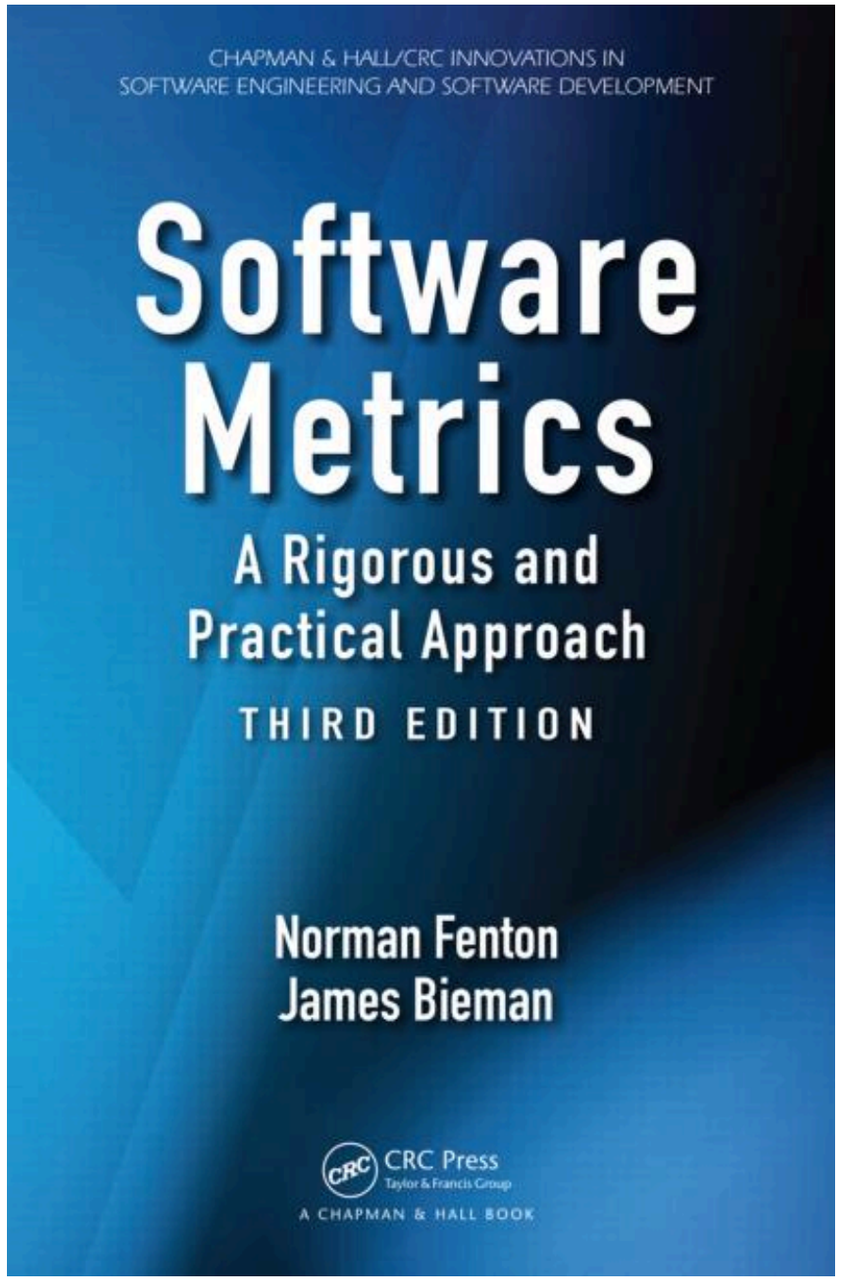
But...

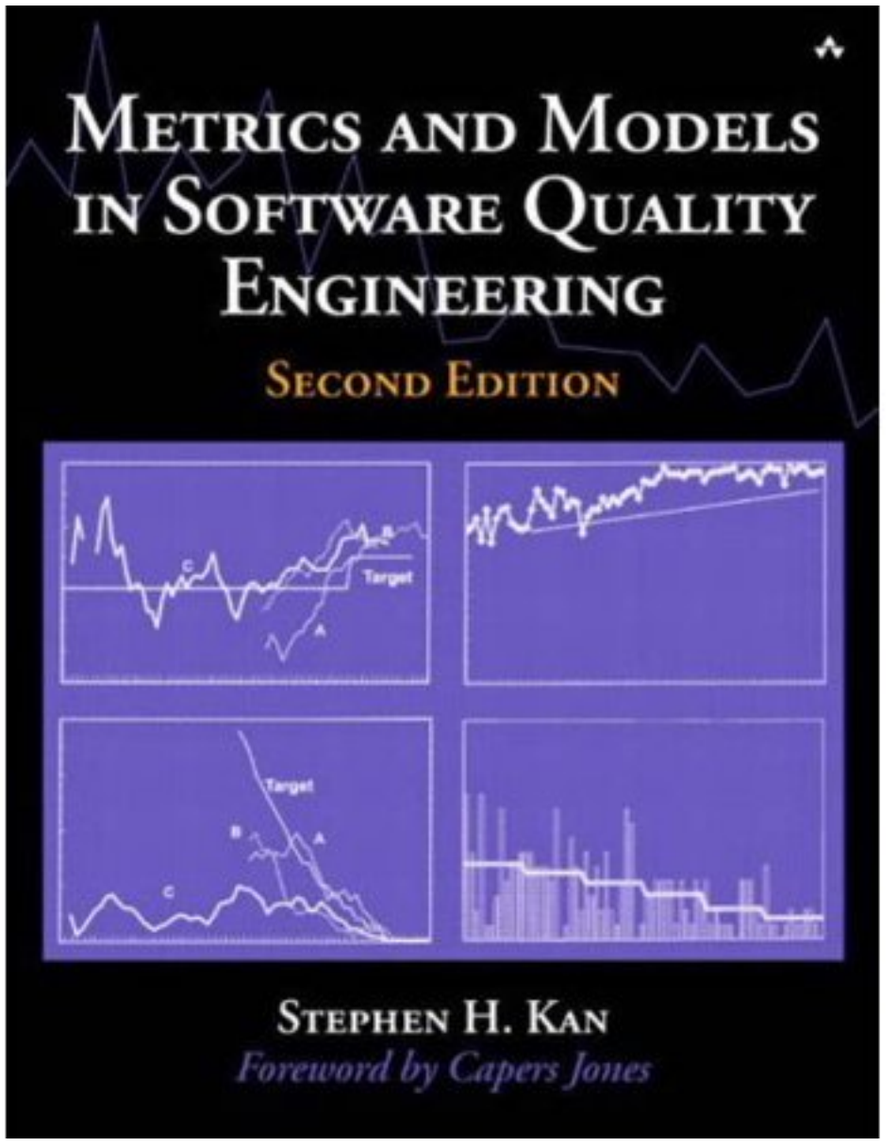
You can't control what you can't measure

(Tom DeMarco)

https://izquotes.com/quote/tom-demarco/you-can-t-control-what-you-can-t-measure-223617

Useful resources





Questions? 

Using Software Metrics to improve Software Quality

How do you check if the bicycle you're buying is good?

source: every dutch backyard

source: bikeexchange.co.nz

How it looks?
Rust?
Are the wheels straight?
Punctures?
Are the lights working?
...

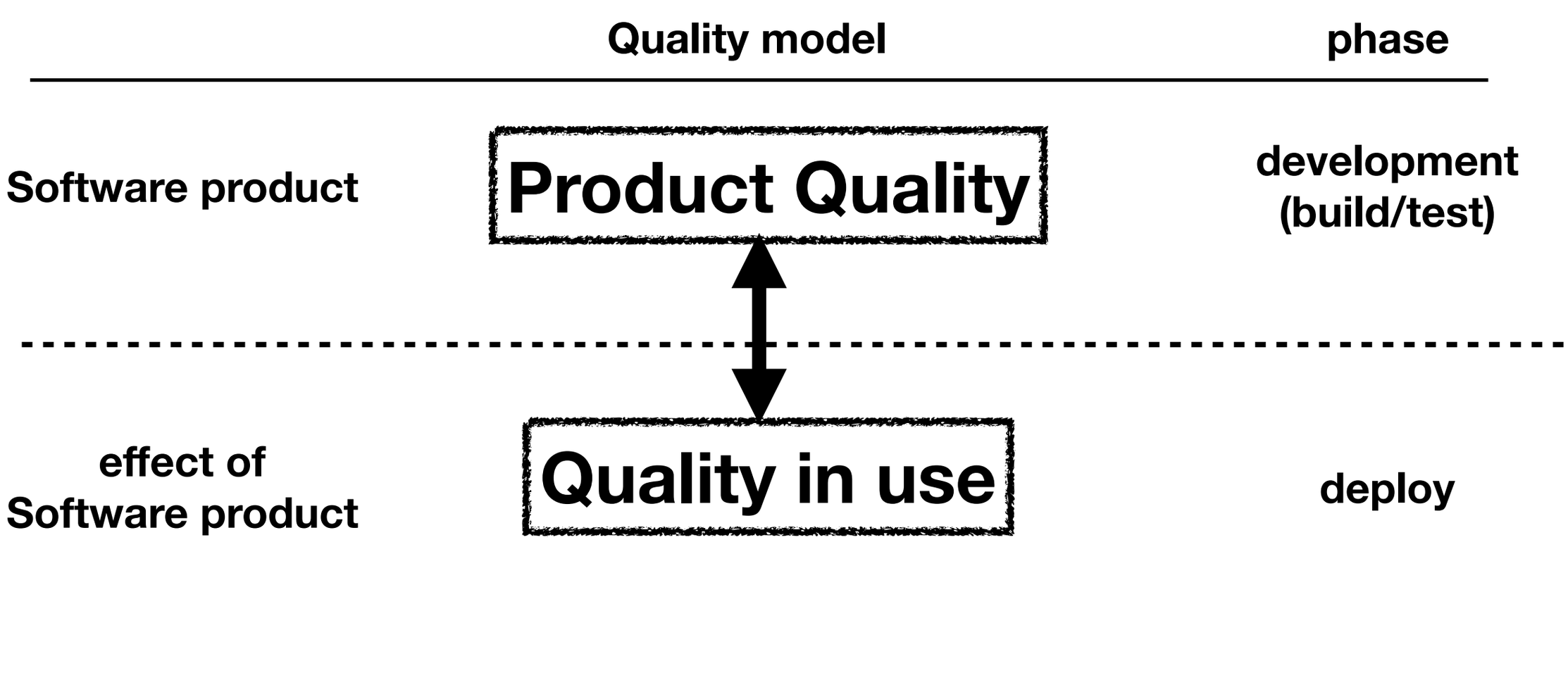
source: every dutch backyard

source: bikeexchange.co.nz

What are “features” you would use to check if a Software is good?

ISO/IEC 25010

- Systems and software Quality Requirements and Evaluation
- Byproduct of two previous ISO models (9126 and 14598)



Product Quality

8 main categories

Sw Maintainability Sub-characteristics

Maintainability

The degree of effectiveness and efficiency with which a system can be modified by the intended maintainers.

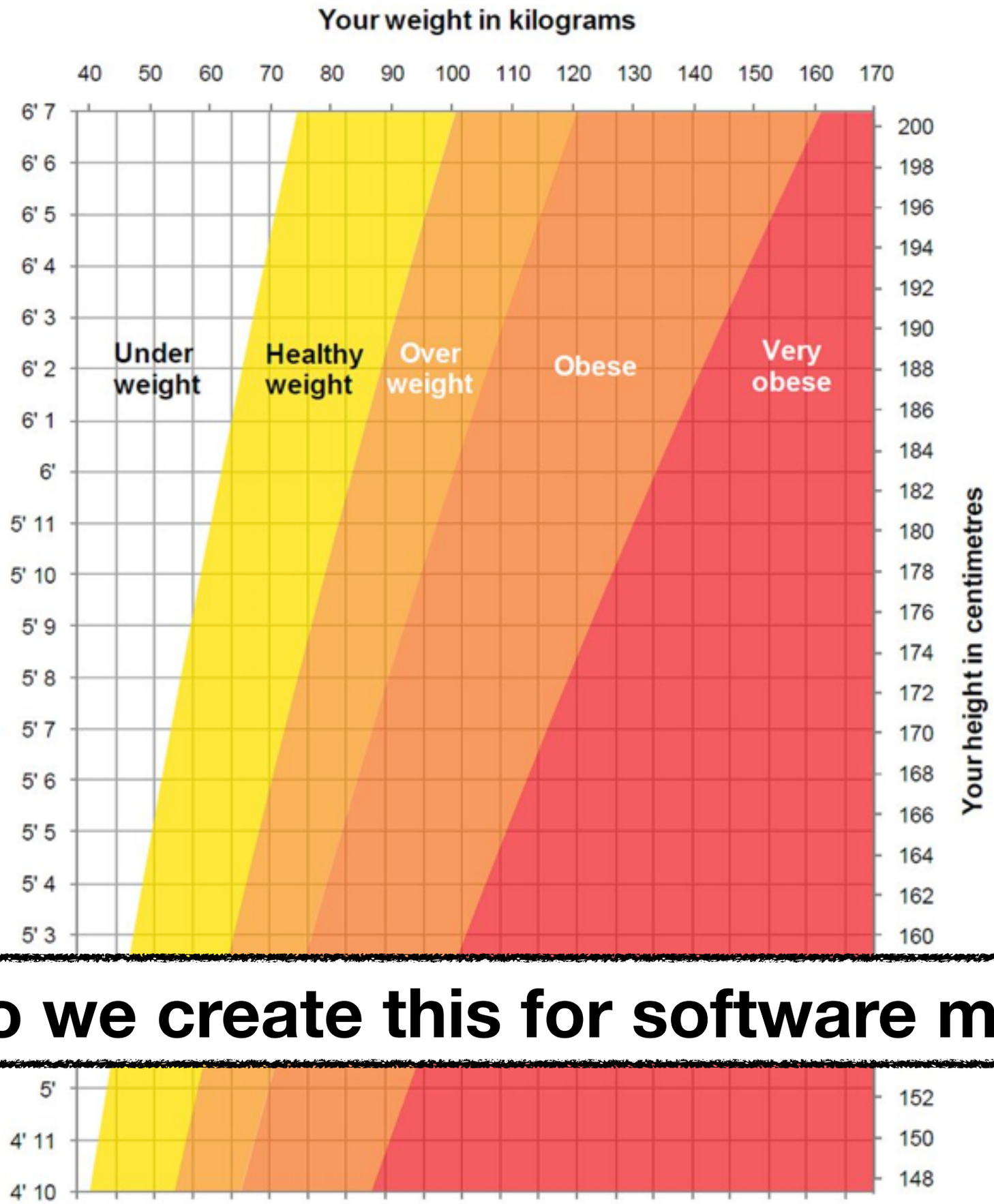
SUB-CHARACTERISTIC	DEFINITION
Analyzability	Degree to which one can navigate a system's structure and assess the impact of an intended change.
Modifiability	Degree to which the system can be modified without introducing defects or degrading system quality.
Testability	Degree to which test criteria can be established and tests can be performed to determine whether those criteria have been met.
Modularity	Degree to which the system is composed of components such that a change to one component has minimal impact on other components.
Reusability	Degree to which an asset can be used in more than once place within the same code base.

Recall the GQM?

- Characterize Maintainability
- of a Software Product
- from the perspective of a project manager
- in the context of risk assessment

SIG Maintainability Model

ISO	Volume	Duplication	Unit size	Unit complexity	Unit interfacing	Module coupling	Component balance	Component independence	Component entanglement
Analyzability	■	■	■				■		
Modifiability		■		■		■			
Testability	■			■				■	
Modularity						■	■	■	■
Reusability			■		■				



How do we create this for software metrics?

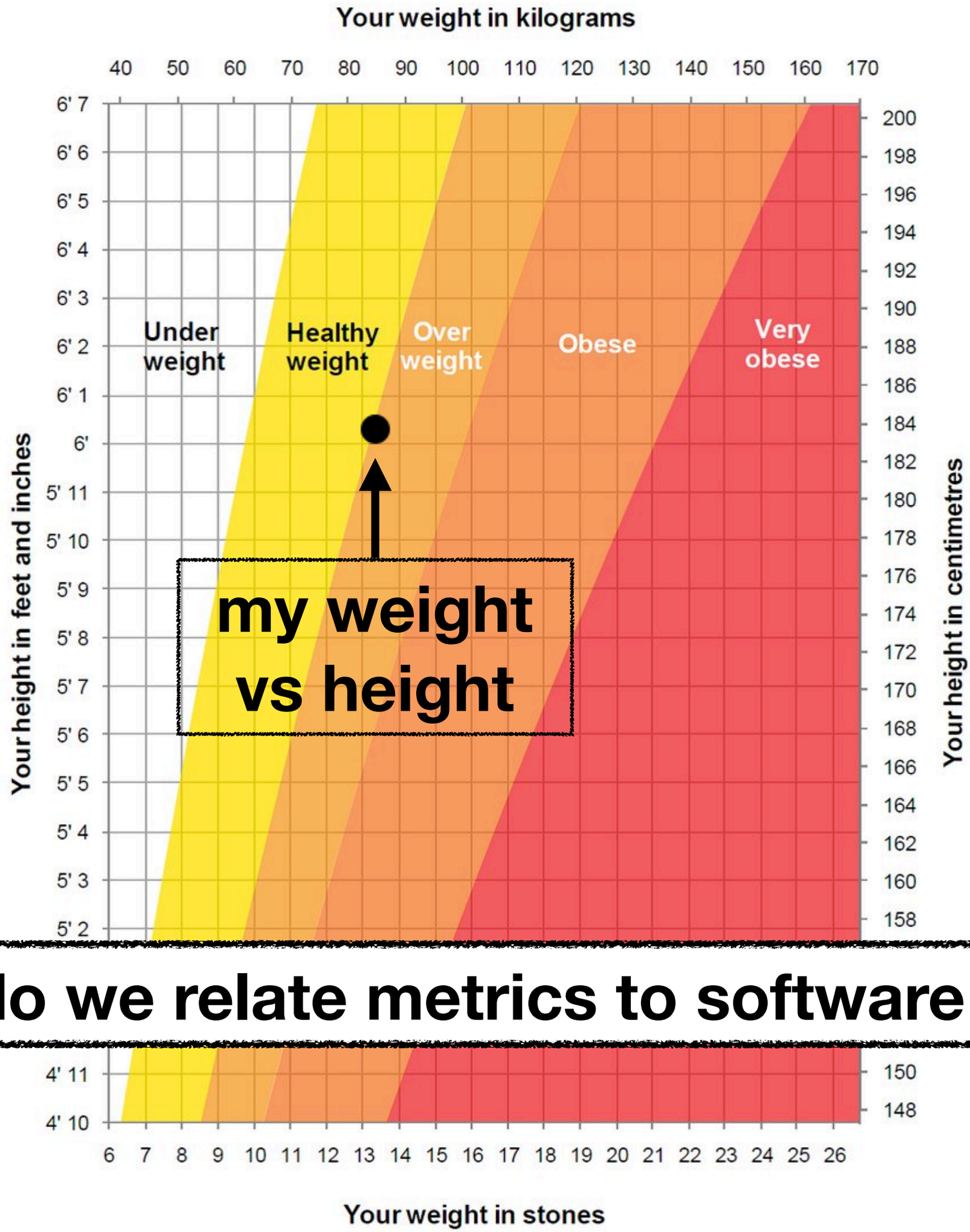
Software metric benchmarking

1. Measure system metrics
2. Summarize measurements
3. Derive thresholds that bring out metric variability
4. Round the thresholds

For example, McCabe Complexity has the following thresholds:

Low	**up to 5**
Medium	**6 to 10**
High	**11 to 25**
Very High	**> 26**

Alves, Ypma, Visser - Deriving metric thresholds from benchmark data - ICSM 2010



How do we relate metrics to software size?

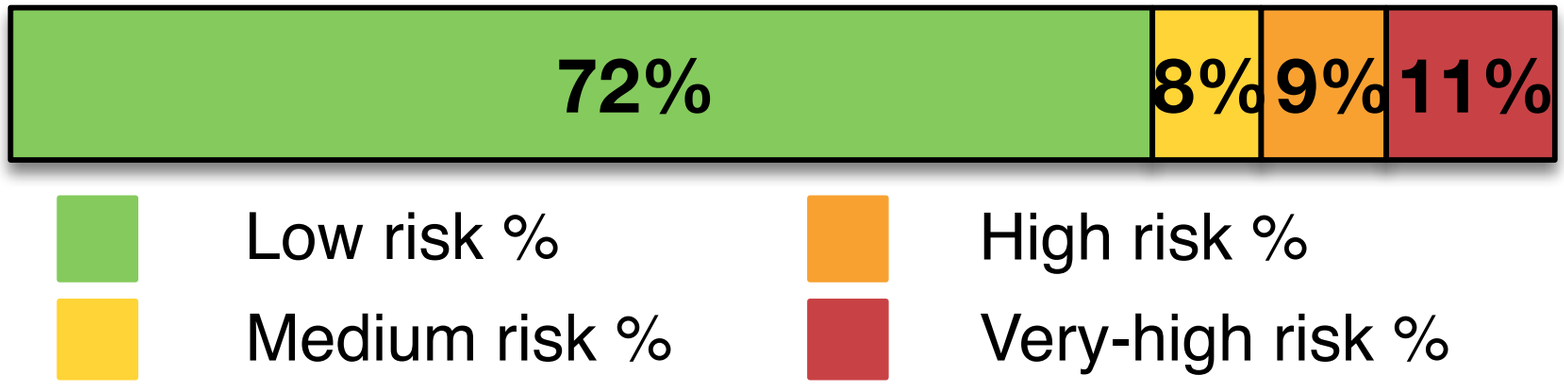
Quality (risk) profiles

McCabe Complexity

Low	**up to 5**
Medium	**6 to 10**
High	**11 to 25**
Very High	**> 26**

1) Sum lines of code per each category
2) Weight against system volume in LOC

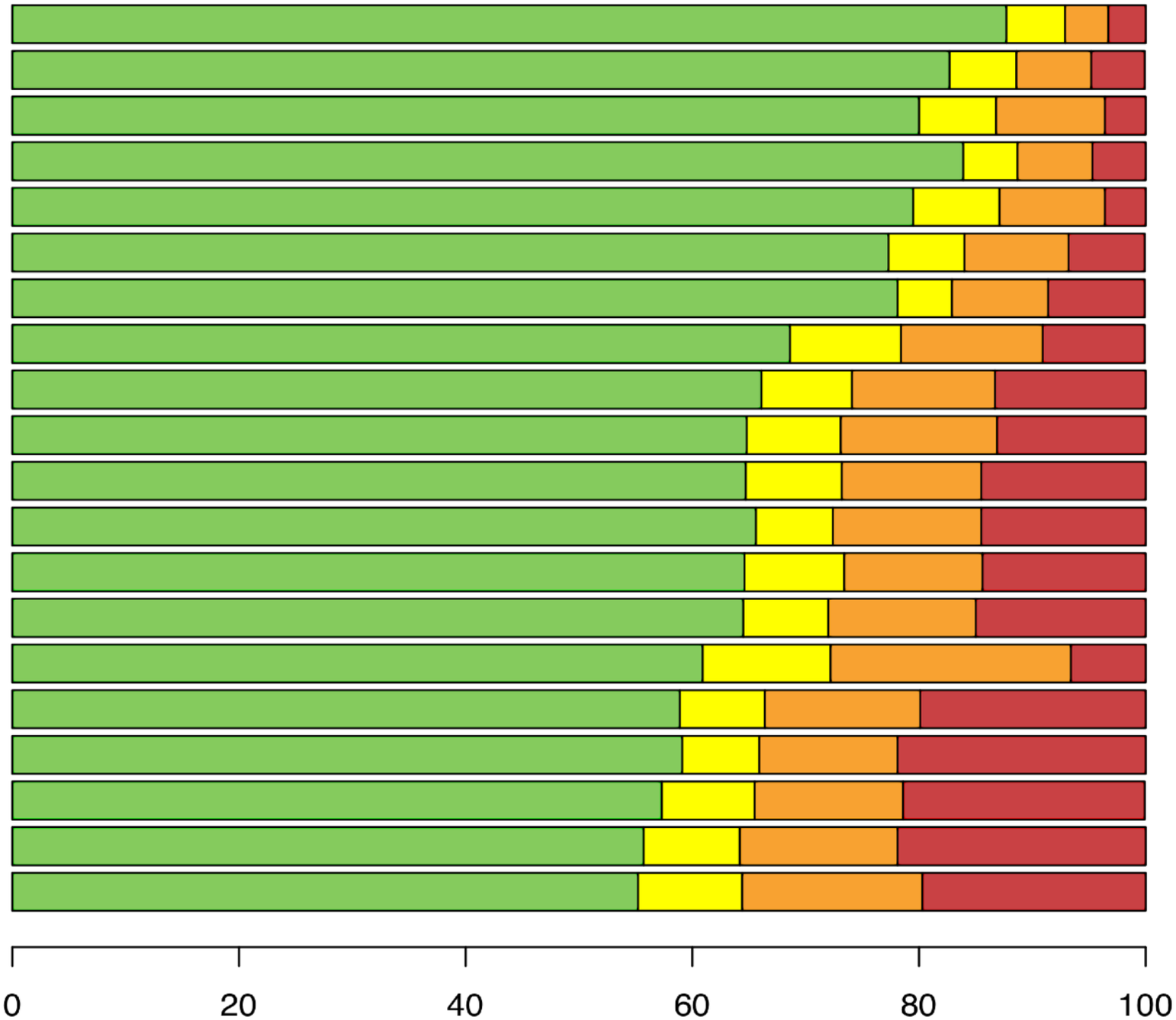
Risk Profile for Unit Complexity (example)



Heitlager, et. al. A Practical Model for Measuring Maintainability, QUATIC 2007

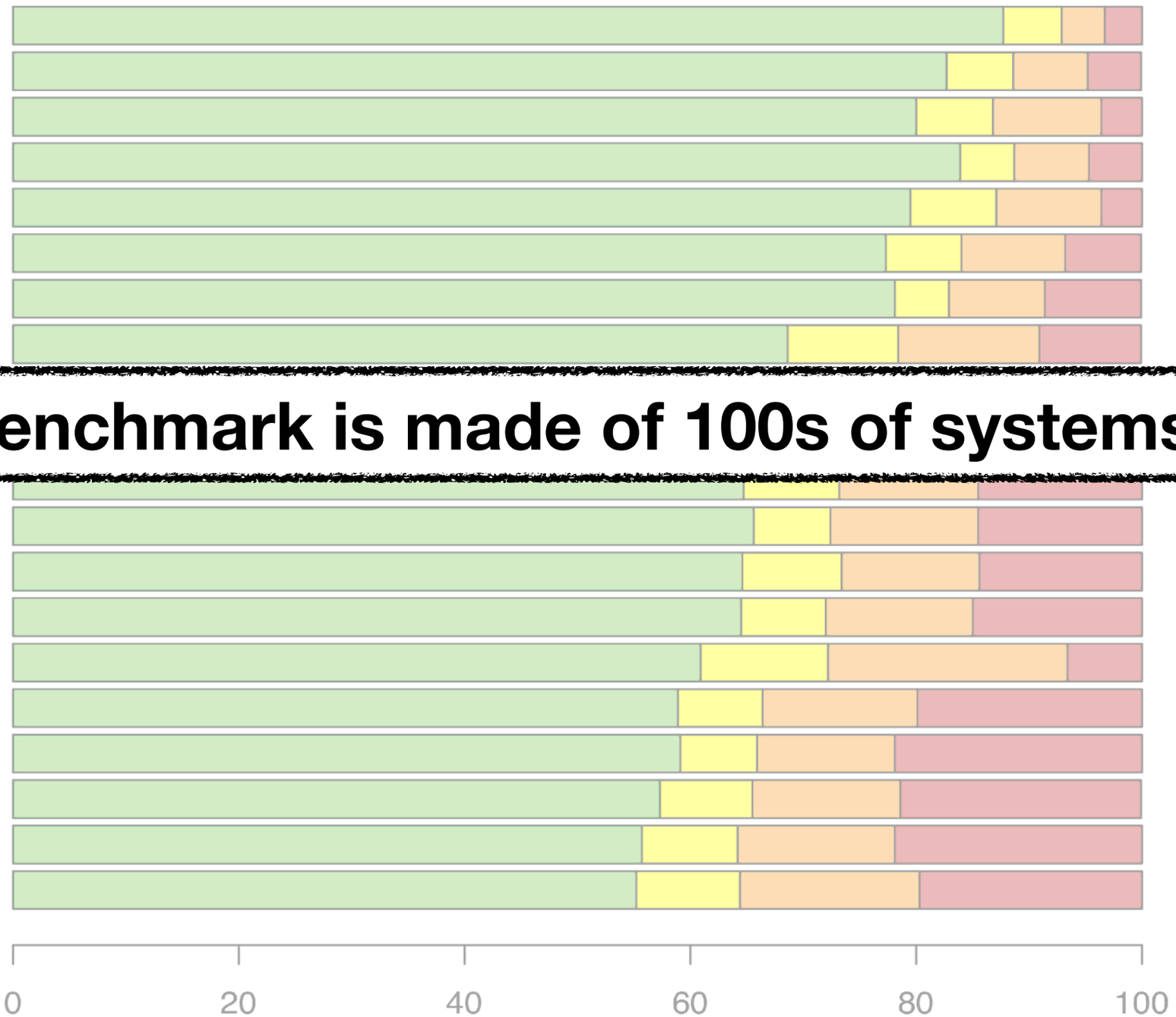
How do you compare risk profiles among systems?

Rank profiles



Alves, et. al., Benchmark-based Aggregation of Metrics to Ratings, IWSM / Mensura 2011

Rank profiles



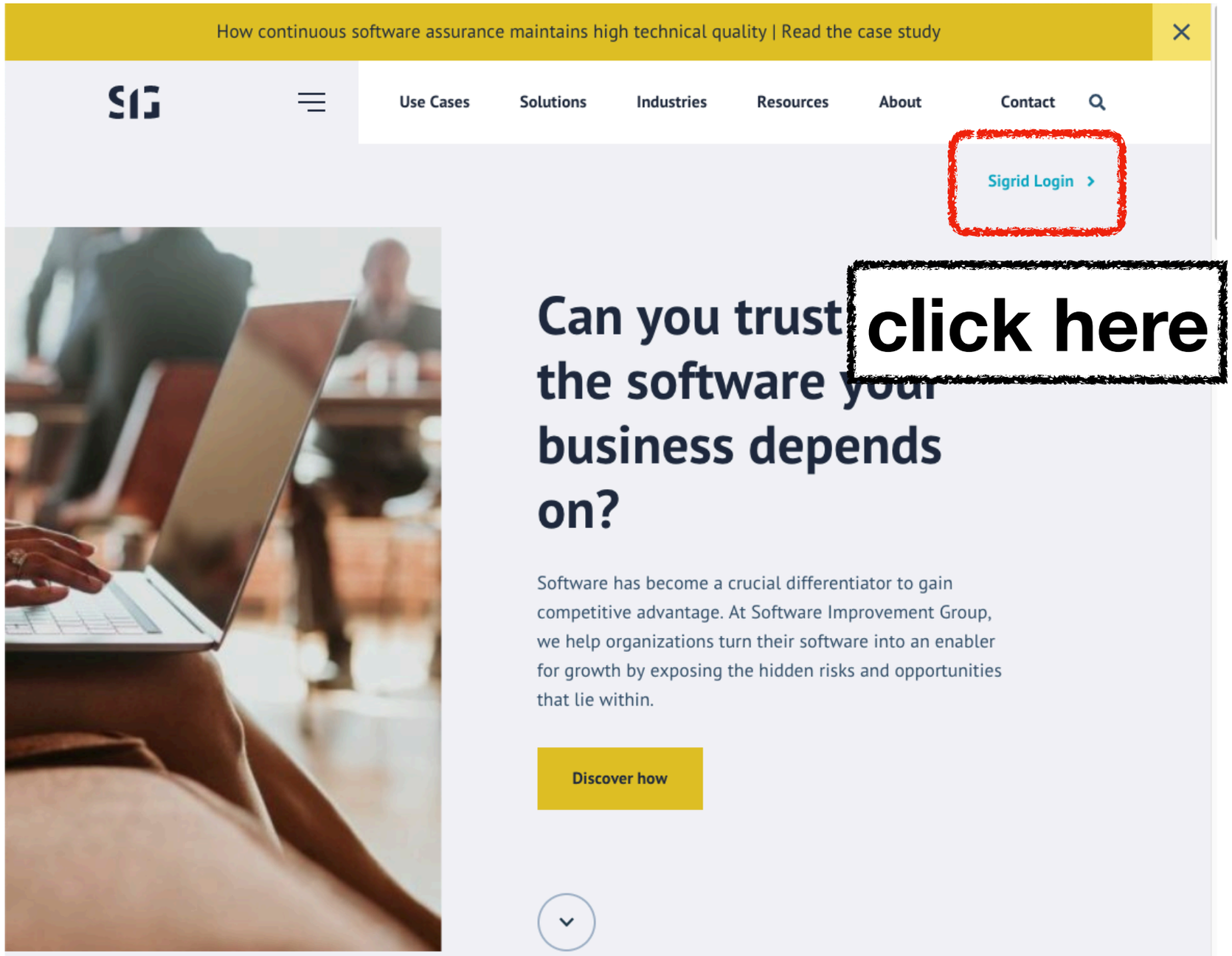
Benchmark is made of 100s of systems

Alves, et. al., Benchmark-based Aggregation of Metrics to Ratings, IWSM / Mensura 2011

Questions? 

Access the SIG tooling for your project

www.softwareimprovementgroup.com



Componentization

- Thanks all for the components!
- Some projects did an amazing job at defining them, and thus I reused them as is
- Some other projects I redefined the components myself. If that’s the case for you, and you want to redefine them, let me know (mattermost or email)

Disclaimer

- **Results of the analysis are preliminary** and are not fully representative of SIG quality of a Software Product

- The Sigrid portal results and data are given to you as-is. Please use them with grain of salt when reporting on them

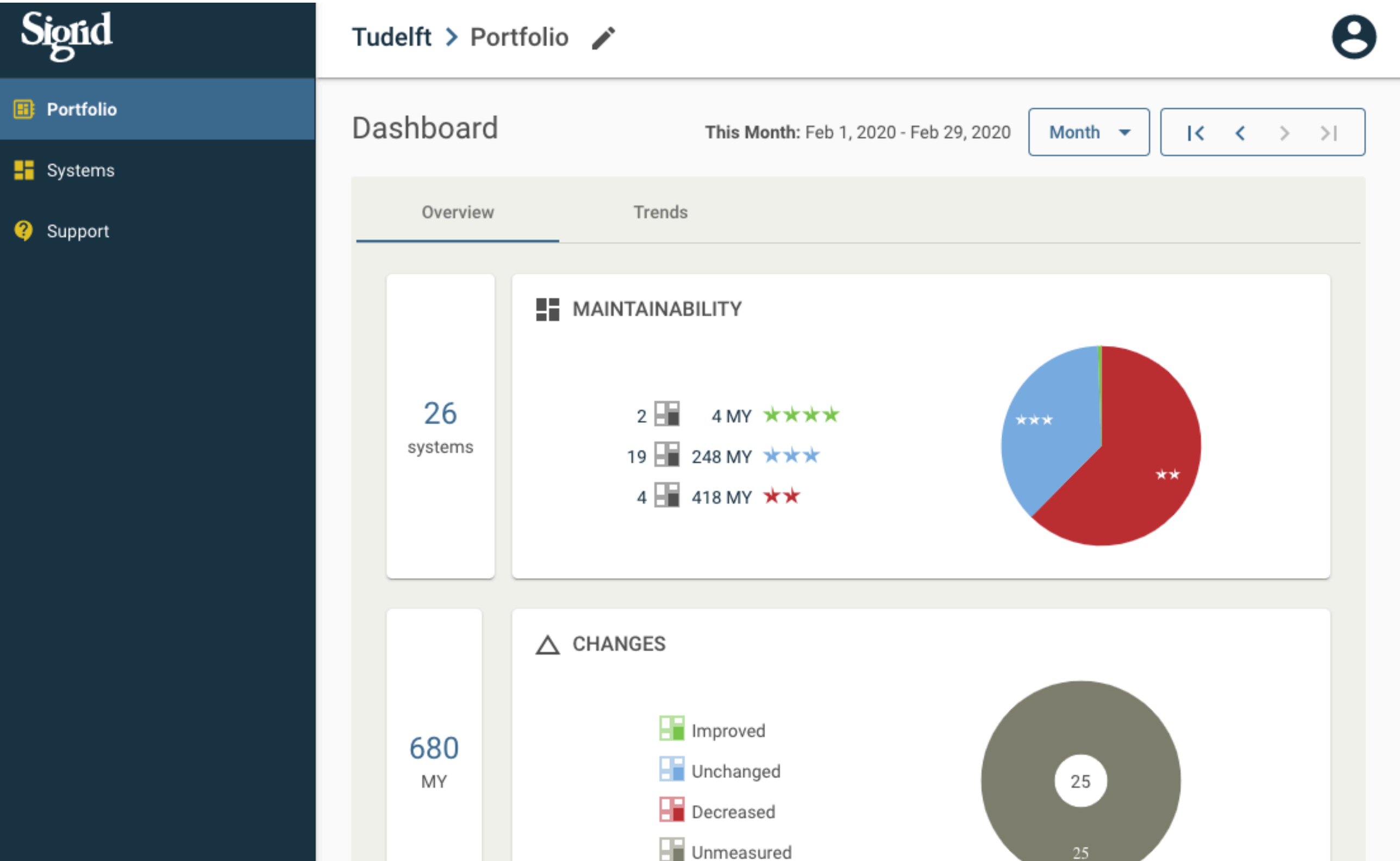
- Results might contain false positive, so be wary of strange results or inconsistent finding

- Do not use screenshots from Sigrid, the Technical Monitor or the rating as-is to report on any aspect of the system

Responsible use of the metrics is just as important as collecting them in the first place

https://blog.codinghorror.com/a-visit-from-the-metrics-maid/

Demo Time


Sigrid
Portfolio
Systems
Support
Tudelft > Portfolio
Dashboard
This Month: Feb 1, 2020 - Feb 29, 2020
Month
Overview
Trends
26
systems
MAINTAINABILITY
2 4 MY
19 248 MY
4 418 MY
680
MY
CHANGES
Improved
Unchanged
Decreased
Unmeasured
25
25

What can you do with Sigrid?

- Explore your projects and leverage information that might be not known to your project maintainers (PS: check out also other teams' project!)

- Check the list of the refactoring candidates; open PR and solve some of your project issues by proposing refactorings

- Tackle low-hanging fruits: if there's an issue open in your project that you try to solve, also leave the code better than you found it

- For the various assignments: use the GQM and formulate meaningful questions that you want to answer, and try to answer them using SIG data

Questions? 

Marco di Biase

@mardibiase

m.dibiase@sig.eu

m.dibiase@tudelft.nl