

WORKING WITH **LEGACY** CODE

Gabor Olah
Developer @ Erlang Solutions



@olikasg
gabor.olah@erlang-solutions.com

Erlang
SOLUTIONS

1.

WHAT IS LEGACY CODE?



“

legacy

adjective | 'lɛgəsi |

denoting or relating to software or hardware that has been superseded but is difficult to replace because of its wide use.

Source: Oxford English Dictionary

“

legacy

adjective | 'lɛgəsi |

denoting or relating to software or hardware that has been **superseded** but is difficult to replace because of its wide use.

Source: Oxford English Dictionary

“

legacy

adjective | 'lɛgəsi |

denoting or relating to software or hardware that has been superseded but is **difficult to replace** because of its wide use.

Source: Oxford English Dictionary

“

legacy

adjective | 'lɛgəsi |

denoting or relating to software or hardware that has been superseded but is difficult to replace because of its **wide use**.

Source: Oxford English Dictionary

WHAT'S MISSING?

Full information:

- ▶ Mindset
- ▶ Context
- ▶ Assumptions

MY DEFINITION

- ▶ Older than $\frac{1}{2}$ year
- OR**
- ▶ More than 5000 lines
- OR**
- ▶ Developed by somebody else

Tests are orthogonal

Code We Do Not Understand



Bad Code

WHAT IS
LEGACY?



SPAGHETTI

OBSCURE

MESSY



TANGLED

AESTHETICS

noun |ɛs 'θɛɪɪks|

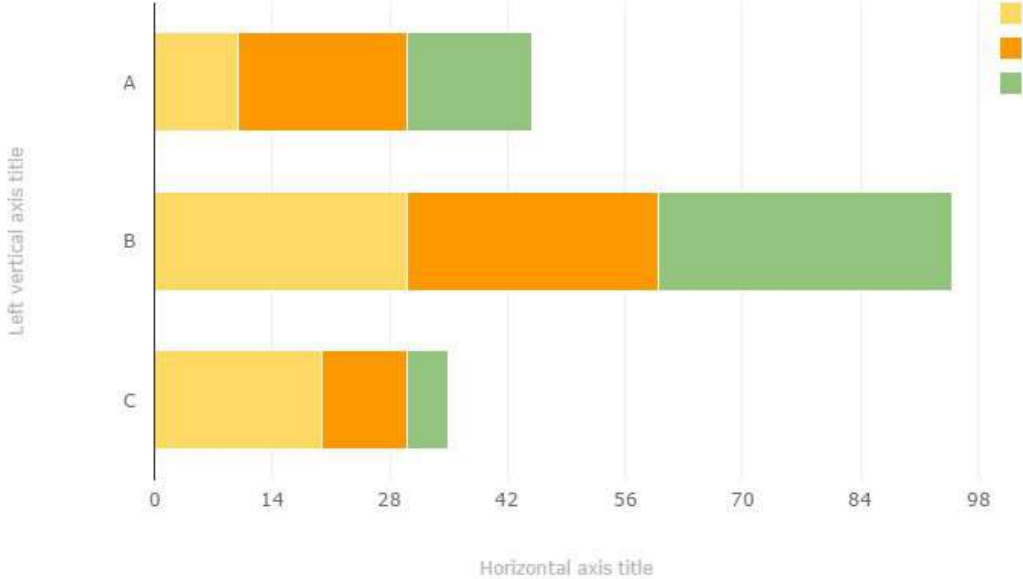
a set of principles concerned with the nature and appreciation of beauty.

the **branch of philosophy** which deals with questions of beauty and artistic taste.

Oxford English Dictionary



WHAT ABOUT SOFTWARE METRICS?



2.

BATTLE STORIES





1.

MODULE TOO LONG

MODULE TOO LONG

Story time ...

- ▶ Everything related in one place
- ▶ Too many requirements
- ▶ Some duplication

MODULE TOO LONG

For the next generation of programmers

- ▶ MINDSET : Refactoring with **good intention**
- ▶ CONTEXT : Creating a library module

No reasonable abstraction possible

- ▶ ASSUMPTION : **LESS** cognitive overhead



2.

NESTING TOO DEEP

NESTING TOO DEEP

```
case #r.f0 of
  A -> [{f1, V11}, {f2, V1}];
  B -> [{f1, V21}, {f2, V1}];
  C -> [{f1, V31}, {f2, V2}]
end
```

NESTING TOO DEEP

```
case #r.f0 of
  A -> case #r.orig_f0 of
    B -> [{f1, V11}, {f2, V1}];
    C -> [{f1, V12}, {f2, V3}]
  end;
  B -> [{f1, V21}, {f2, V1}];
  C -> [{f1, V31}, {f2, V2}]
end
```

NESTING TOO DEEP

```
case #r.f0 of
  ...
end
```

- ▶ Mapping values, no algorithm
- ▶ Too many requirements
- ▶ Some duplication

NESTING TOO DEEP

For the next generation of programmers

- ▶ MINDSET : Refactoring with **good intention**
- ▶ CONTEXT : Remove the nesting and duplication

Algorithm for brevity

- ▶ ASSUMPTION : Code will be **EASIER** to understand

3.

NO DEADLINE

NO DEADLINE

Story time ...

- ▶ “Change the timestamp in the logs”
- ▶ No pressure from project management
- ▶ Good software practices followed

NO DEADLINE

For the next generation of programmers

- ▶ MINDSET : If it changes once, it will change later...
- ▶ CONTEXT : Customisable **feature**

Developer wants to learn

- ▶ ASSUMPTION : Good code **IMPROVES** the software



3.

MINDSET



BE HUMBLE



**WORK
TOGETHER**

BE PATIENT



**BE
PROFESSIONAL**





4.

TOOLS TO USE

Understand



Verify and Refine



Modify

THE PROCESS

WHAT TO UNDERSTAND

- ▶ Requirements
- ▶ Application structure
- ▶ Static structure
- ▶ Dynamic structure
- ▶ Data model



GO ANALOG

HOW TO UNDERSTAND

- ▶ RTM
- ▶ Read the code
- ▶ Draw interactions
- ▶ Write down questions
- ▶ Ask

LOGS

- ▶ Use it to verify your understanding
- ▶ Read it
- ▶ What's not there?
- ▶ Timings
- ▶ Timelines
- ▶ Uniqueness of entries

TRACING

Look into the clockworks of the VM.

Erlang in Anger by Fred Herbert

<https://www.erlang-in-anger.com>

Tutorial at **Code Elixir London**

<https://codesync.global/conferences/code-elixir-ldn-2019/tutorial/>

WHEN TO TRACE?

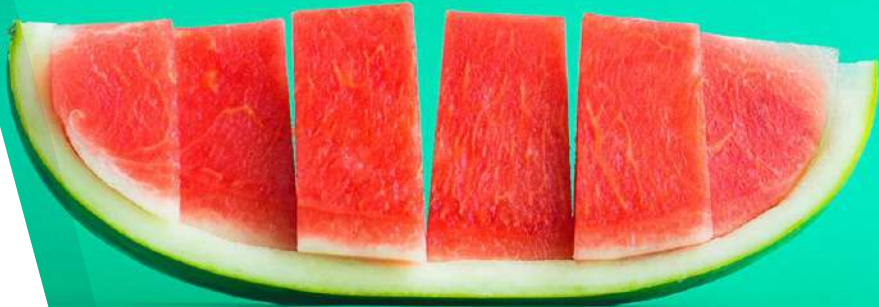
- ▶ You have access to the running system
- ▶ You have a theory to validate
- ▶ (Or you have no idea)

MODIFY
CAREFULLY



5.

SUMMARY



WORKING WITH LEGACY CODE

- ▶ Realise the difference bad code and code you don't understand
- ▶ Be humble and professional
- ▶ Understand before you modify
- ▶ Use all tools available

