Digital Dragons

KRAKOWSKI PARK TECHNOLOGICZNY

How to not default on technical debt?

Szymon Sirocki

Szymon Sirocki

friends call me **Simon**

11 years in game dev

Game, Tools, Back-end, IT, Architecture

Developer, Leader, Architect, ex-CTO

code-for-a-living FEBRUARY 27, 2023

Stop saying “technical debt”

Everyone who says "tech debt" assumes they know what we’re all talking about, but their individual pictures differ quite a bit.

Chelsea Troy

We were supposed to release this feature three weeks ago.

One developer got caught in a framework update. Another got stuck reorganizing the feature flags. A third needed to

(...) technical debt (...) is the implied **cost of additional rework** caused by choosing an easy (limited) solution now instead of using a better approach that would take longer.

https://en.wikipedia.org/wiki/Technical_debt

All code is technical debt - some code just has a higher interest rate.

Paul McMahon (tokyodev.com)


ALWAYS HAS BEEN
IS IT ALL LEGACY CODE?

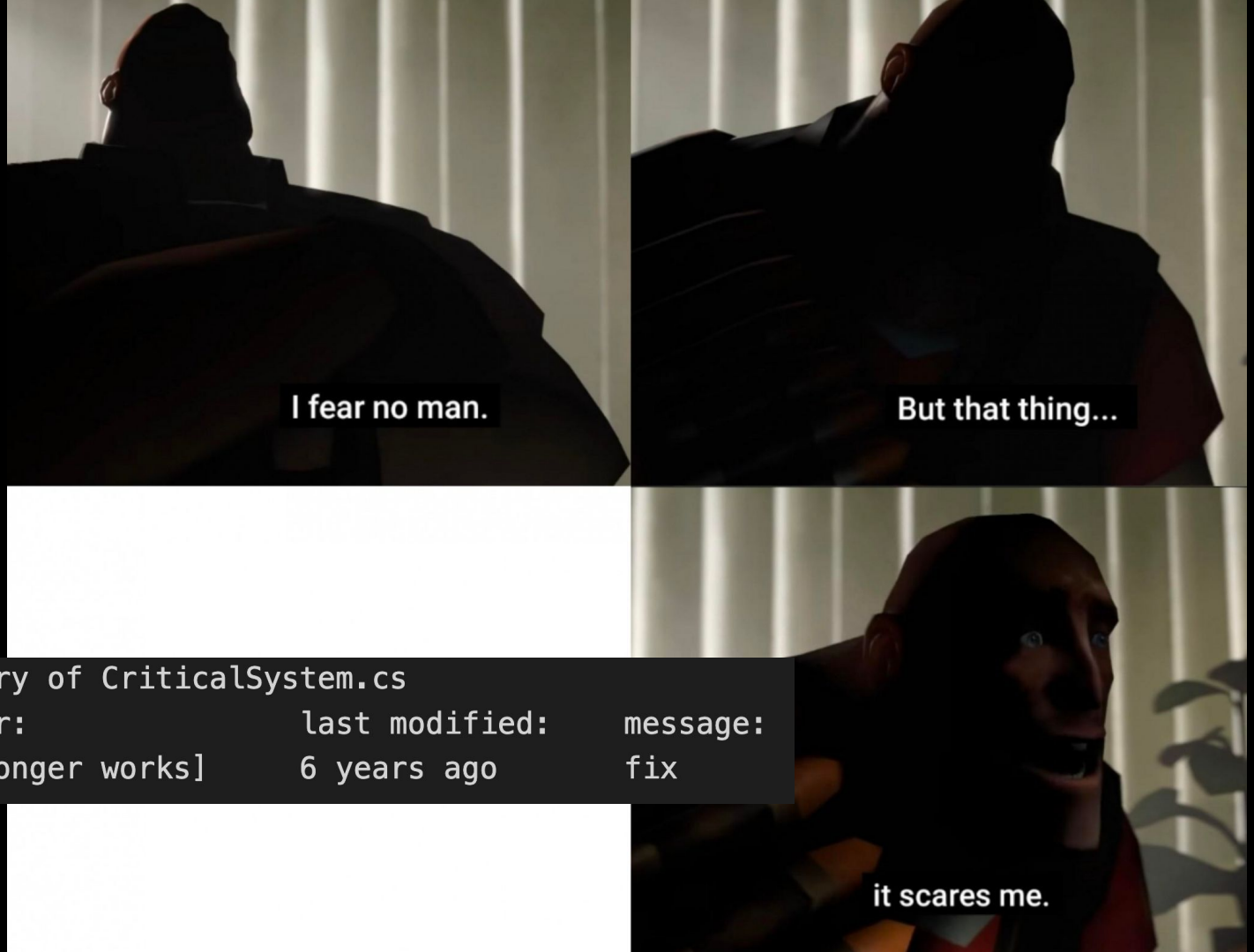
Technical debt is a property of a project that **degrades the ability to develop** it further.

How does technical debt manifest?

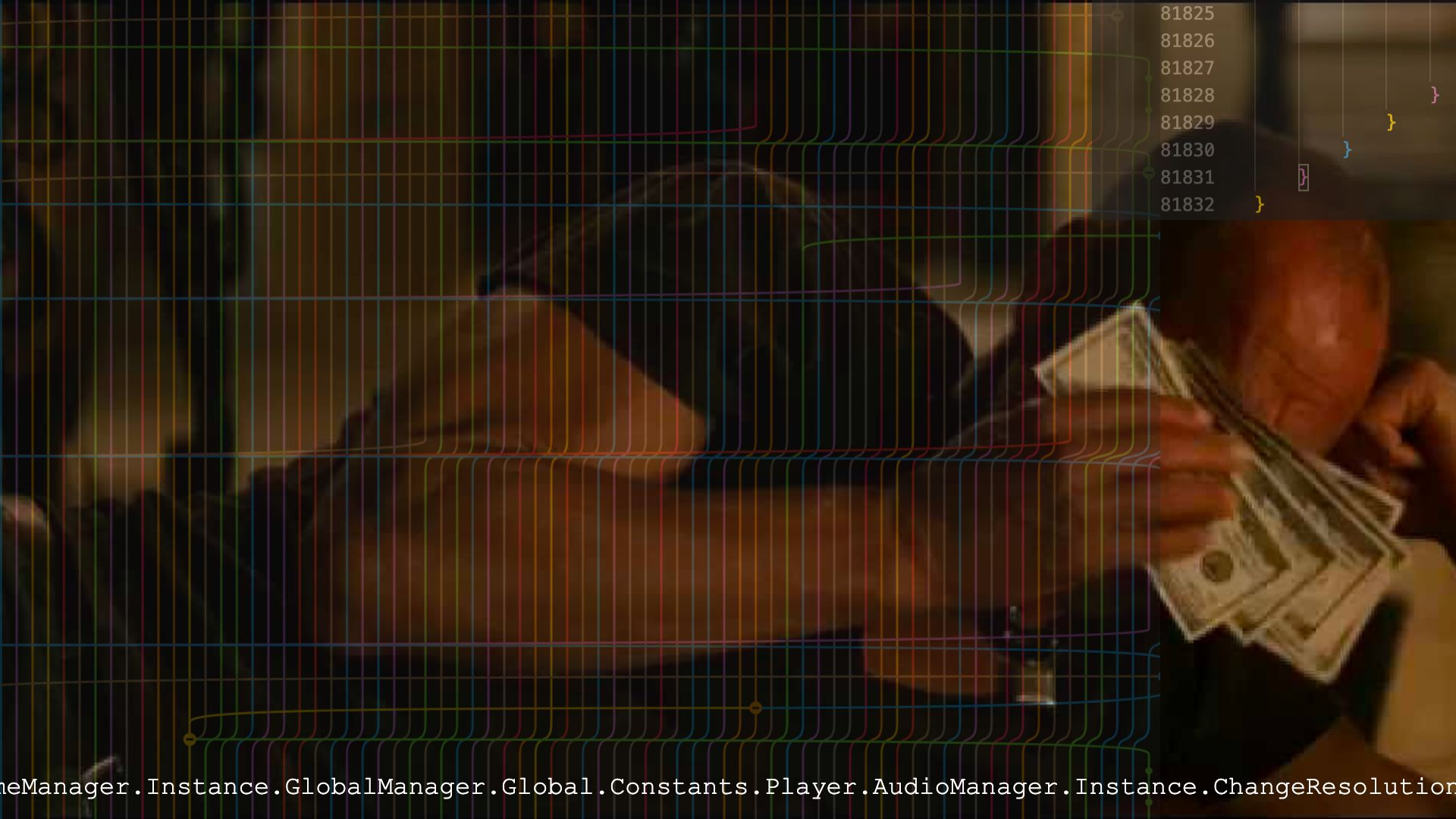
We often start thinking about debt when the first consequences appear

How to check if your project has technical debt?

- Features take more time to develop?
- Developers say: *“This needs a rewrite”* or *“The architecture doesn’t support it”* ?
- Time lost on “refactoring” instead of moving forward?
- People lose interest, motivation and leave the project?
- Hate your ~~past~~ decisions?
- Dread to come back to work on the feature?
- Debugging is a nightmare?
- Hard to onboard new members?


I fear no man.
But that thing...
History of CriticalSystem.cs
author: last modified: message:
[no longer works] 6 years ago fix
it scares me.

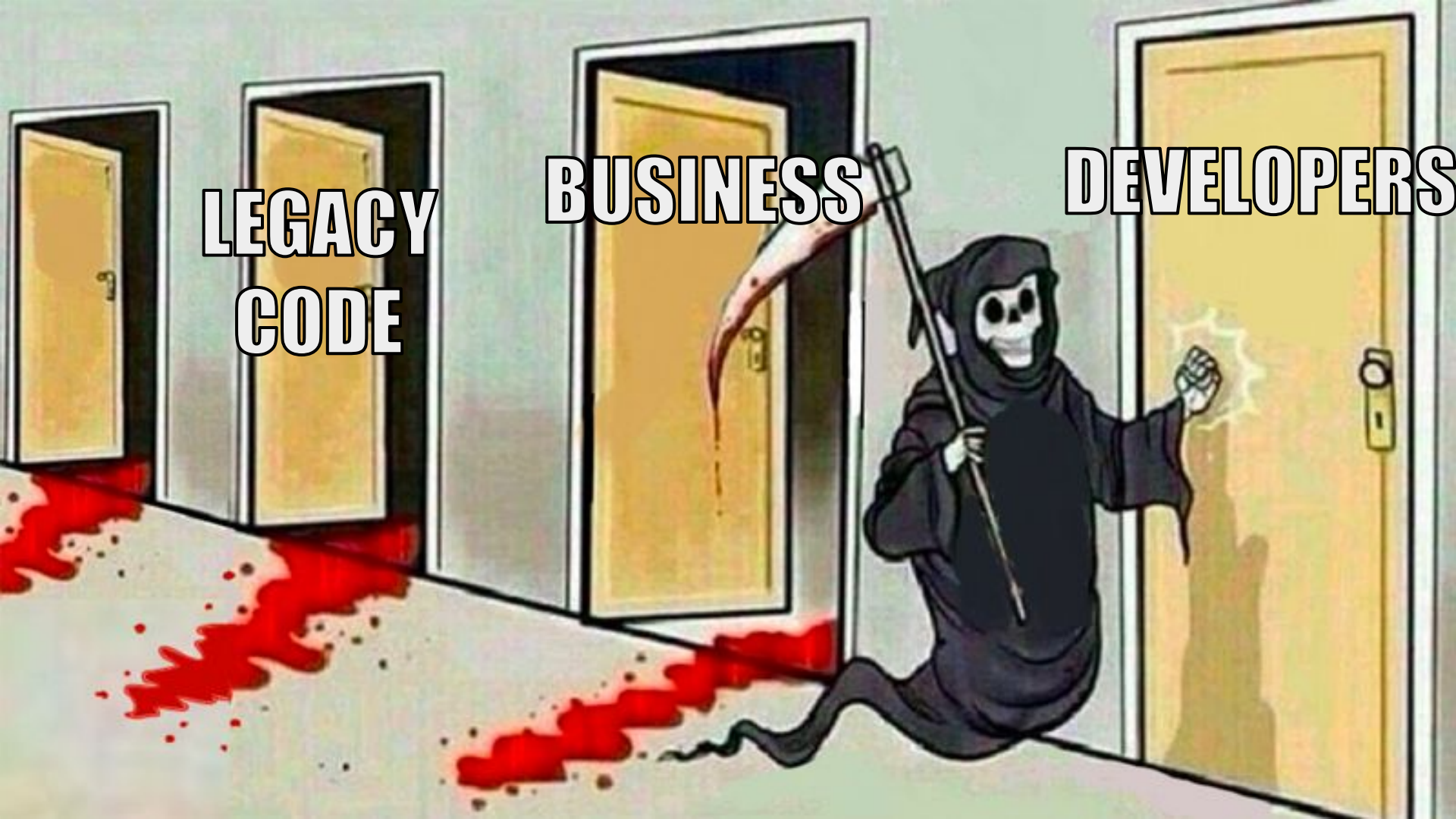
Okay, but how did it get there?


81825
81826
81827
81828
81829
81830
81831
81832
}
}
}
}
}
meManager.Instance.GlobalManager.Global.Constants.Player.AudioManager.Instance.ChangeResolution

Why is the games industry so burdened with crunch? It starts with labor laws.

By Michael Thomsen
March 24, 2021 at 3:09 p.m. EDT

(The Washington Post illustration; iStock)


LEGACY CODE
BUSINESS
DEVELOPERS

No architecture

~~No~~ bad architecture

Architecture is the stuff that's hard to change

Mark RIchards & Neal Ford
Fundamentals of Software Architecture: An Engineering Approach

No guidelines

(...) ritual inclusion of code or (...) structures that serve no real purpose

Wikipedia - Cargo-cult programming

No code review

Poor code quality

Voodoo programming

And once we know we have Technical debt

… it only gets worse


DEV
TECHNICAL DEBT
PATTERNS

It is a mistake to think you can solve any major problem just with potatoes.

Douglas Adams
Life, Universe, and Everything.

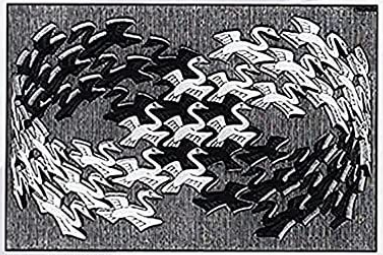
It is a mistake to think you can solve any major problem just with ~~potatoes~~ patterns.

Douglas Adams
Life, Universe, and Everything.

Design Patterns

Elements of Reusable Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides



Foreword by Grady Booch

ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Building the code you **might** need

But not necessarily right now

Complexity

Developers are drawn to complexity like moths to a flame, often with the same outcome.

Neal Ford

The bigger something is the harder it is to get rid of.


OK
SO LIKE, WHAT DO I DO NOW?

Guidelines

- Naming conventions
- Language use
- Style
- Structure

SCA

Static Code Analysis

- In IDE
- External tools
- Roslyn analyzers

sonarqube

CodeScene

Don’t wait

Prioritize early

- Make effort to identify issues
- Allocate improvements every sprint
- Think about debt while writing code

Gradual improvement
=
Prevention of decay

Challenge yourself

- Ask multiple ‘why’s
- **Really** identify issues.
- Discuss with yourself
- Find others to challenge you

Take small steps

Don’t over-engineer

Build what’s needed now

- Abstract only when necessary
- Don’t build code not not backed in business requirements
- Focus on what is needed now in the scope of a greater whole

Thank you

simon#7000

codenamesimon

szymonsirocki