

Chad Green

Director of Architecture
Glennis Solutions



Agenda

- What is Technical Debt
- Pros and Cons of Technical Debt
- What Causes Technical Debt
- Types of Technical Debt
- Technical Debt Identification
- Tracking Technical Debt



What is Technical Debt?



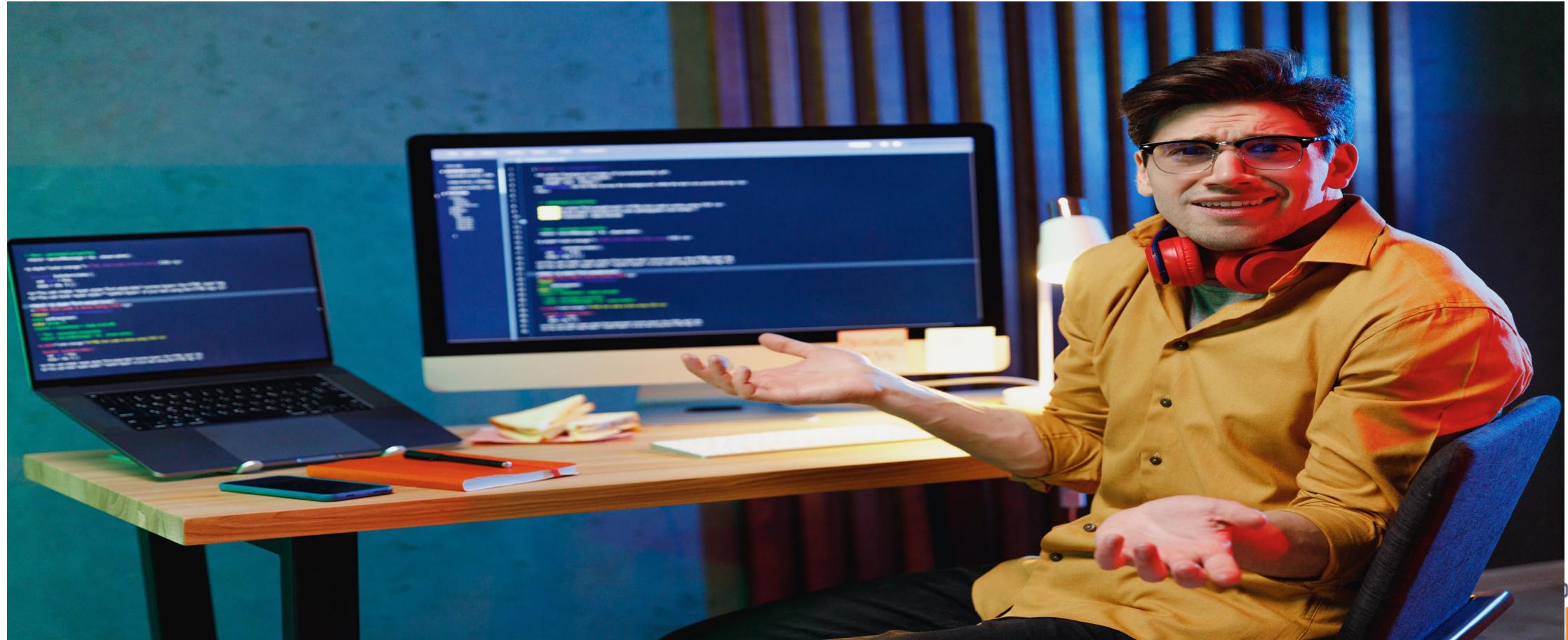
What is Technical Debt?



What is Technical Debt?



What is Technical Debt?



What is Technical Debt?



What is Technical Debt?



What is Technical Debt?



What is Technical Debt?



What is Technical Debt?



What is Technical Debt?

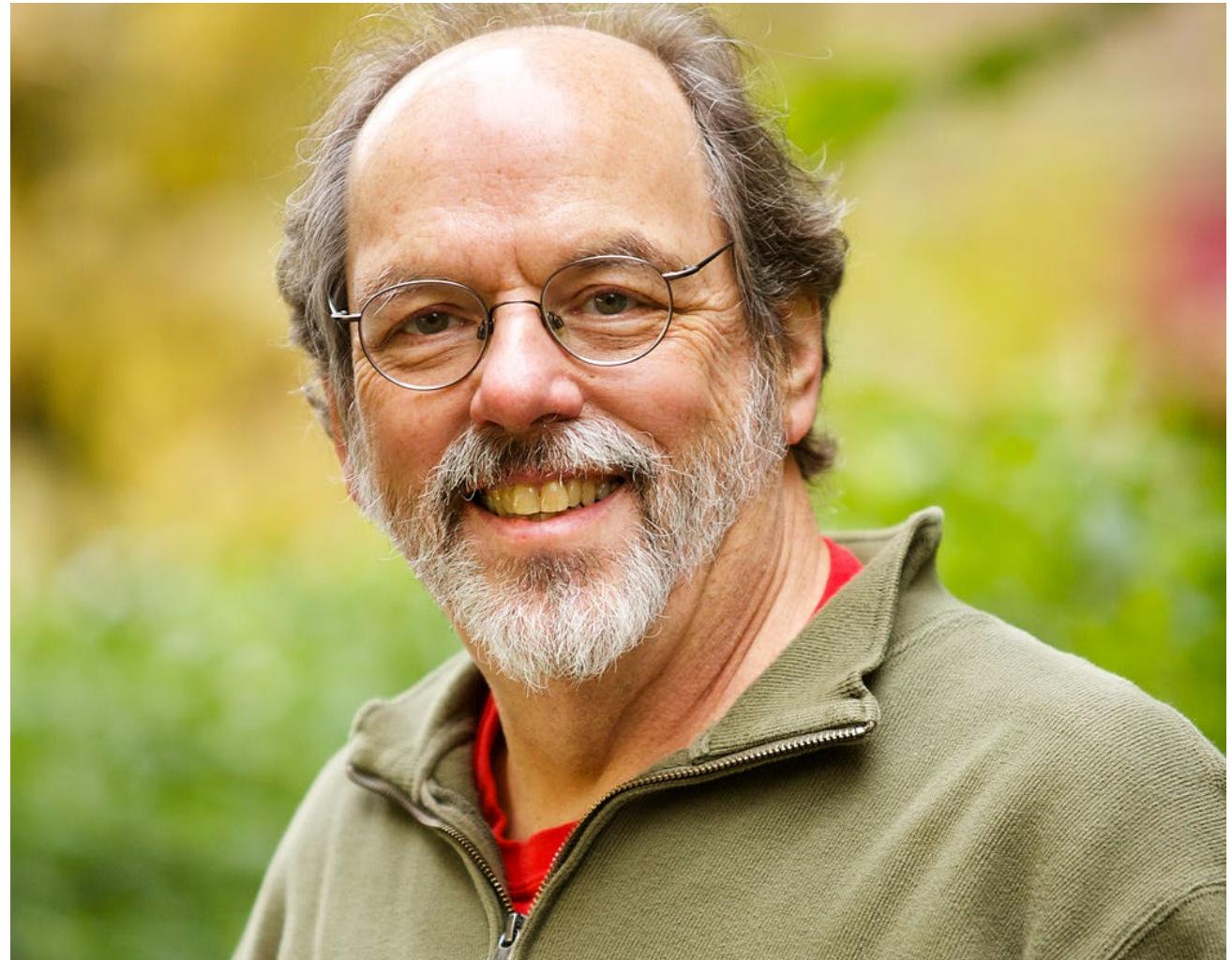


What is Technical Debt?



What is Technical Debt?

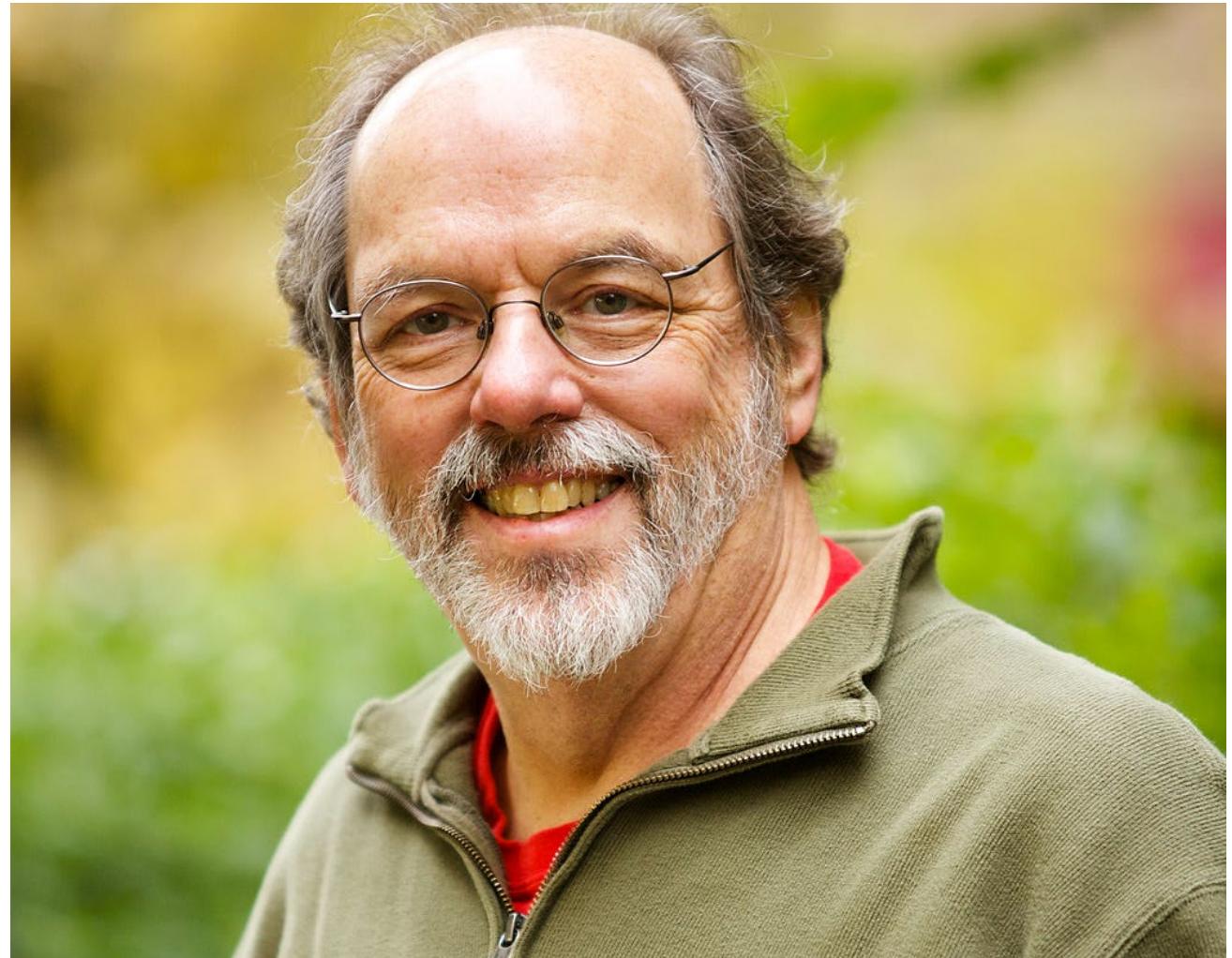
First described by Ward Cunningham
in 1992



What is Technical Debt?

First described by Ward Cunningham
in 1992

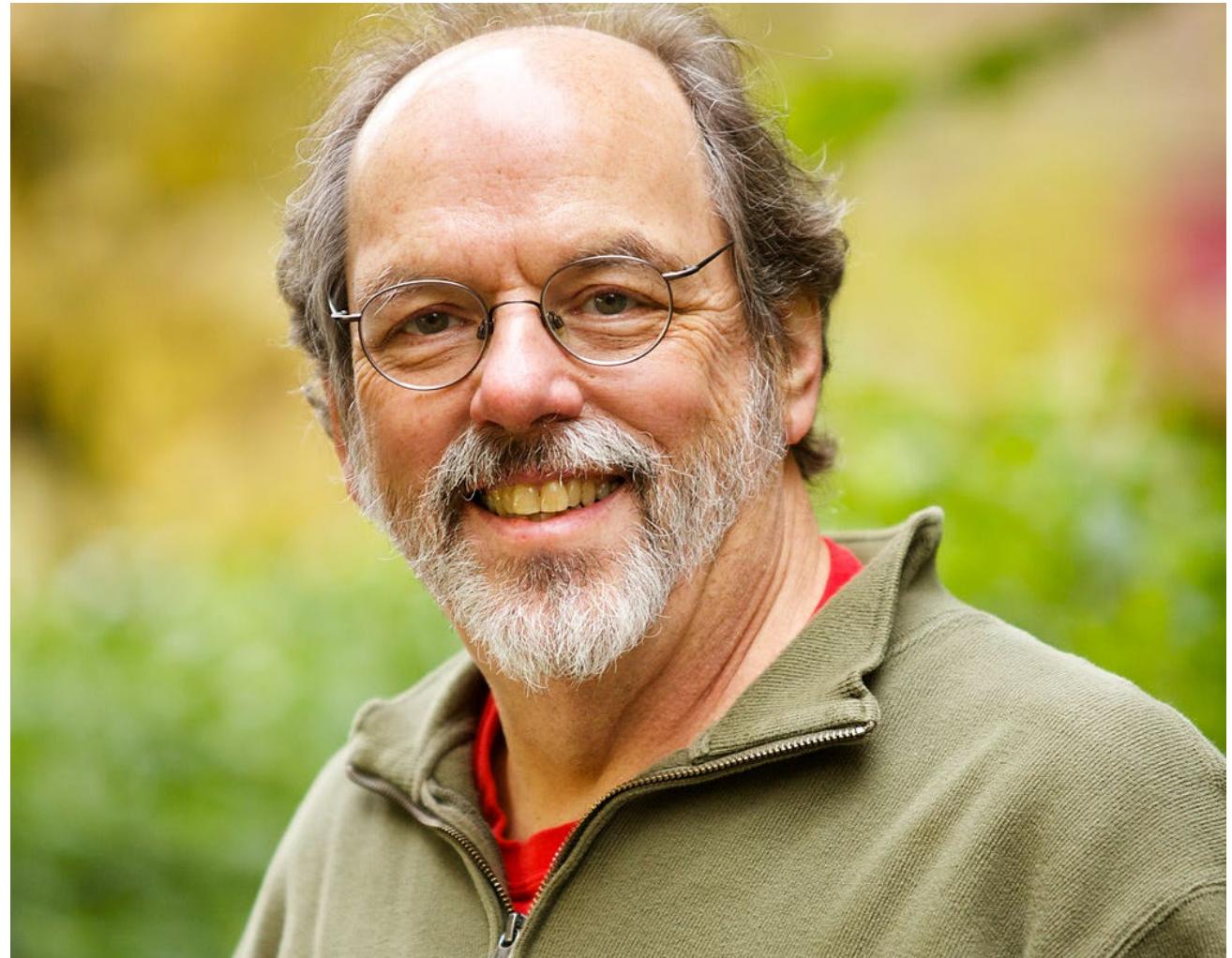
The danger occurs when
the debt is not repaid



What is Technical Debt?

First described by Ward Cunningham
in 1992

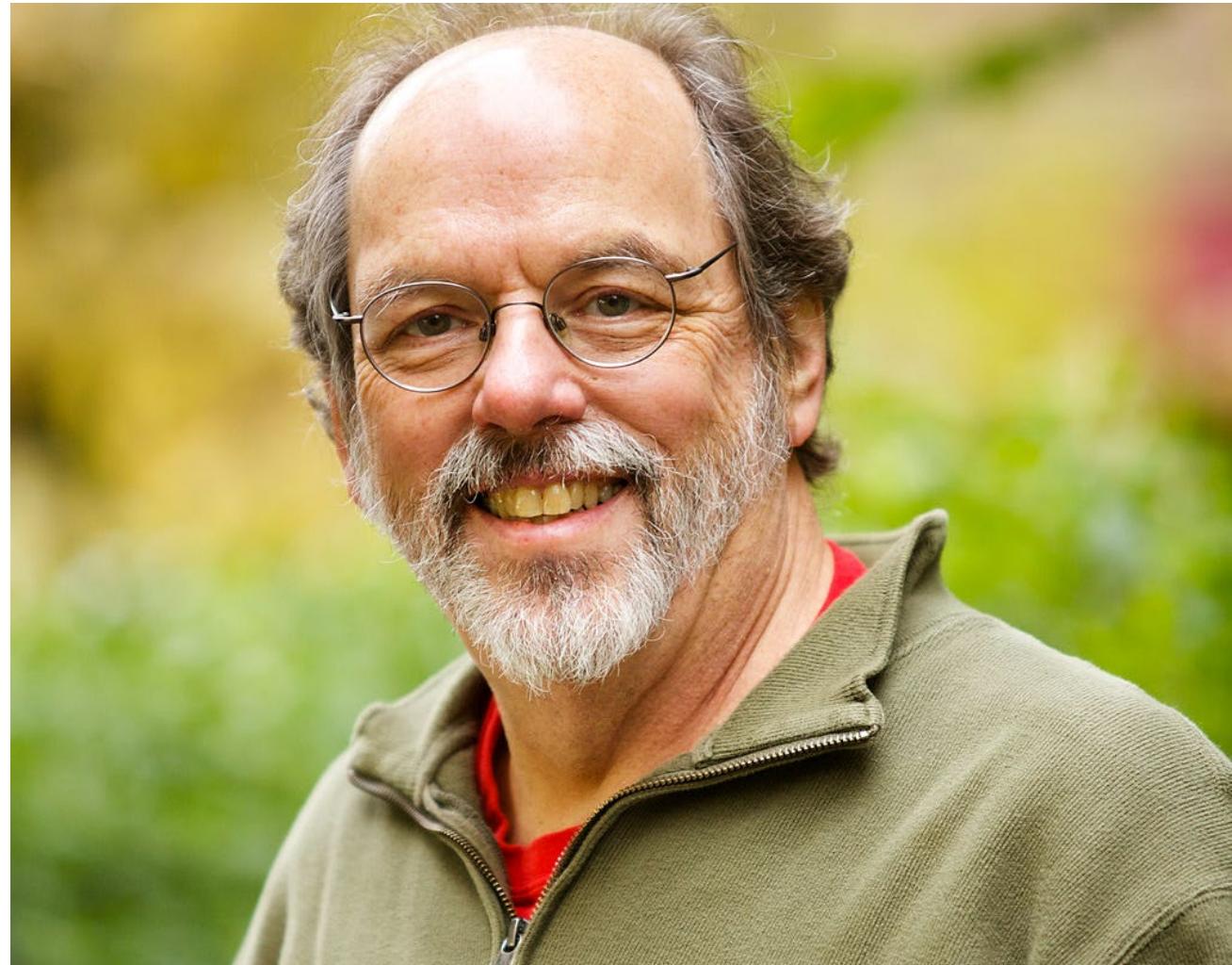
Every minute spent on
not-quite-right code
counts as interest on that
debt.



What is Technical Debt?

First described by Ward Cunningham
in 1992

...brought to a stand-still
under the debt load of an
unconsolidated
implementation



What is Technical Debt?

Common definition comes from
Martin Fowler



What is Technical Debt?

Common definition comes from
Martin Fowler

quick and dirty way



What is Technical Debt?

Common definition comes from
Martin Fowler

incurs interest payments



What is Technical Debt?

Common definition comes from
Martin Fowler

extra effort that we have
to do in future
development



What is Technical Debt?

More academic definition

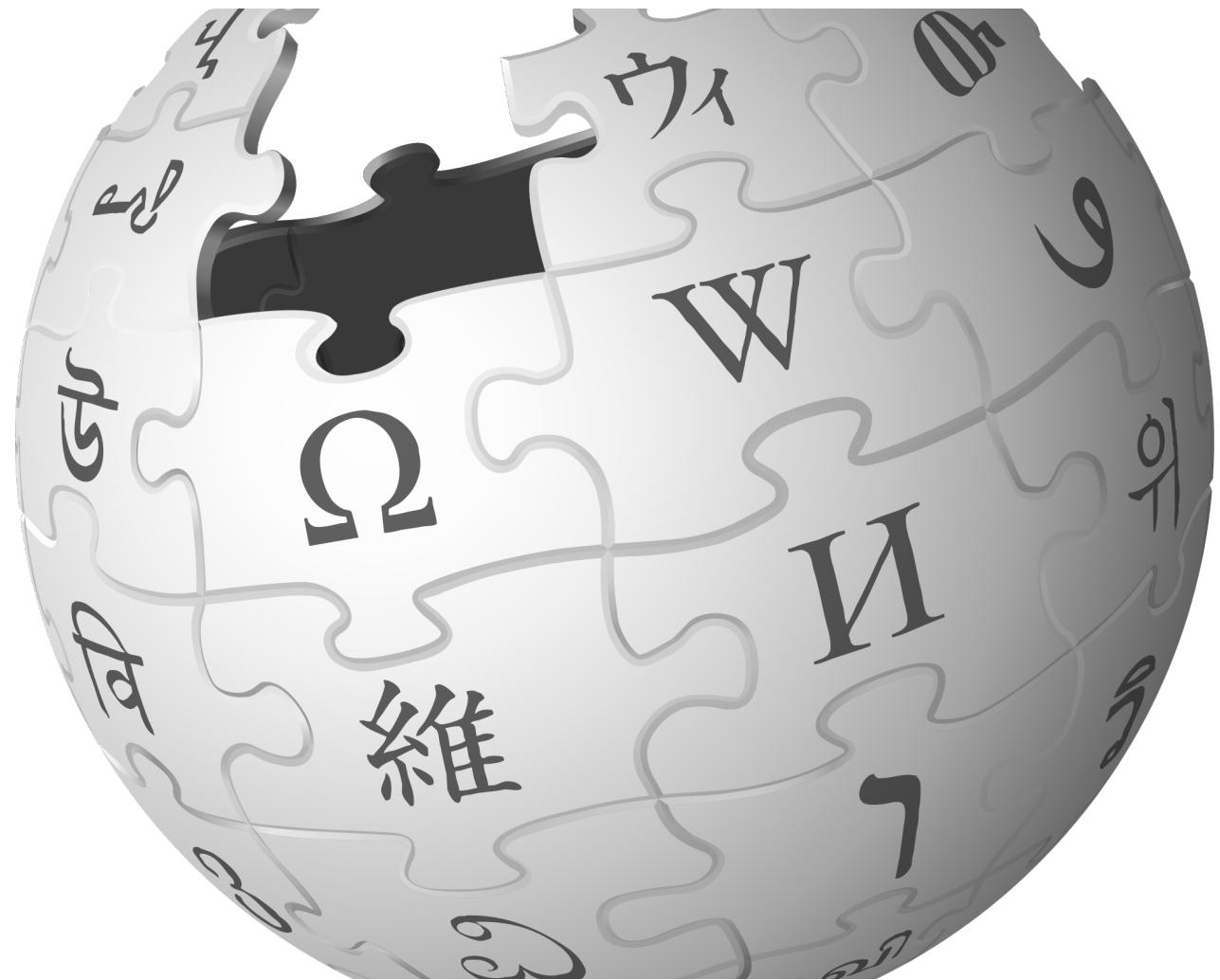
implied cost



What is Technical Debt?

More academic definition

rework



What is Technical Debt?

More academic definition

choosing easy



What is Technical Debt?

More academic definition

better approach



A more reasonable definition

A state of unnecessary complexity, duplication, subpar code, or otherwise deficient code that introduces risks in the form of time, quality, security, team morale, or other forms. Typically the results of past action or inaction.

Matt Eland
killalldefects.com/2019/12/23/defining-technical-debt



Technical Debt is Bad



AGILE SOFTWARE TESTING & DEVOPS CONFERENCE

Technical Debt is Bad



AGILE SOFTWARE TESTING & DEVOPS CONFERENCE

Technical Debt can be Bad



Technical Debt can be Devastating



Technical Debt can be Useful



Technical Debt can be Beneficial



Technical Debt is not Free



Technical Debt Example

No User Roles

Permission for
Specific
Requirement

Differentiation of
Users

Yet Another
Permission
Change

Another Technical Debt Example

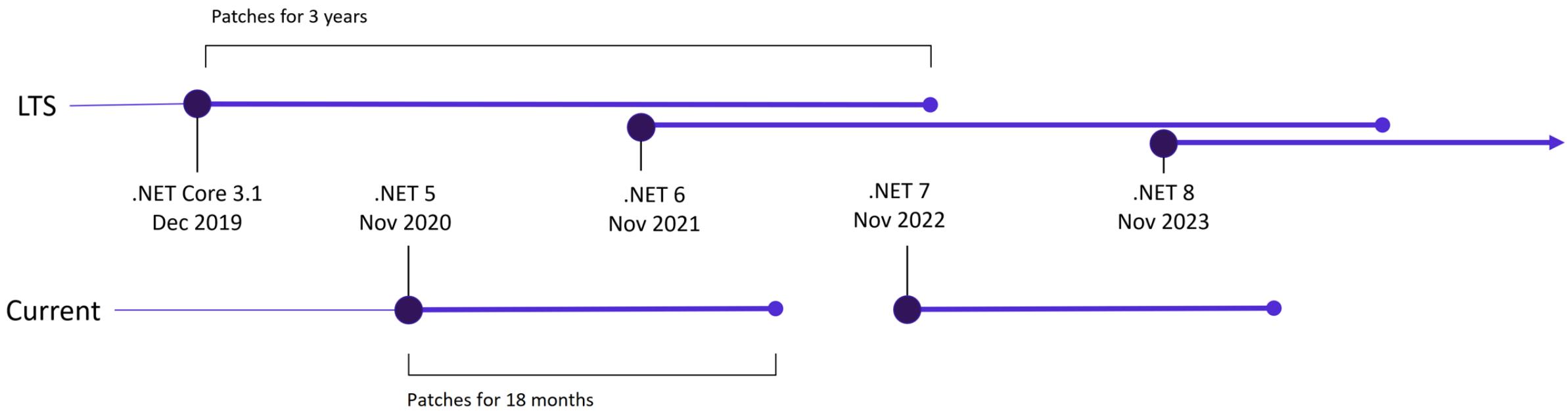
.NET 5

Current Release

.NET 6

LTS Release

Another Technical Debt Example



Pros and Cons of Technical Debt

Technical Debt Pros

Supplier

- Faster time to market
- Short term competitive advantage

Customer

- New features faster
- Short term capital preservation

Technical Debt Cons

Supplier

- Degradation of quality
- Limits ability to innovate
- Longer development cycles
- Less responsive to market changes
- Customer churn
- Developer churn

Customer

- Higher upgrade costs
- System compatibility issues
- Poor end user experience
- Less responsive to market changes
- Poor IT morale

What causes technical debt?

Insufficient up-front definition

What causes technical debt?



Business Pressures

What causes technical debt?



Lack of process or understanding

What causes technical debt?



Tightly-coupled components

What causes technical debt?



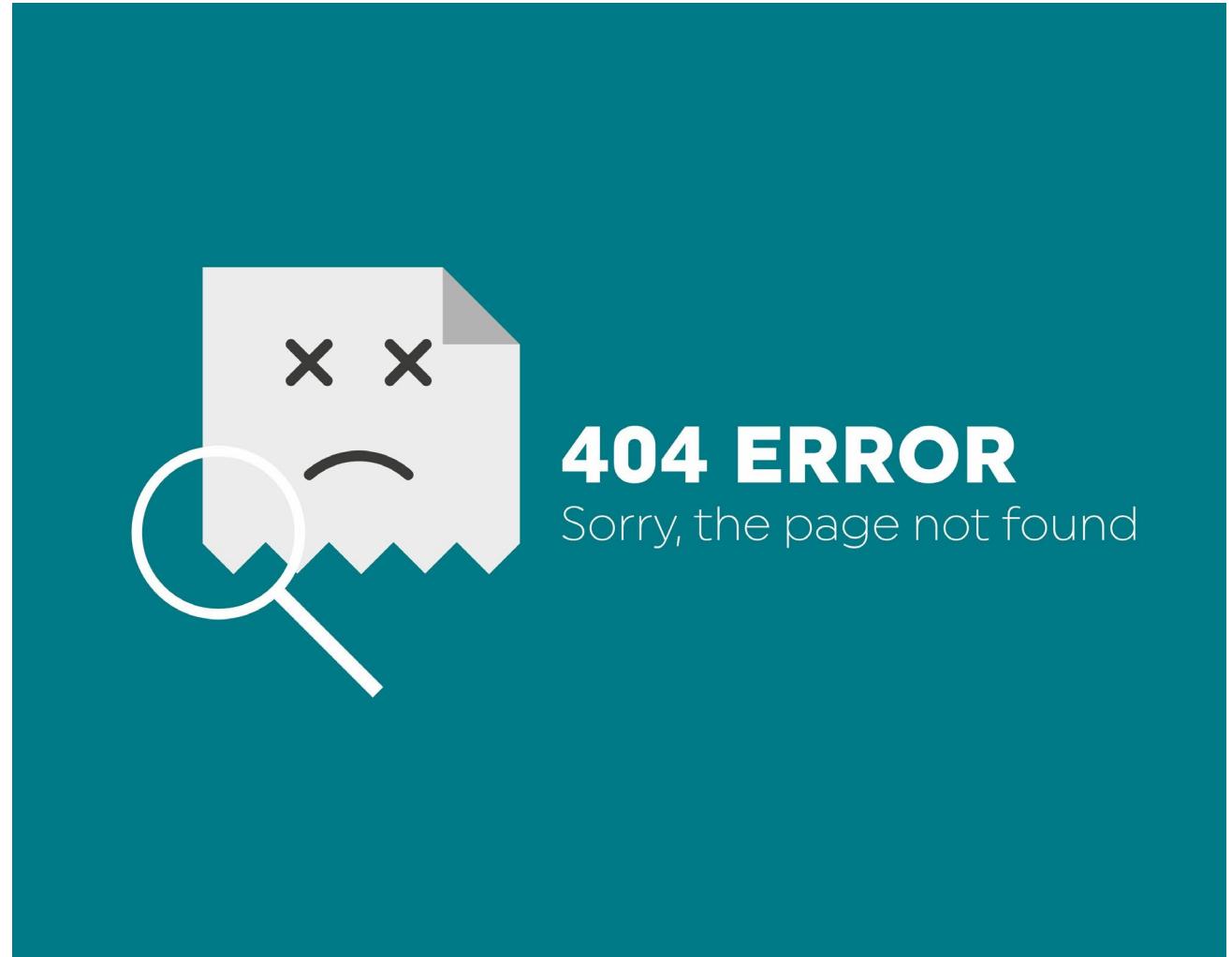
Lack of test suite

What causes technical debt?



Lack of documentation

What causes technical debt?



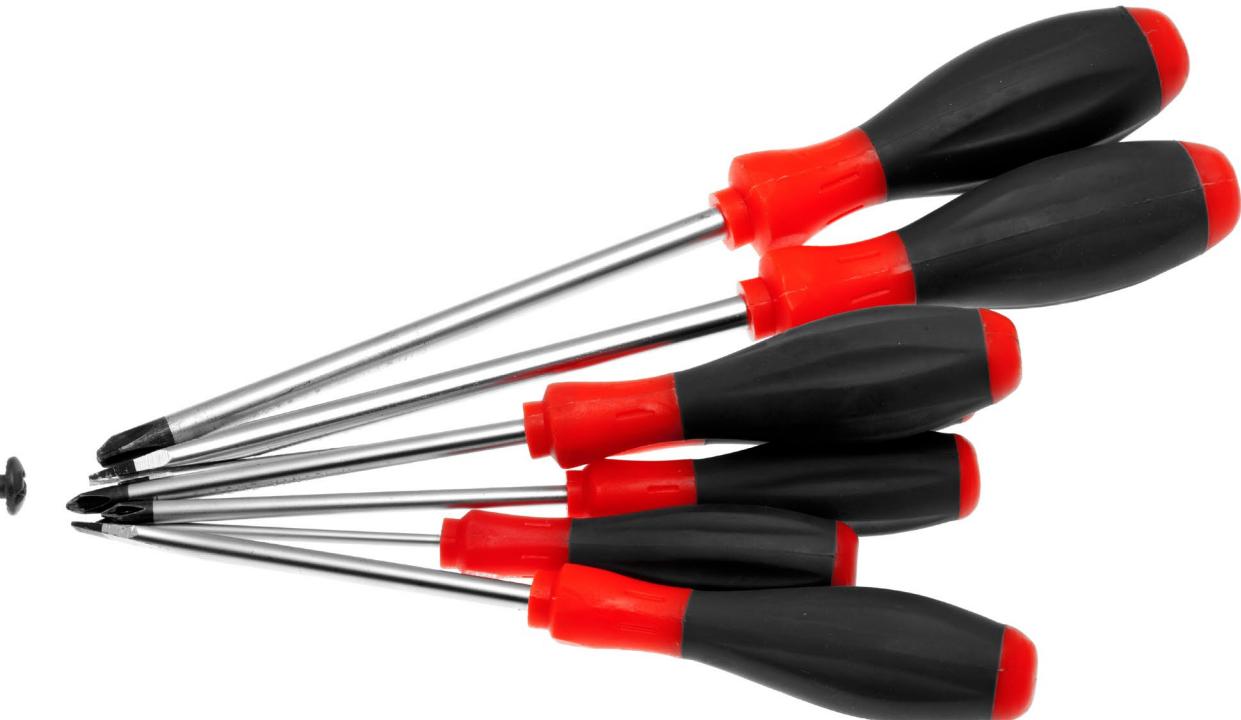
Lack of collaboration

What causes technical debt?



Parallel development

What causes technical debt?



Delayed refactoring

What causes technical debt?



Lack of alignments to standards

What causes technical debt?



Lack of knowledge

What causes technical debt?



Lack of ownership

What causes technical debt?



Poor technical leadership

What causes technical debt?



Last minute specification changes

What causes technical dbt?



What causes technical debt?

- Insufficient up-front definition
- Business pressures
- Lack of process of understanding
- Tightly-coupled components
- Lack of test suite
- Lack of documentation
- Lack of collaboration
- Parallel development
- Delayed refactoring
- Lack of alignment to standards
- Lack of knowledge
- Lack of ownership
- Poor technical leadership
- Last minute specification

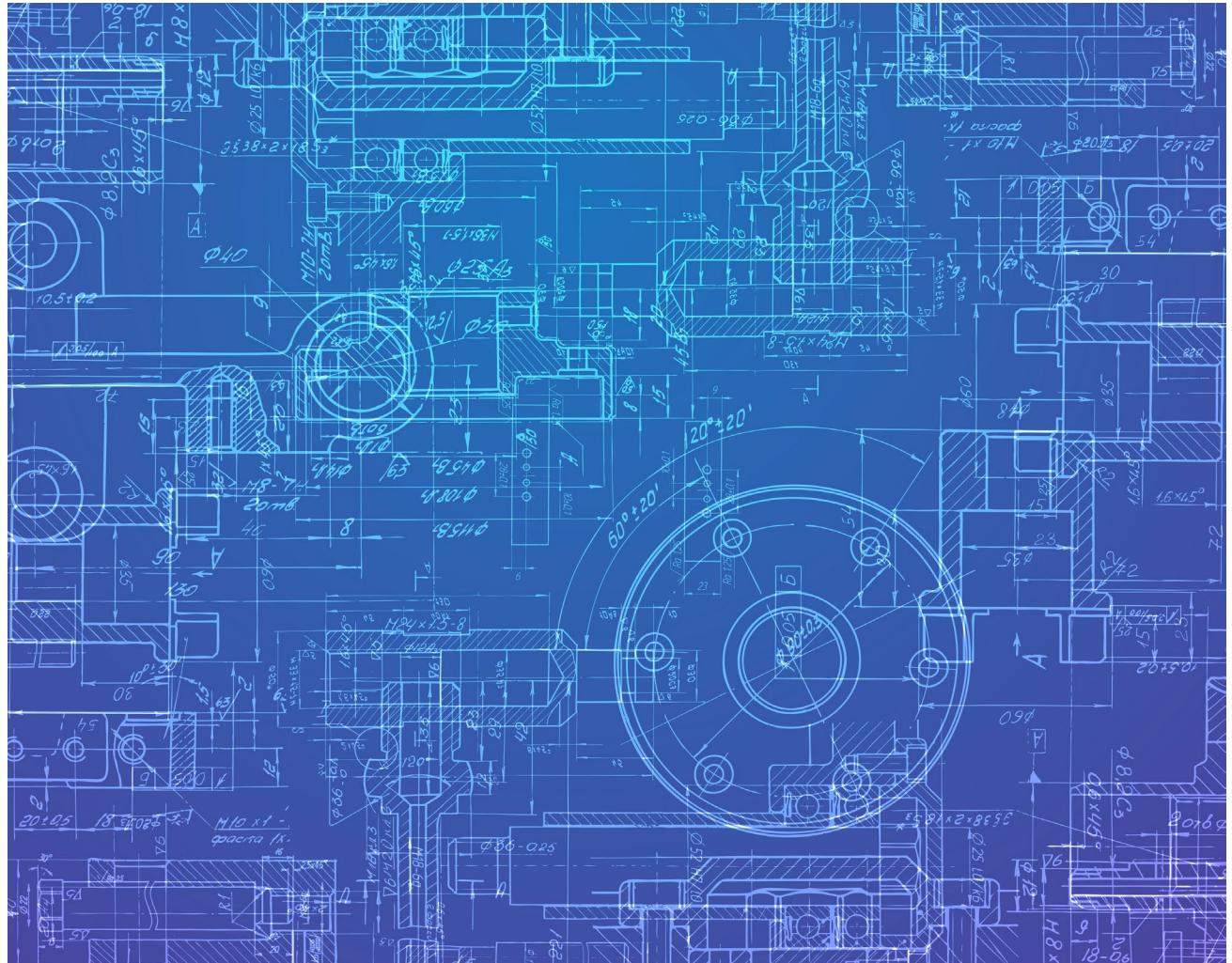
Types of Technical Debt

Architecture

Types of Technical Debt



AGILE SOFTWARE TESTING & DEVOPS CONFERENCE



Build

Types of Technical Debt



Code

Types of Technical Debt



Defect

Types of Technical Debt



Design

Types of Technical Debt



Documentation

Types of Technical Debt



Infrastructure

Types of Technical Debt



People

Types of Technical Debt



Process

Types of Technical Debt



Requirement

Types of Technical Debt



Service

Types of Technical Debt



Test Automation

Types of Technical Debt



Test

Types of Technical Debt



Types of Technical Debt

- Architecture
- Build
- Code
- Defect
- Design
- Documentation
- Infrastructure
- Process
- Requirement
- Service
- Test Automation
- Test

Technical Debt Identification

Technical Debt Identification

Code Quality

Clarity

Technical Debt Identification

Code Quality

Maintainability

Technical Debt Identification

Code Quality

Documentation

Technical Debt Identification

Code Quality

Refactoring

Technical Debt Identification

Code Quality

Well-Tested

Technical Debt Identification

Code Quality

Efficiency

Technical Debt Identification

Code Quality

Technical
Deficiencies

Have other teams built similar solutions?

Technical Debt Identification

Code Quality

Technical
Deficiencies

Are there inconsistent user experiences?

Technical Debt Identification

Code Quality

Technical
Deficiencies

Are solutions using the same
reusable frameworks?

Technical Debt Identification

Code Quality

Technical
Deficiencies

Are best practices being followed?

Technical Debt Identification

Code Quality

Technical
Deficiencies

Are deployments taking too long?

Technical Debt Identification

Code Quality

Technical
Deficiencies

Are limits being exceeded, or are
users seeing errors?

Technical Debt Identification

Code Quality

Technical
Deficiencies

Business
Misalignment

Opportunities to Identify Technical Debt

Coding

Opportunities to Identify Technical
Debt



Manifesto for Software Craftsmanship

Not only working software,
but also **well-crafted software**

Not only responding to change
but also **steadily adding value**

Not only individuals and interactions
but also **a community of professionals**

Not only customer collaboration
but also **productive partnerships**

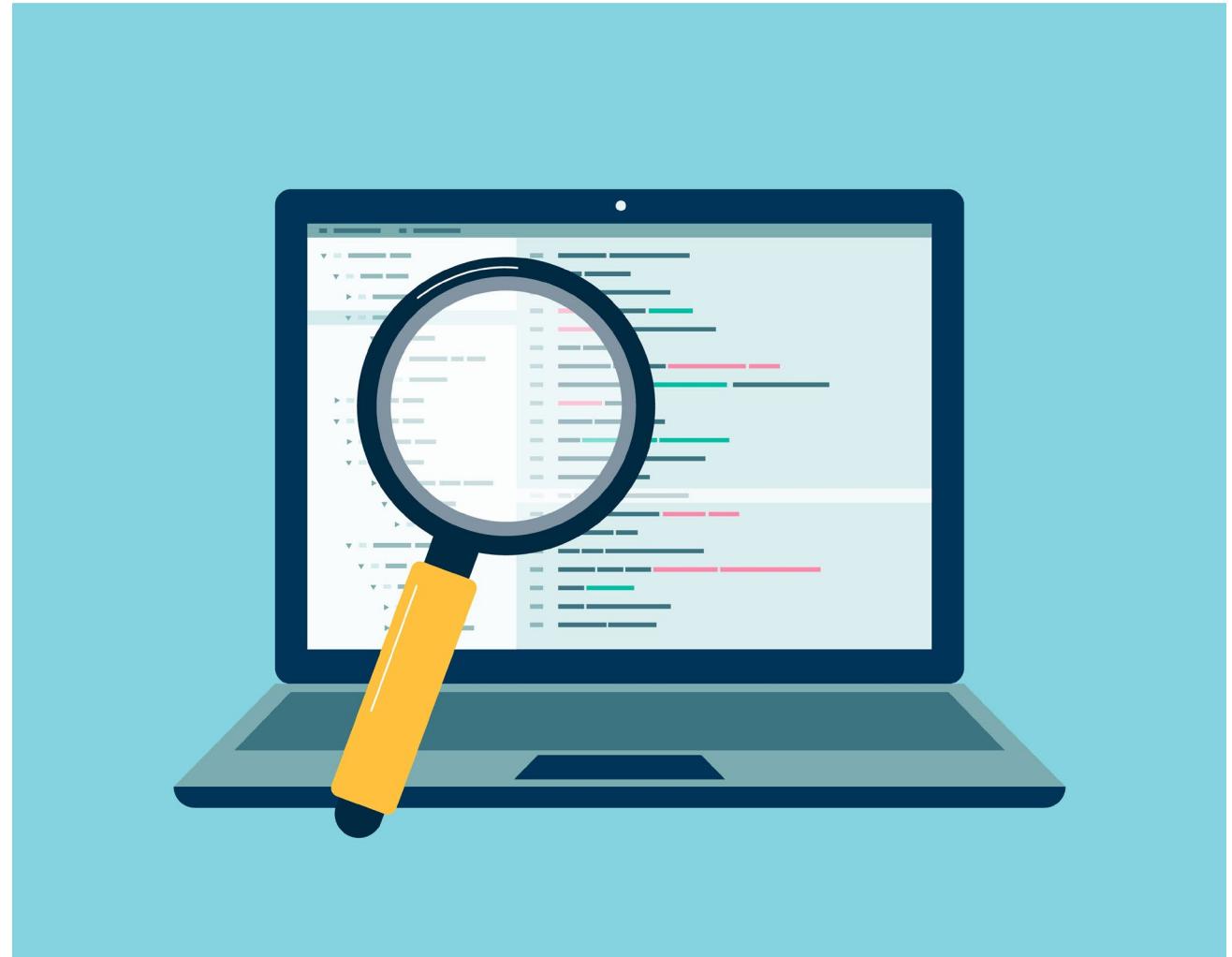
Code Reviews

Opportunities to Identify Technical Debt



Code Analysis Tools

Opportunities to Identify Technical Debt



Pull Requests

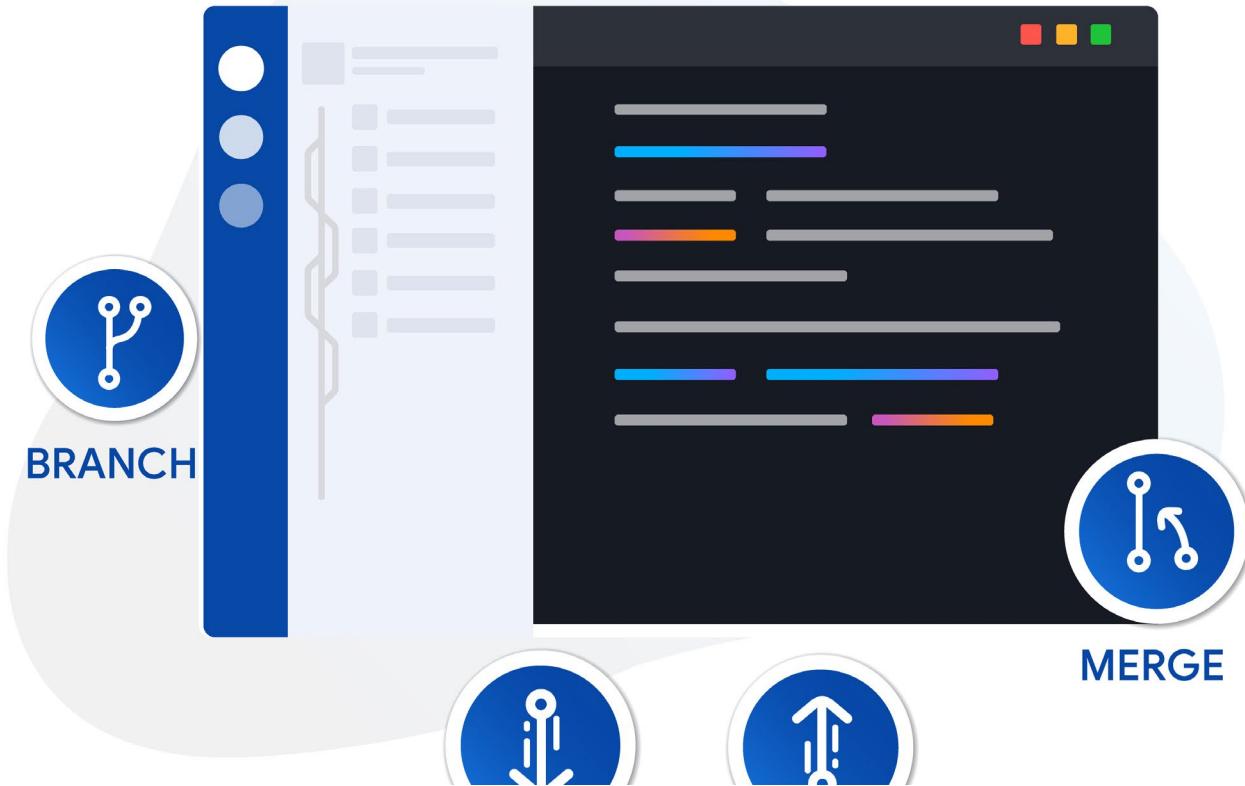
Opportunities to Identify Technical Debt



GIT REPOSITORIES Software Development



GIT FLOW



Stand Up

Opportunities to Identify Technical Debt



Sprint Review

Opportunities to Identify Technical
Debt



Often hidden in plain sight

Opportunities to Identify Technical Debt



Recording Technical Debt



Recoding Technical Debt



- User Story
- Product Backlog Item
- Requirement
- Issue

Recoding Technical Debt



- User Story
- Product Backlog Item
- Requirement
- Issue
- Bug

Key Information to Record

- Problem Statement

We need to refactor our SSO.

Key Information to Record

- Problem Statement

In 22Q4, 20% of system outages were due to the single-sign-on service straining under scale constraints. Given our focus on the enterprise segment, we are taking some time this quarter to ensure that part of the system is ready for use to continue to scale up this segment.

ON, DC

APRIL 19-21

Key Information to Record

- Problem Statement
- Story Points
- Criticality
- Complexity
- Reward
- Remediation Cost



Tracking Technical Debt



Tracking Technical Debt



Tracking Technical Debt



Technical Debt Register

- Product
- User Story #
- Title
- Criticality
- Complexity
- Reward Remediation Cost
- Technical Debt Score



Technical Debt Score

R : Reward

Cr : Criticality

Co: Complexity

$$(R \times .2) + (CR \times .15) - (CO \times .15)$$

Technical Debt Score

$$(R \times .2) + (CR \times .15) - (CO \times .15)$$

| Id | Criticality (CR) | Complexity (CO) | Reward (R) | Score |
|------|------------------|-----------------|------------|-------|
| 1451 | 5 | 1 | 5 | 1.6 |
| 1452 | 1 | 5 | 1 | -0.4 |
| 1453 | 3 | 2 | 3 | 0.75 |

Technical Debt Score

$$(R \times .2) + (CR \times .15) - (CO \times .15)$$

| Id | Criticality (CR) | Complexity (CO) | Reward (R) | Score | Rank |
|------|------------------|-----------------|------------|-------|------|
| 1451 | 5 | 1 | 5 | 1.6 | 1 |
| 1453 | 3 | 2 | 3 | 0.75 | 2 |
| 1452 | 1 | 5 | 1 | -0.4 | 3 |

ONE MORE THING



Technical Debt Ratio



Technical Debt Ratio

$(\text{Remediation Cost} / \text{Development Cost}) \times 100\%$

| Project | Remediation Cost | Development Cost | Technical Debt Ratio |
|-----------|------------------|------------------|----------------------|
| Project 1 | 12.1 | 3,633.12 | 0.33% |
| Project 2 | 702 | 18,728.55 | 3.75% |
| Project 3 | 1,058.25 | 9,593.37 | 11.03% |

Summary



Technical Debt Is Not Free

- What is technical debt?
- Pros and cons of technical debt
- Causes of technical debt
- Types of technical debt
- Identifying technical debt
- Recording technical debt
- Technical Debt Score



Chad Green

Director of Architecture
Glennis Solutions

