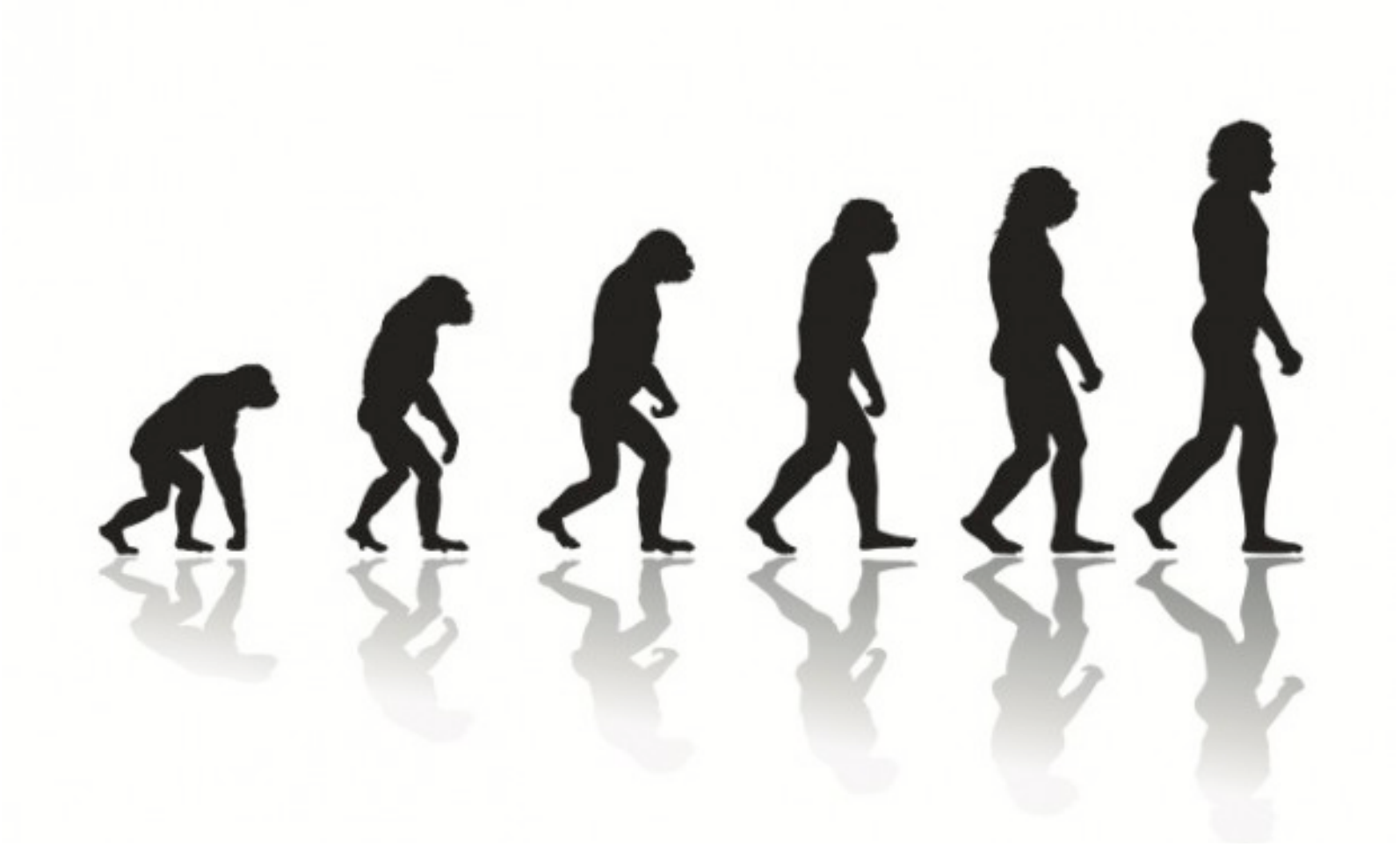
Technical Debt

From Acceptance to Action

Martin Hertz

Engineering Manager, JAMF Software
martin@martinhertz.com

- What is technical debt?
  - Ward Cunningham's metaphor
  - Symptoms, causes, and examples
  - Expanded and alternatives views
- How to approach technical debt?
- Ideal state of technical debt management
- Beyond technical debt



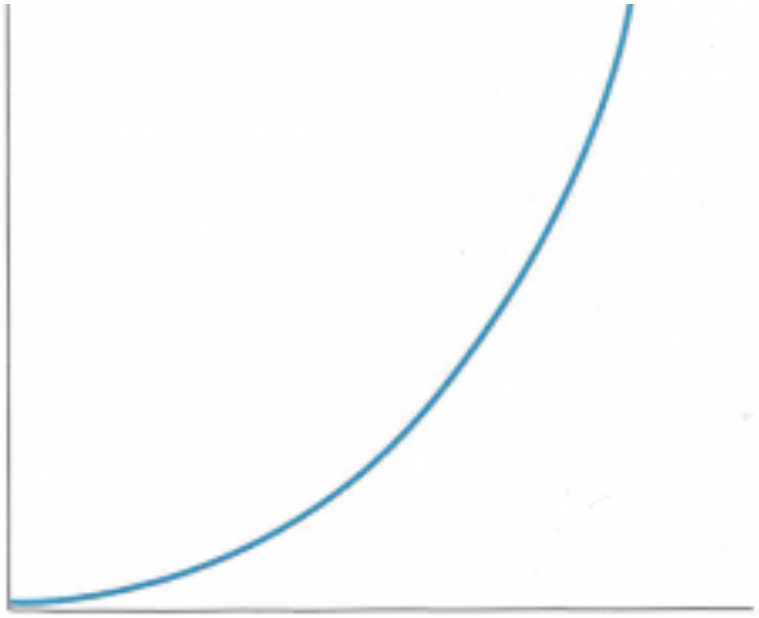
my evolution

“Technical Debt includes those internal things that you choose not to do now, but which will impede future development if left undone. This includes deferred refactoring.

Technical Debt doesn't include deferred functionality, except possibly in edge cases where delivered functionality is "good enough" for the customer, but doesn't satisfy some standard (e.g., a UI element that isn't fully compliant with some UI standard).”

~ Ward Cunningham


BANK



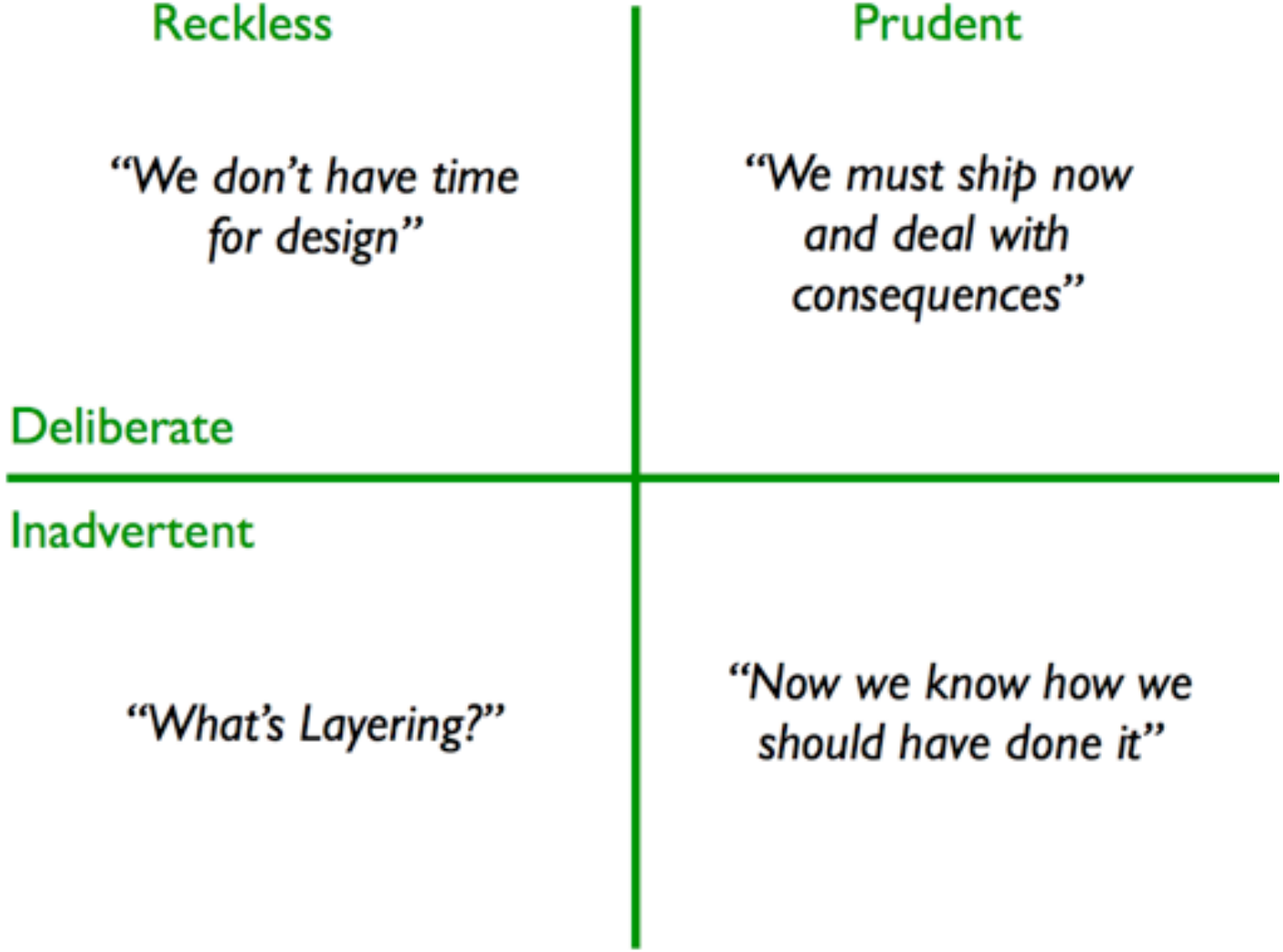
Symptoms

- decreased velocity / increased time cost for delivery of features
- increased cost of regression testing
- extensive post-release stabilization
- limited expertise in talent market
- developer disengagement

Causes

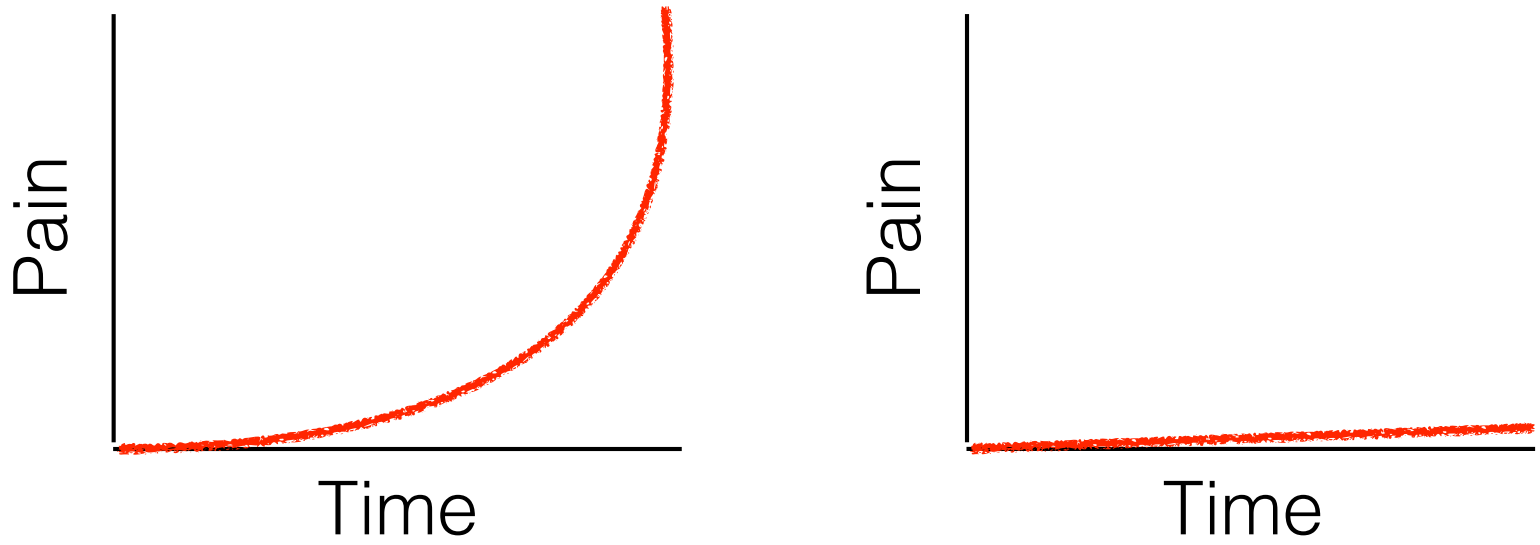
- time pressure
- inexperience / lack of knowledge
- poor communication (around architecture, design, technical concerns, etc.)
- whack-a-mole patching vs. wholistic quality vision
- inattentiveness over time

Examples



~ Martin Fowler

All Tech Debt != Bad



Interest on technical debt compounds over time, but some debt comes with 0% interest

Managing Technical Debt

- **Define** what it is
- **Capture** existing problems and emerging ones
- **Communicate** the issue to everyone all the time
- **Incorporate** it into your development process
- **Celebrate** when you win battles

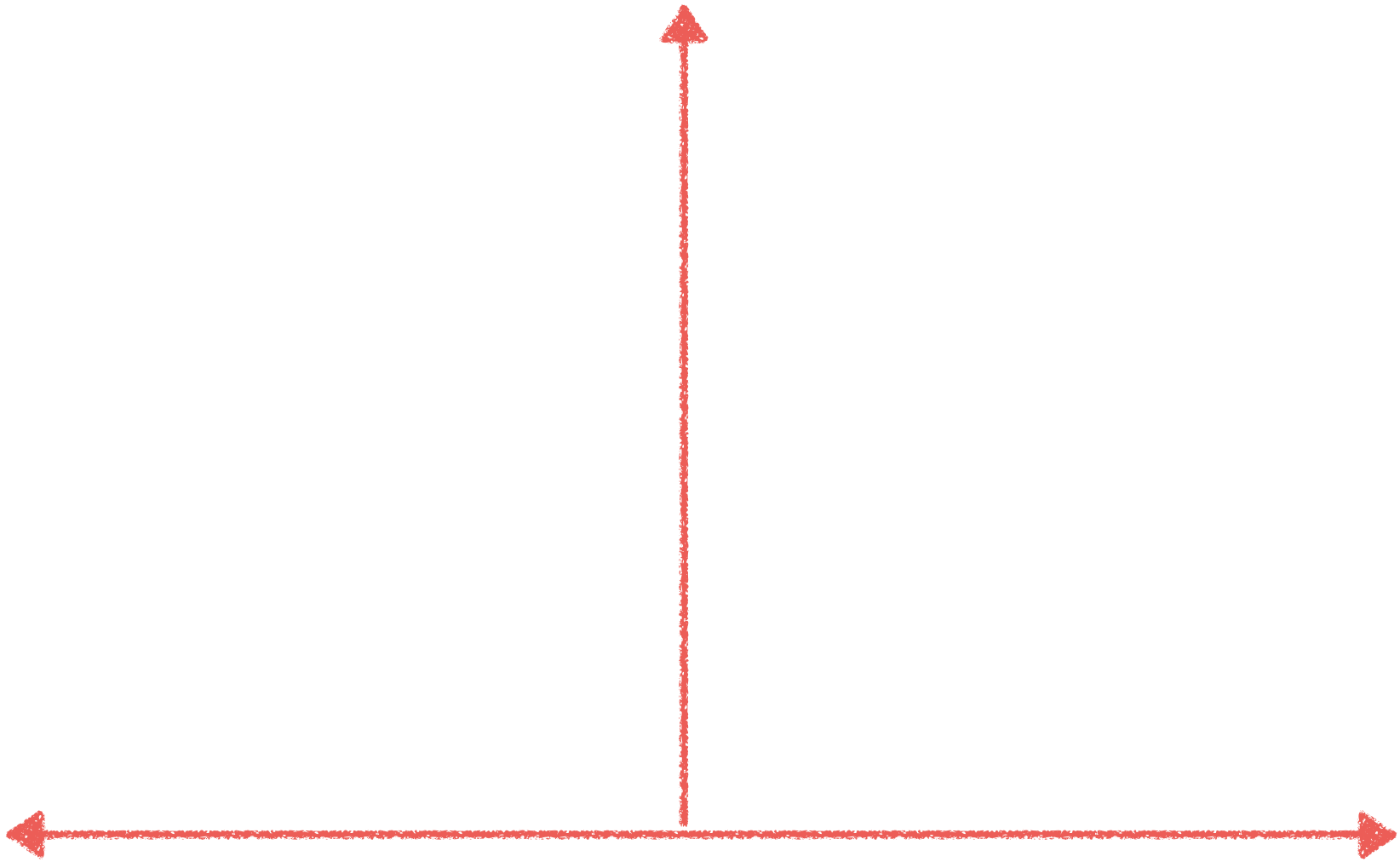
Define Technical Debt

- Unimplemented Features?
- Defects?
- Performance?
- Security?

Capture Technical Debt

- Individual
- Team
- Organization

Communicate Technical Debt

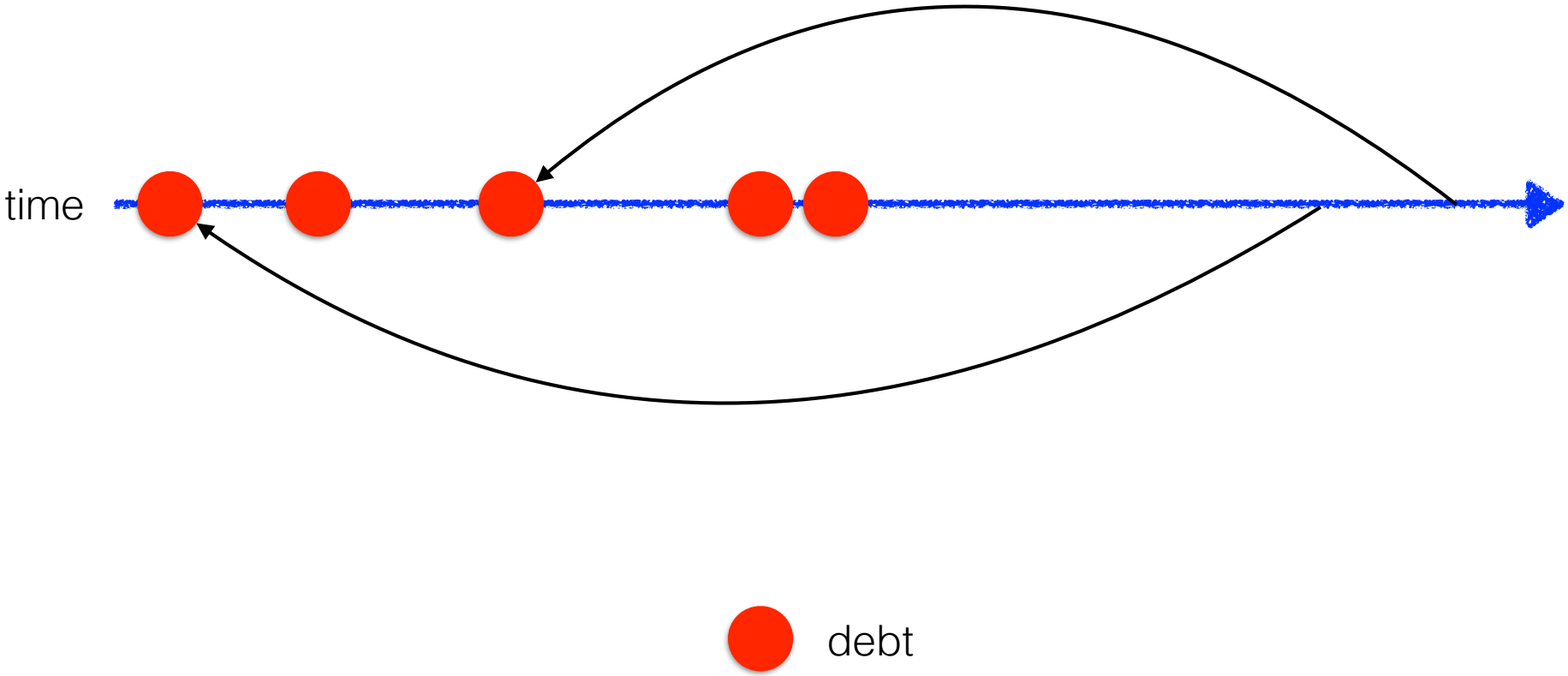


Incorporate Technical Debt

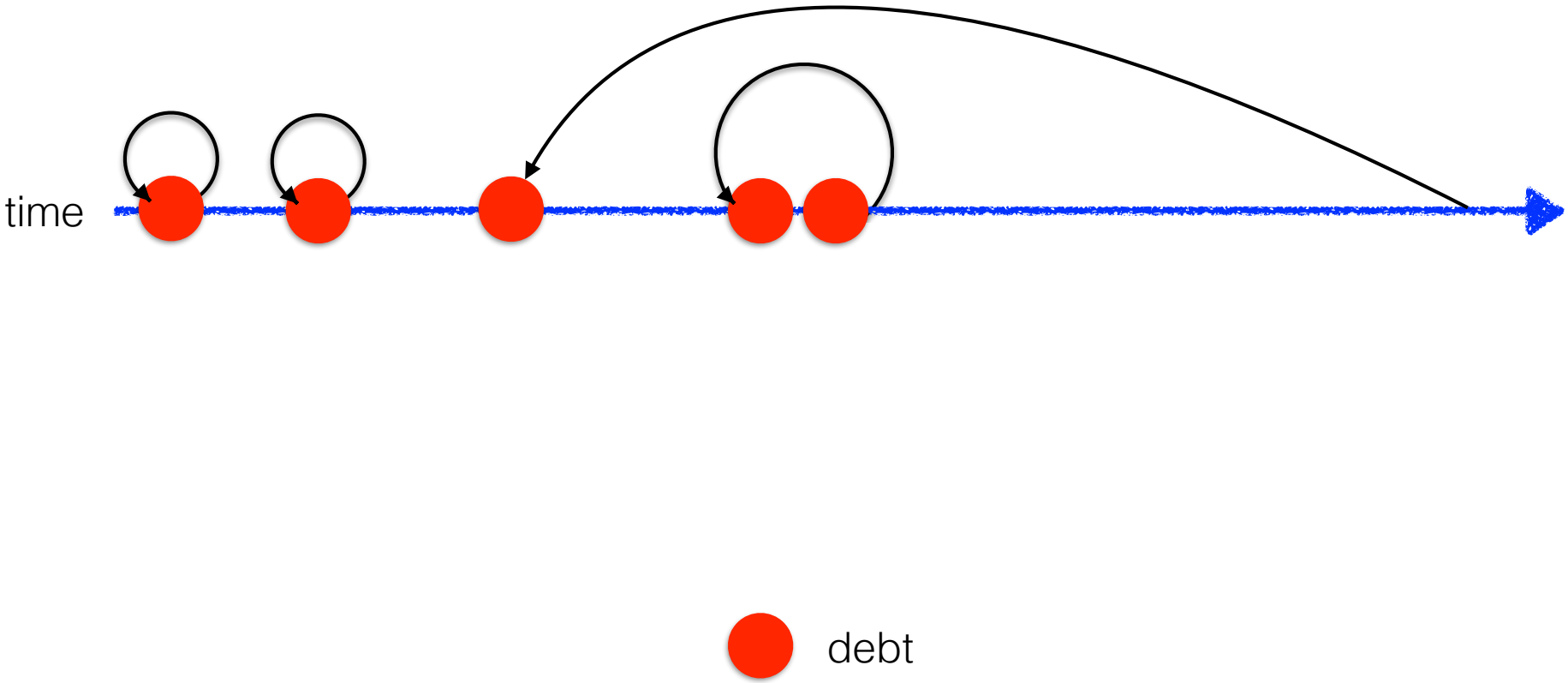
- Process
  - Backlog
  - Retrospectives
  - Planning
- Roadmaps
- Goals
  - Team
  - Individual

Celebrate Technical Debt

Not Ideal State



Ideal State



Quality Debt

Configuration Debt

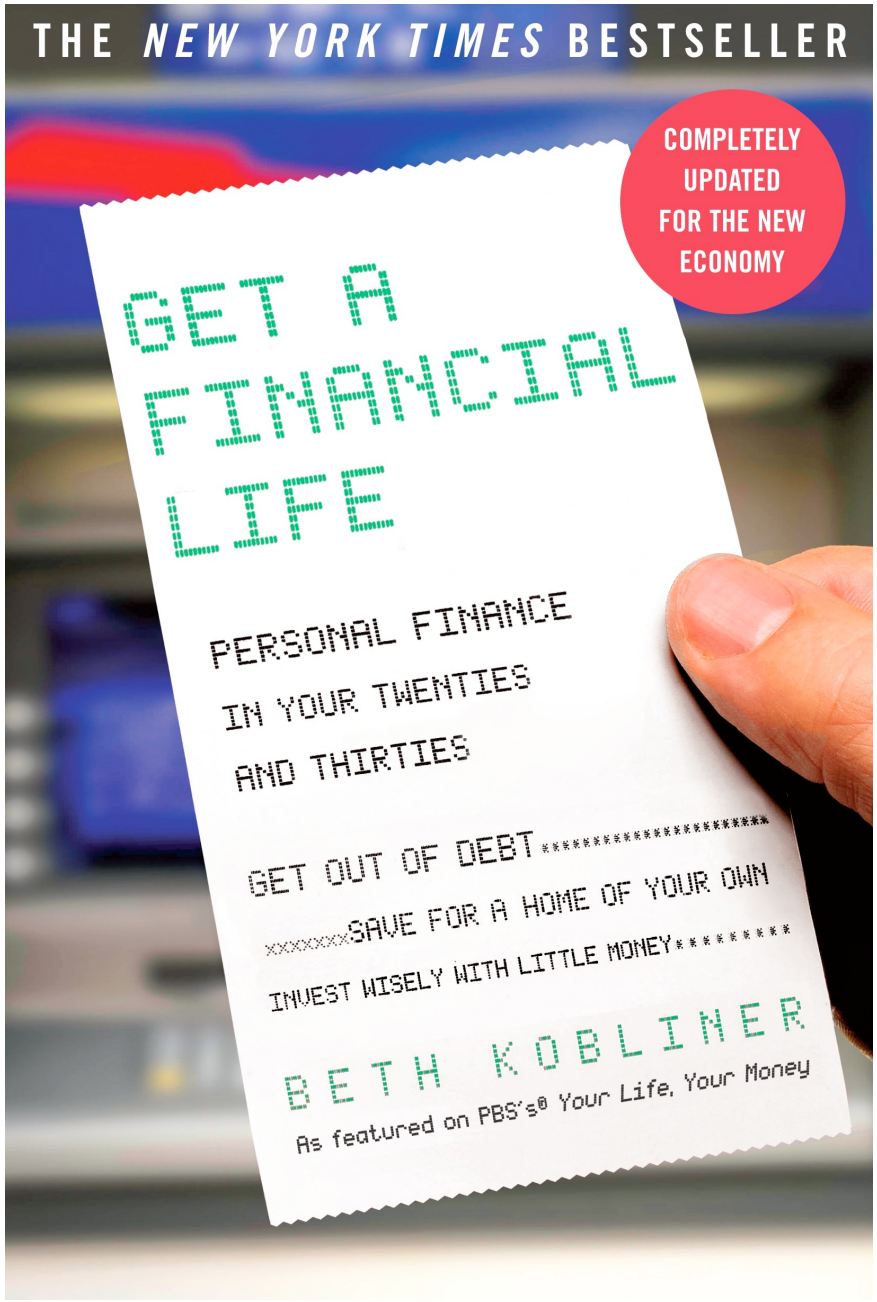
Design Debt

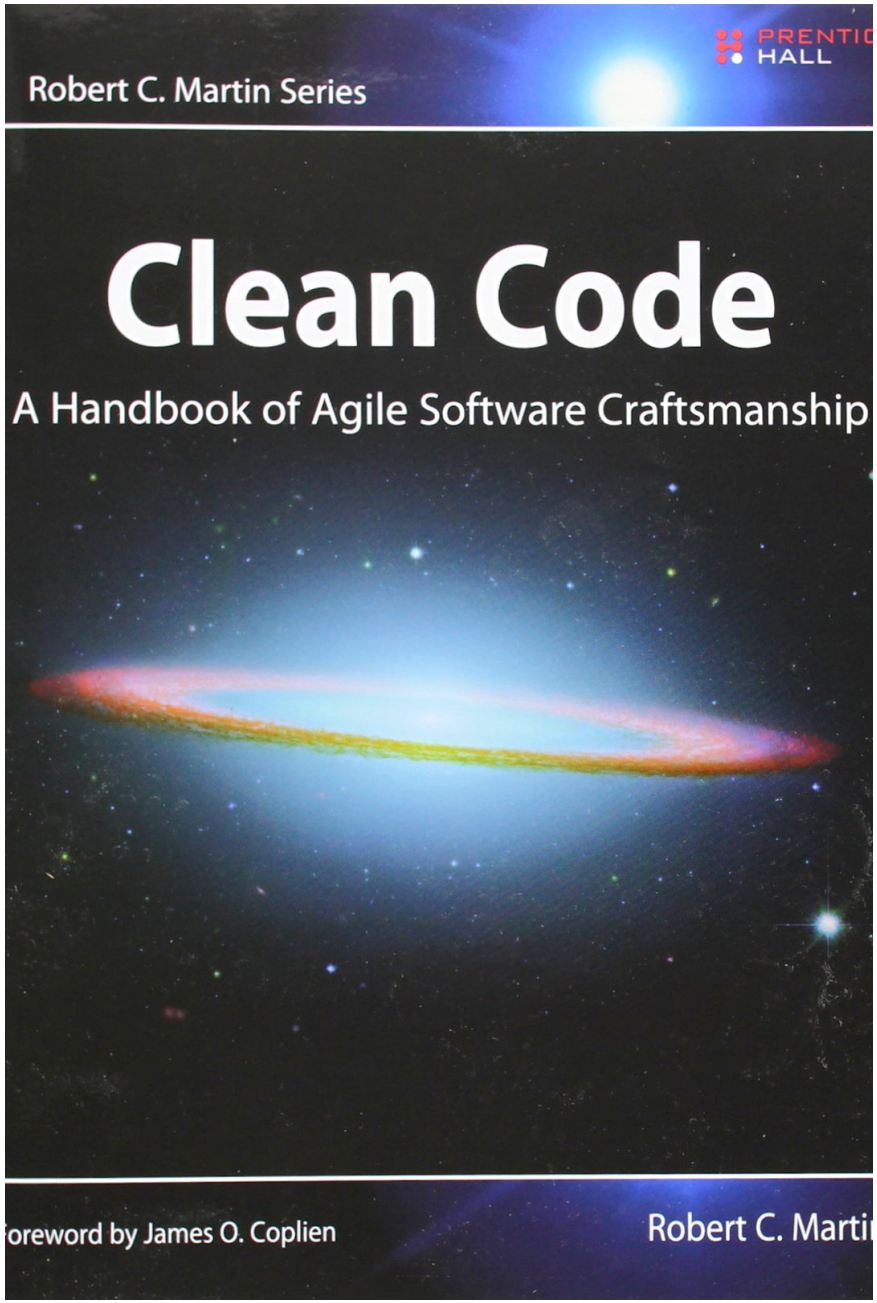
Software Debt

Technical Communications

Testing

Management


THE NEW YORK TIMES BESTSELLER
COMPLETELY UPDATED FOR THE NEW ECONOMY
GET A FINANCIAL LIFE
PERSONAL FINANCE IN YOUR TWENTIES AND THIRTIES
GET OUT OF DEBT
SAVE FOR A HOME OF YOUR OWN
INVEST WISELY WITH LITTLE MONEY
BETH KOBLINER
As featured on PBS's® Your Life, Your Money


Robert C. Martin Series
PRENTICE HALL
Clean Code
A Handbook of Agile Software Craftsmanship
Foreword by James O. Coplien
Robert C. Martin

Thank You!

Questions?

References

- “Managing Software Debt” by Chris Sterling
- Ward Cunningham, via c2.com
- Software Engineering Radio, Episode 224: Sven Johann and Eberhard Wolff on Technical Debt
- Martin Fowler, via martinfowler.com