CS 350 – Software Engineering

November 13, 2017

Week 12, Class #36

Technical Debt

Mark Seaman

MWF – 10:00-11:30 - Kepner 0095F

This Week – Risk

Last Week

Dev Ops

Monday, 11-13

Technical Debt

Wednesday, 11-15

Best Practices

Friday, 11-17

Metrics

Next Week

Teams

Exercises

✧ Version Control

- Markdown Exercise 10/20
- Github Login 10/30
- Git Push 11/13

✧ Development

- Design Plan 10/23
- Development Exercise 10/25
- Pair Programming Exercise 10/30
- Unit Test Exercise 11/3

Technical Debt

✧ What is technical debt?

Technical Debt

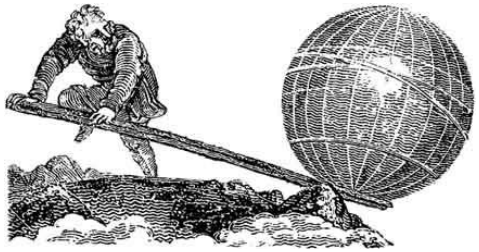
✧ What is technical debt?

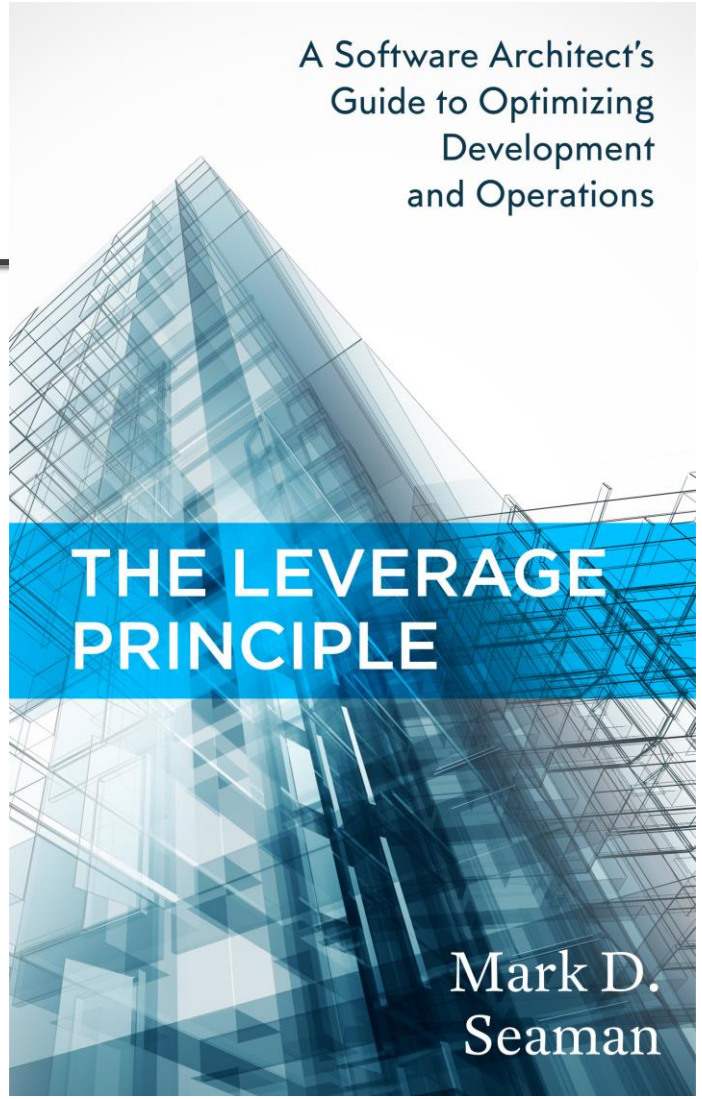
- Promise of later work
- Shortcut for immediate benefit
- Compromise of the right way
- Cumulative effects of bad decisions

Leverage Principle

- ✧ Technical Debt determines cost
- ✧ Best Practices reduce debt

- ✧ Areas of concern
  - Development
  - Operations
  - Teams





Look inside

The Leverage Principle: A Software Architect's Guide to Optimizing Development and Operations Kindle Edition

by Mark D. Seaman (Author), Stacie Seaman (Editor)

3 customer reviews

Software development is expensive and it is far more expensive than it needs to be. The pace of development has increased dramatically with the arrival of cloud-based apps and continuous delivery and the processes for software development and operations have to adapt to this new reality.

Development Lifecycle

- ✧ Technology
  - Fitness of tools used
  - Technical skill level
- ✧ Design
  - Too much or too little design
- ✧ Code
  - Complexity
  - Non-incremental development
- ✧ Test
  - Lack of test or maintenance
  - Poor coverage

Software Operations

- Deployment
  - Capabilities of hosting service
  - Lack of automation
- Release Cycle
  - Too long until next release
- Services
  - Complexity of service interactions
  - Inappropriate scale
- Monitoring
  - Lack of transparency
  - Human observation

Building Teams

- ✧ Knowledge
  - Lacking knowledge management system
- ✧ Teamwork
  - Tolerating bad team members - knowledge hoarders and primadonas
- ✧ Learning
  - Not investing in the training of the developers
- ✧ Project Planning
  - Lack of flexibility
  - Ambiguous priorities
  - No leverage

See you Wednesday

T

- S
  - x