



# Software Development Pearls

**Presenter:**

Karl Wieggers

Principal Consultant

Process Impact

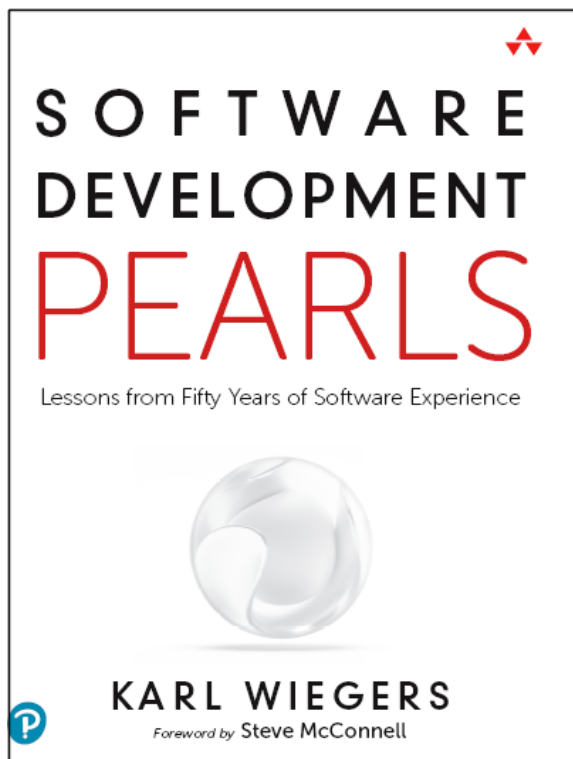
# Presenter: Karl Wiegers

Karl Wiegers is Principal Consultant with Process Impact, a software development consulting and education company in Portland, Oregon. He has a PhD in organic chemistry. His interests include requirements engineering, project management, software quality, and software process improvement. Karl is the author of 13 books and many articles on software and other topics.



Contact Karl at [ProcessImpact.com](http://ProcessImpact.com) or [KarlWiegers.com](http://KarlWiegers.com)

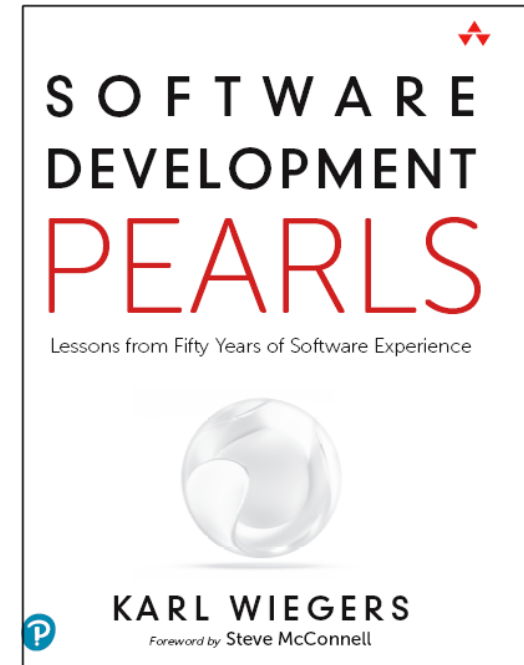
# SOFTWARE DEVELOPMENT PEARLS



Karl Wieggers  
***PROCESS IMPACT***  
[processimpact.com](http://processimpact.com)  
[tinyurl.com/sdpearls](http://tinyurl.com/sdpearls)

# Software Development Pearls: Agenda

- Book structure
- 16 lessons about requirements
- 6 lessons about design
- 12 lessons about project management
- 8 lessons about culture and teamwork
- 8 lessons about quality
- 9 lessons about process improvement
- What to do next



# Structure of the Book

- **Each topic chapter includes:**

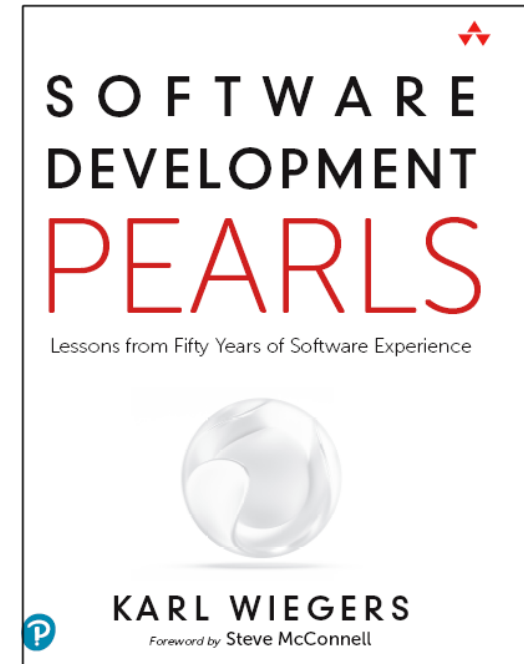
- introduction to the topic
- First Steps
- 6 to 16 lessons
- Next Steps

- **First Steps**

- list current practices you're good at
- identify problems, impacts, and root causes

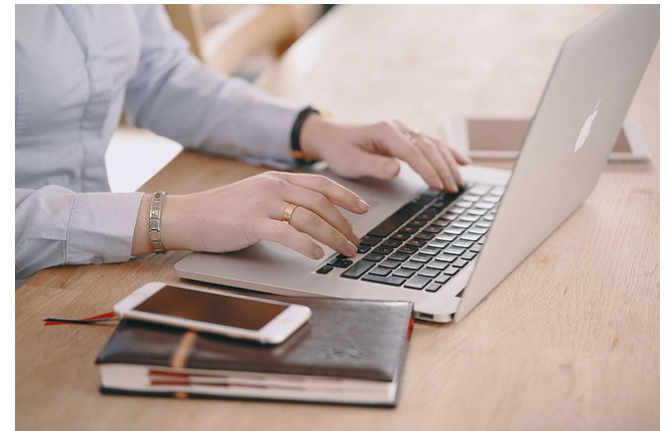
- **Next Steps**

- identify relevant lessons and your own lessons
- identify practices that might address problems
- identify barriers to implementing each practice
- put templates, procedures, and guidance in place



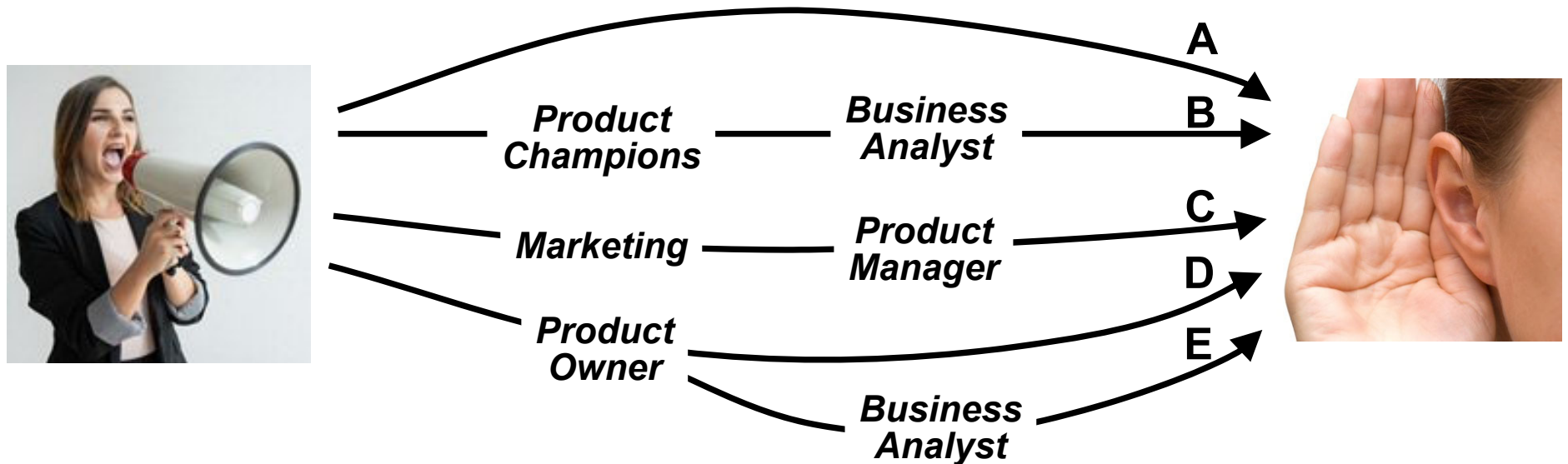
# Some Requirements Lessons

1. If you don't get the requirements right, it doesn't matter how well you execute the rest of the project.
4. A usage-centric approach to requirements will meet customer needs better than a feature-centric approach.
6. Agile requirements aren't different from other requirements.
7. The cost of recording knowledge is small compared to the cost of acquiring knowledge.



# A Requirements Example

**12. Requirements elicitation must bring the customer's voice close to the developer's ear.**

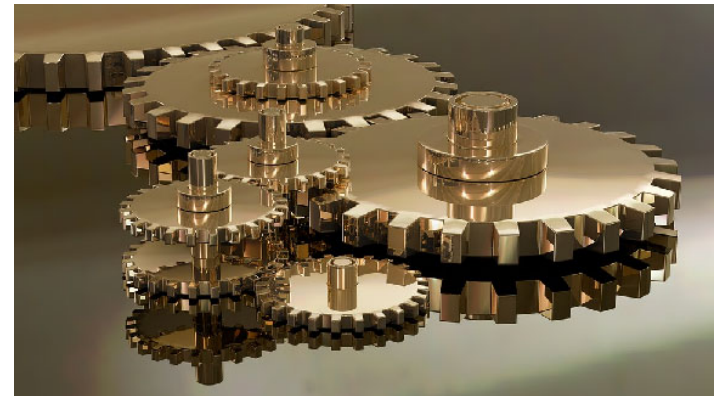


# Some Design Lessons

---

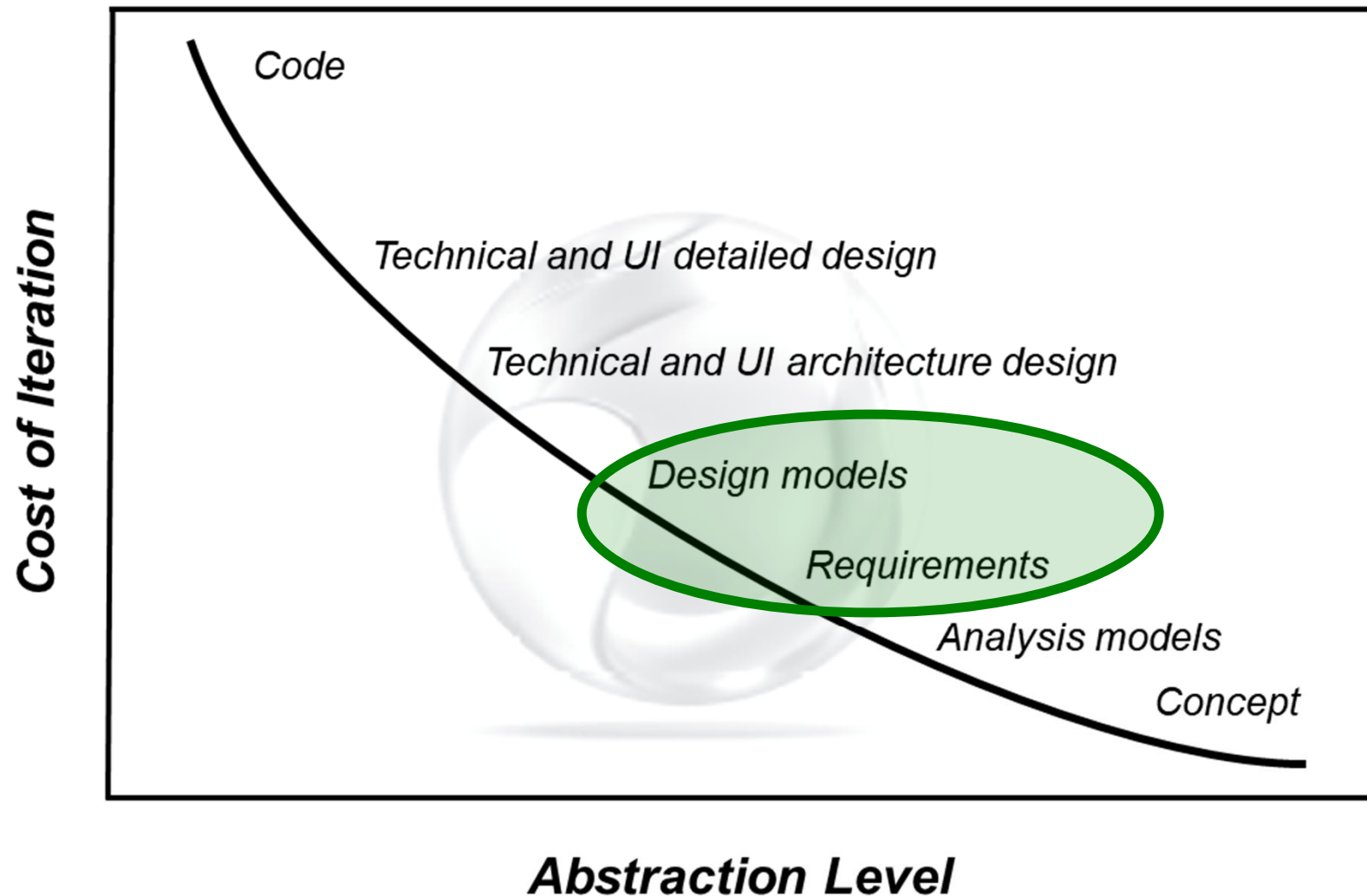
17. Design demands iteration.

18. It's cheaper to iterate at higher levels of abstraction.





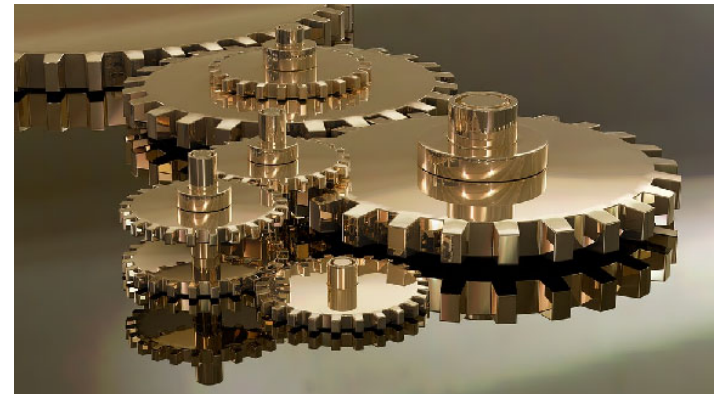
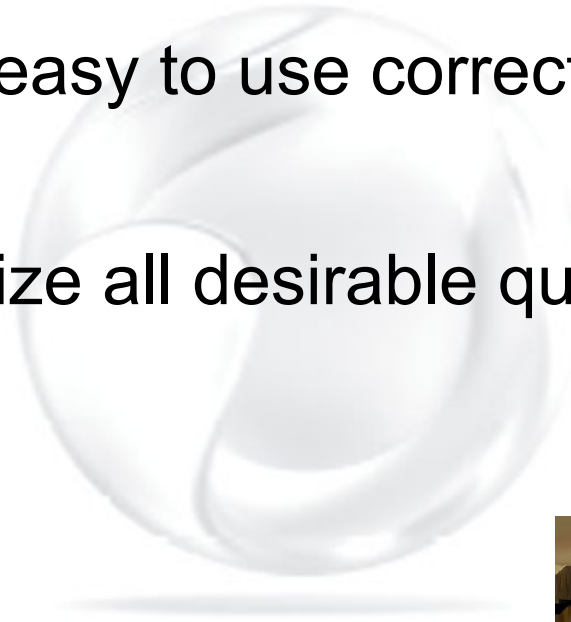
# Iteration and Levels of Abstraction



# Some Design Lessons

---

- 17. Design demands iteration.
- 18. It's cheaper to iterate at higher levels of abstraction.
- 19. Make products easy to use correctly and hard to use incorrectly.
- 20. You can't optimize all desirable quality attributes.



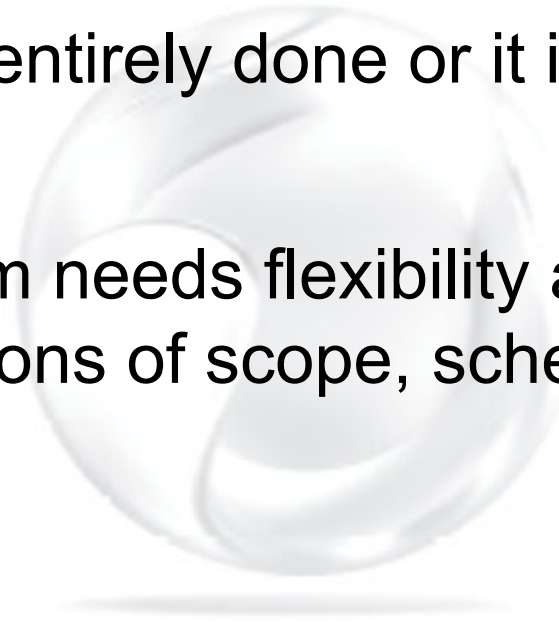
# A Design Example

## 21. An ounce of design is worth a pound of recoding.

- Rework clobbers productivity
- Technical debt
  - shortcomings that someone must resolve in the future to maintain the product's proper functioning and expandability
  - design *minimizes* technical debt
- Refactoring
  - restructuring existing code to improve its design without changing its functioning
  - refactoring *pays off* technical debt
- Architectural restructuring is highly disruptive
- Balance **pre**factoring [aka design] against **re**factoring

# Some Project Management Lessons

- 24. Don't give anyone an estimate off the top of your head.
- 25. Icebergs are always larger than they first appear.
- 30. A task is either entirely done or it is not done: no partial credit.
- 31. The project team needs flexibility around at least one of the five dimensions of scope, schedule, staff, budget, and quality.



# A Project Management Example

## 23. Work plans must account for friction.

- People do not multitask—they task switch.
- Excessive task switching reduces productivity.
  - it takes time to mentally switch contexts
  - interruptions disrupt **flow**
- Effective project hours < hours at work
  - plan on about 5 effective hours/day
- Other sources of project friction:
  - maintenance interruptions
  - communication over distance
  - cultural differences



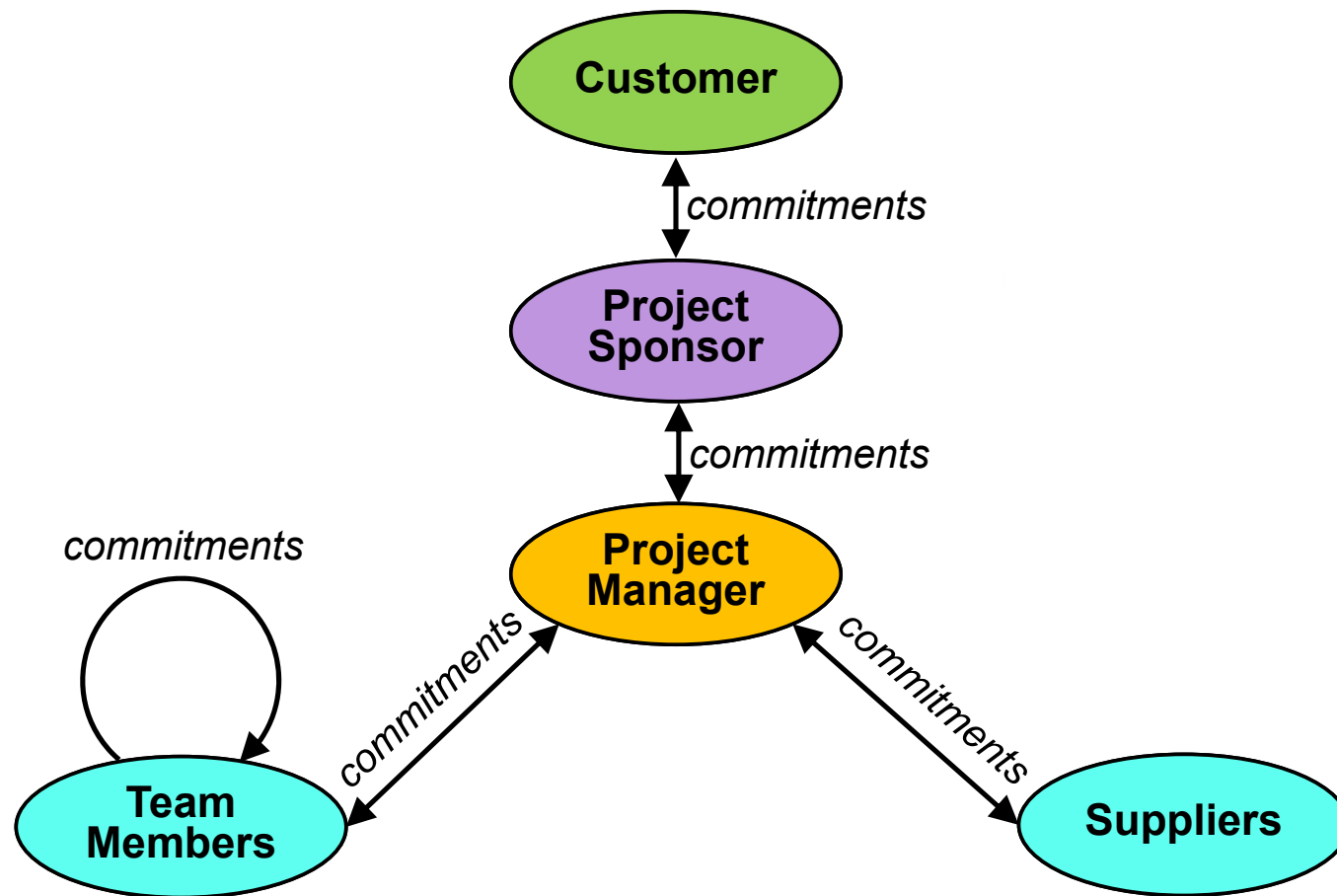
# Some Culture and Teamwork Lessons

- 35. Knowledge is not zero-sum.
- 37. Without training and better practices, don't expect higher productivity to happen by magic.
- 38. People talk a lot about their rights, but the flip side of every right is a responsibility.
- 41. Don't underestimate the challenge of changing an organization's culture as it moves toward new ways of working.



# A Culture and Teamwork Example

**36. No matter how much pressure others exert, never make a commitment you know you can't fulfill.**



# Some Quality Lessons

---

- 43. When it comes to software quality, you can pay now or pay more later.
- 45. Organizations never have time to build software right, yet they find the resources to fix it later.
- 48. Strive to have a peer, rather than a customer, find a defect.
- 50. Today's "gotta get it out right away" development project is tomorrow's maintenance nightmare.

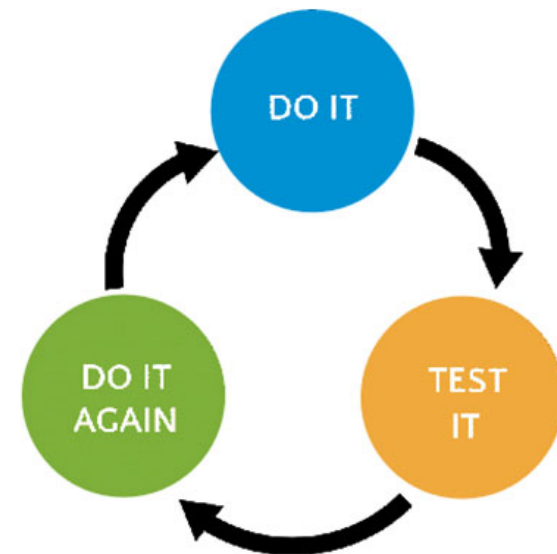




# A Quality Example

## 44. High quality naturally leads to higher productivity.

- Why isn't your productivity as high as you like?
- Rework reduces productivity
  - some rework is unavoidable
  - could be 40 to 50% of total effort
  - what's your organization's rework level?
- 4 Components of the cost of quality
  - defect prevention
  - quality appraisal
  - internal failure
  - external failure
- Increase quality to increase productivity!



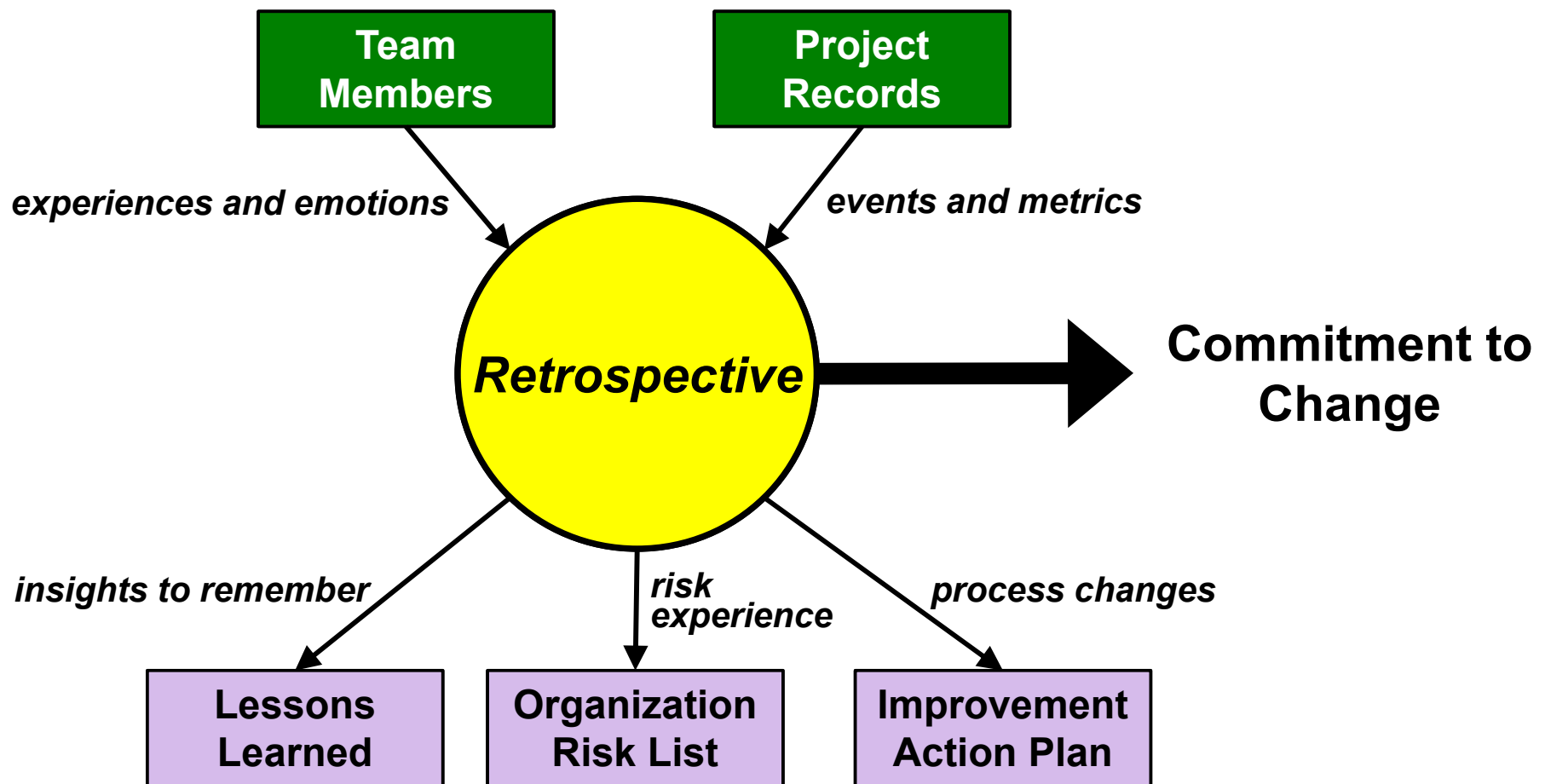
# Some Process Improvement Lessons

- 51. Watch out for “Management by Businessweek.”
- 52. Ask not, “What’s in it for me?” Ask, “What’s in it for us?”
- 53. The best motivation for changing how people work is pain.
- 54. When steering an organization toward new ways of working, use gentle pressure, relentlessly applied.



# A Process Improvement Example

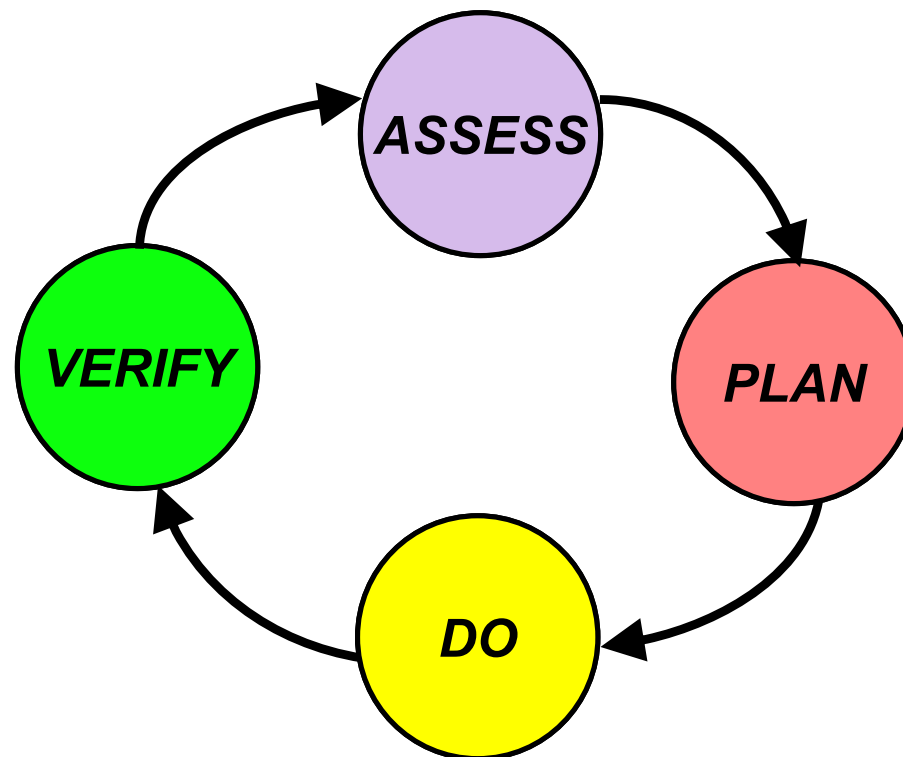
58. Unless you take the time to learn and improve, don't expect the next project to go any better than the last one.



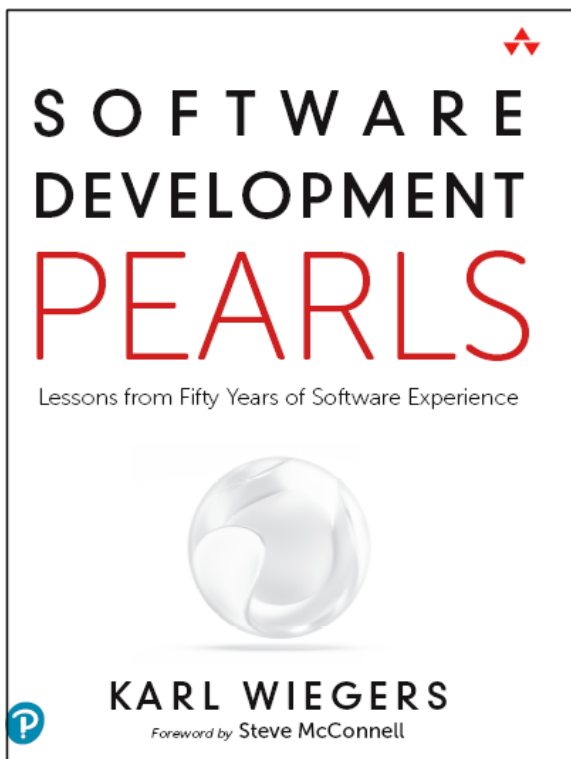
# What To Do Next

## 60. You can't change everything at once.

- A keyword in process improvement: **FOCUS**
  - assess problem areas, select useful changes, and prioritize them
  - choose 2 areas to improve on every project



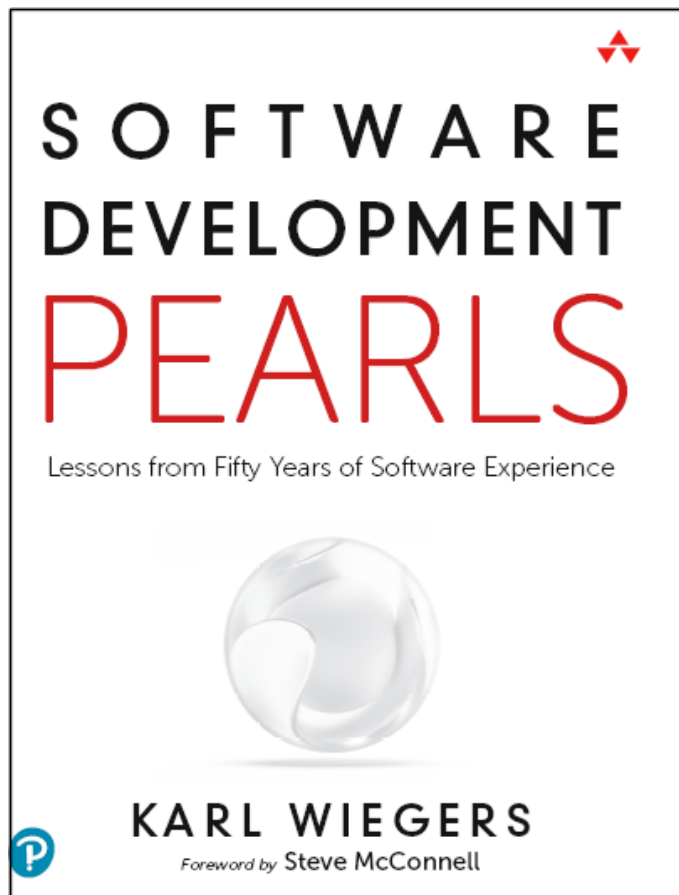
# SOFTWARE DEVELOPMENT PEARLS



Karl Wieggers  
***PROCESS IMPACT*** 

[processimpact.com](http://processimpact.com)  
[tinyurl.com/sdpearls](http://tinyurl.com/sdpearls)

R ughu# #Vdyh#68 ( -#dw#qirp wfrp 2z hjhuw



- Use code **WIEGERS** during checkout
- Offer only good at informit.com
- Print book – free U.S. shipping\*\*
- eBook – DRM-Free PDF, EPUB, & MOBI

\* Discount code WIEGERS is only good at informit.com and cannot be used on the already discounted book + eBook bundle or combined with any other offer.

\*\*If your order is shipping outside of the U.S., it may be subject to import duties and taxes, which are levied once the package reaches the destination country.

[tinyurl.com/sdpearls](http://tinyurl.com/sdpearls)

# Questions

## ◆ For more information

**[tinyurl.com/sdpearls](https://tinyurl.com/sdpearls)**

**[KarlWiegers.com](https://KarlWiegers.com)**

**[ProcessImpact.com](https://ProcessImpact.com)**