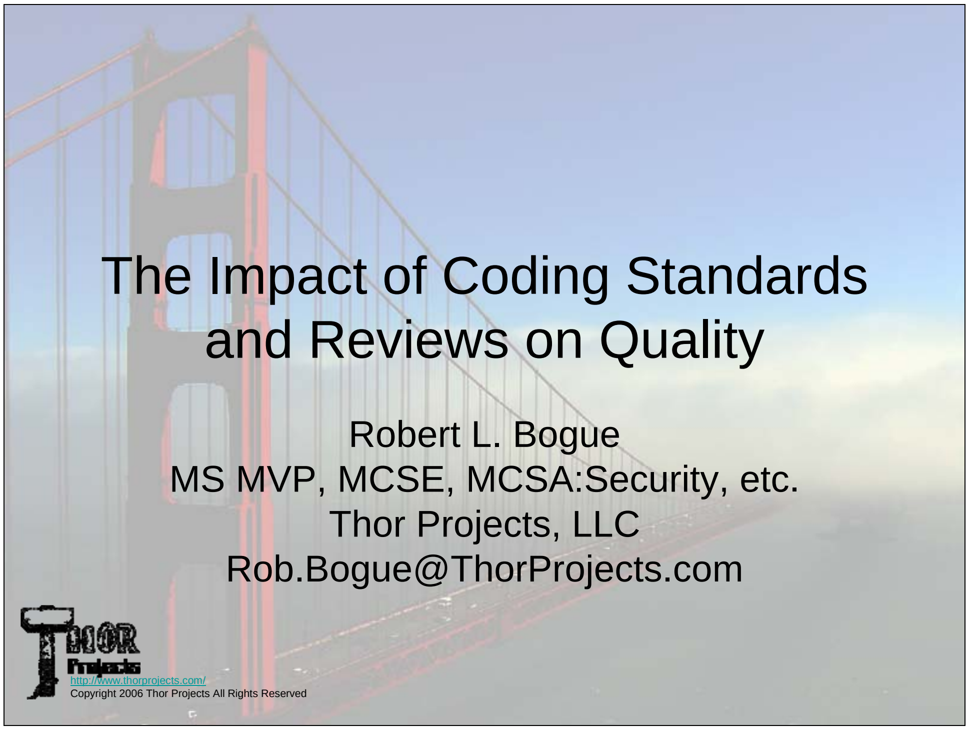
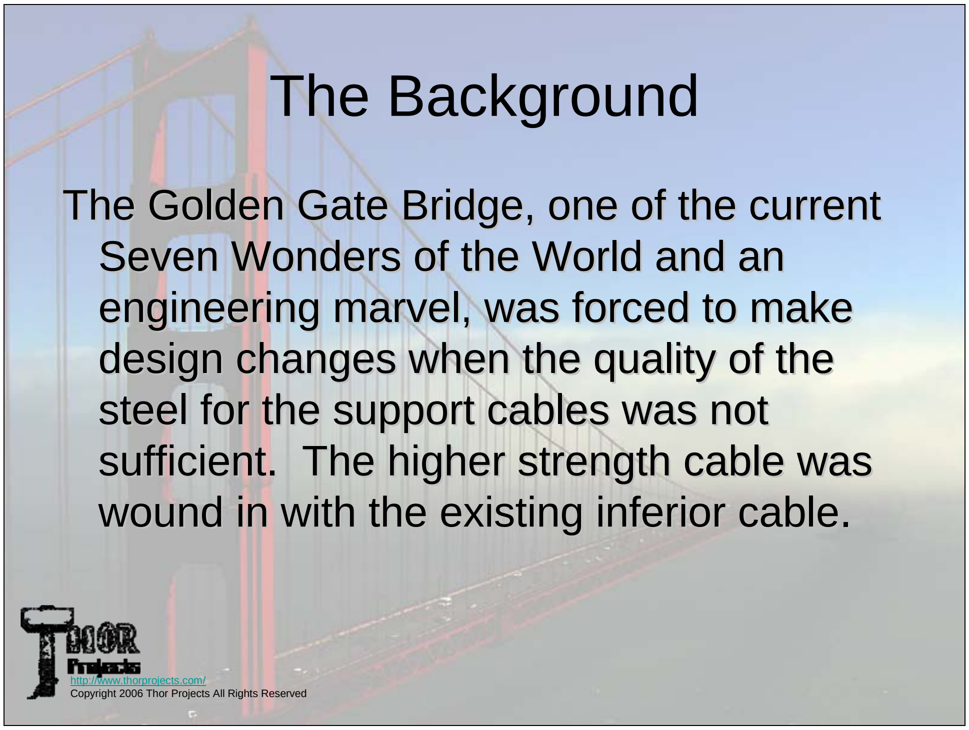


The Impact of Coding Standards and Reviews on Quality

Robert L. Bogue
MS MVP, MCSE, MCSA:Security, etc.
Thor Projects, LLC
Rob.Bogue@ThorProjects.com

http://www.thorprojects.com/

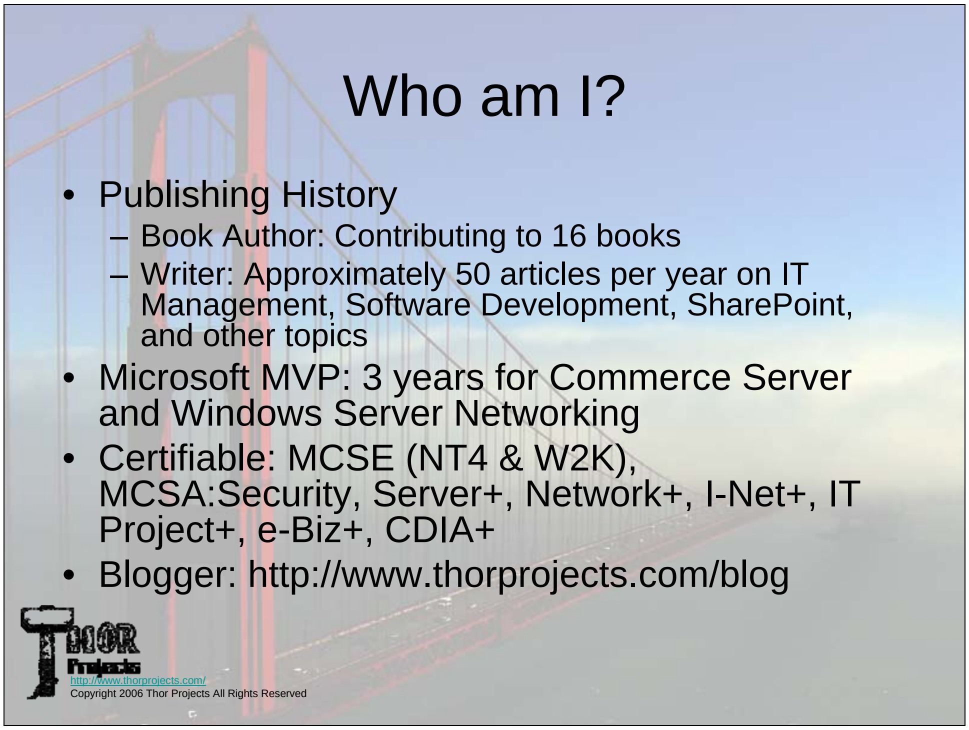


The Background

The Golden Gate Bridge, one of the current Seven Wonders of the World and an engineering marvel, was forced to make design changes when the quality of the steel for the support cables was not sufficient. The higher strength cable was wound in with the existing inferior cable.

http://www.thorprojects.com/

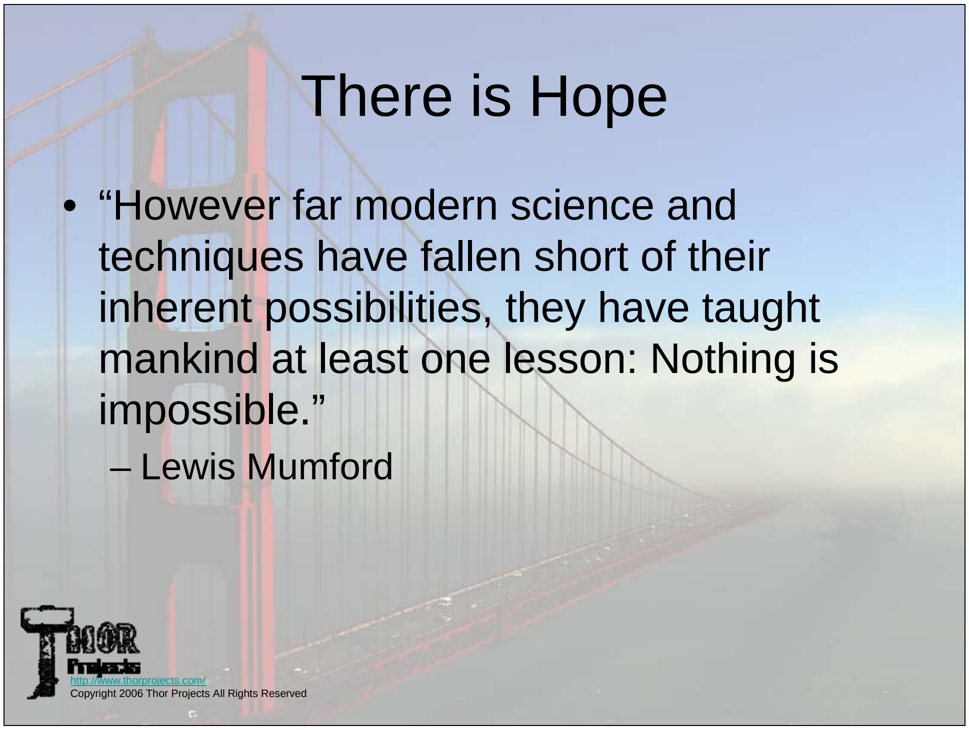
To Engineer Is Human : The Role of Failure in Successful Design
by Henry Petroski



Who am I?

- Publishing History
  - Book Author: Contributing to 16 books
  - Writer: Approximately 50 articles per year on IT Management, Software Development, SharePoint, and other topics
- Microsoft MVP: 3 years for Commerce Server and Windows Server Networking
- Certifiable: MCSE (NT4 & W2K), MCSA:Security, Server+, Network+, I-Net+, IT Project+, e-Biz+, CDIA+
- Blogger: http://www.thorprojects.com/blog

http://www.thorprojects.com/

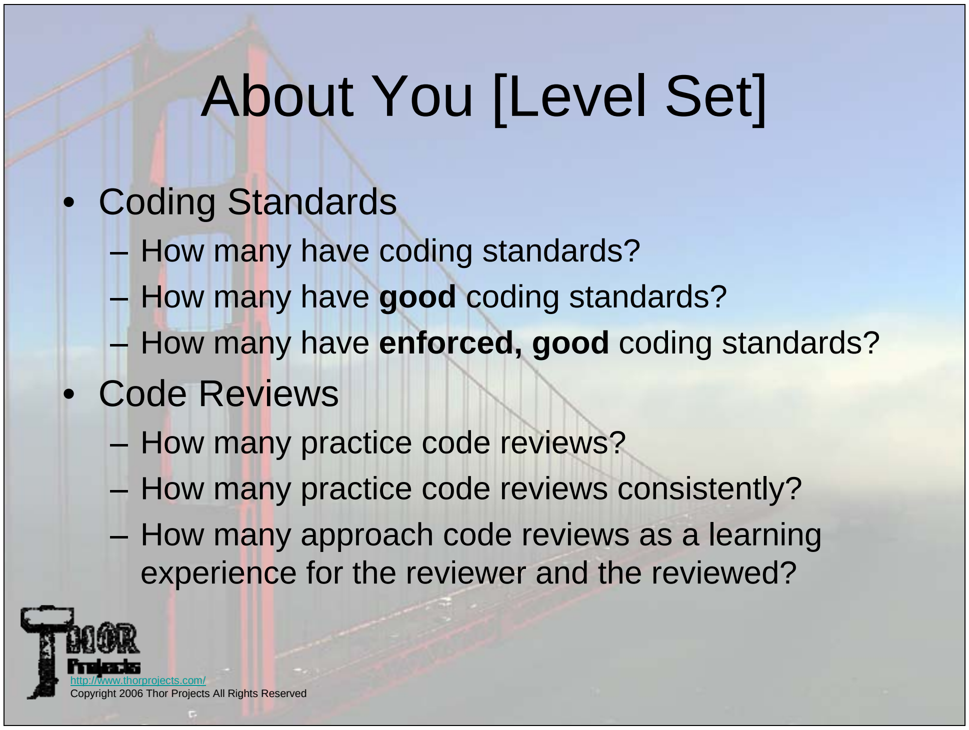


There is Hope

- “However far modern science and techniques have fallen short of their inherent possibilities, they have taught mankind at least one lesson: Nothing is impossible.”
  - – Lewis Mumford

THOR Projects

http://www.thorprojects.com/



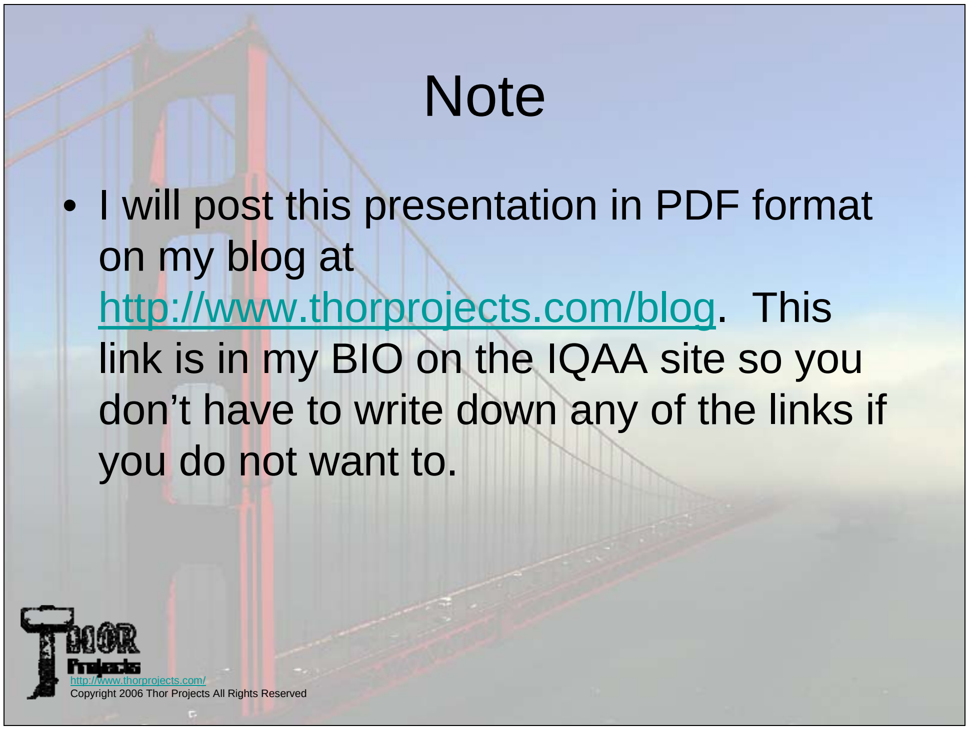
About You [Level Set]

- Coding Standards
  - How many have coding standards?
  - How many have **good** coding standards?
  - How many have **enforced, good** coding standards?
- Code Reviews
  - How many practice code reviews?
  - How many practice code reviews consistently?
  - How many approach code reviews as a learning experience for the reviewer and the reviewed?

Agenda

- Current State of the Industry
- Coding Standards
- Code Reviews
- What You Can Do
- Special Offer

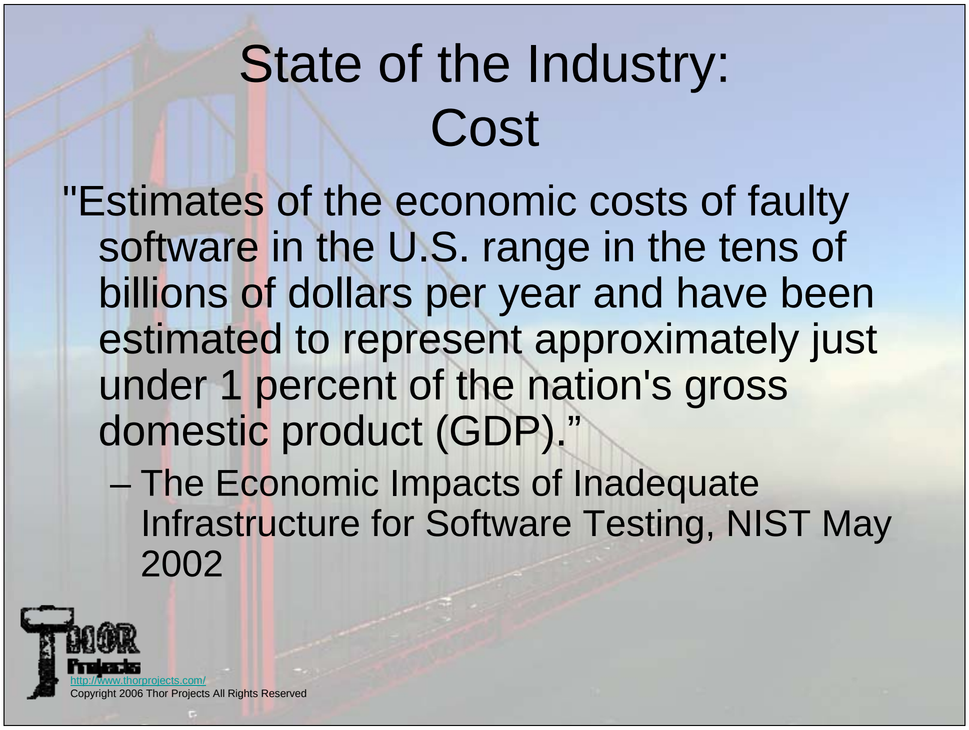
http://www.thorprojects.com/



Note

- I will post this presentation in PDF format on my blog at http://www.thorprojects.com/blog. This link is in my BIO on the IQAA site so you don’t have to write down any of the links if you do not want to.

http://www.thorprojects.com/



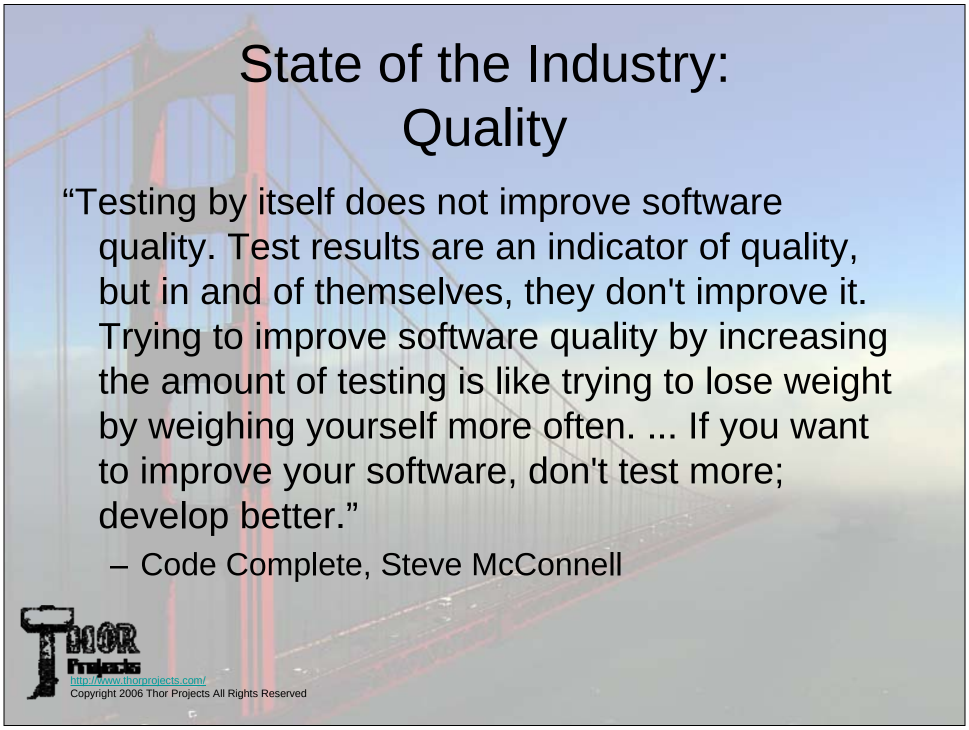
State of the Industry: Cost

"Estimates of the economic costs of faulty software in the U.S. range in the tens of billions of dollars per year and have been estimated to represent approximately just under 1 percent of the nation's gross domestic product (GDP)."

– The Economic Impacts of Inadequate Infrastructure for Software Testing, NIST May 2002

THOR Projects

http://www.thorprojects.com/



State of the Industry: Quality

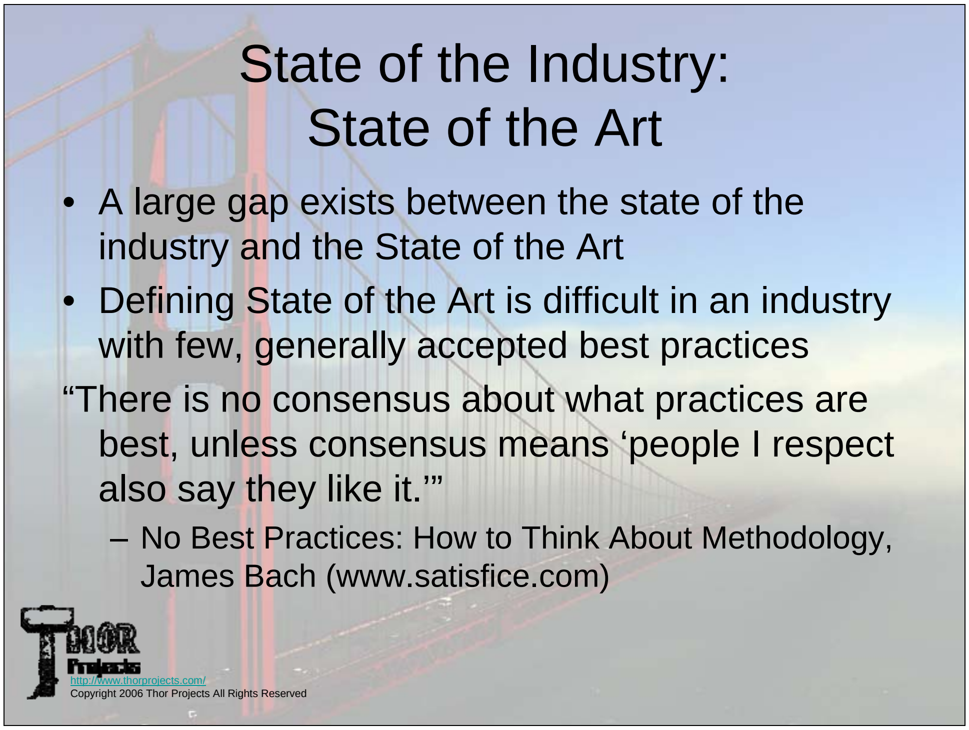
> "Testing by itself does not improve software quality. Test results are an indicator of quality, but in and of themselves, they don't improve it. Trying to improve software quality by increasing the amount of testing is like trying to lose weight by weighing yourself more often. ... If you want to improve your software, don't test more; develop better."
>
> – Code Complete, Steve McConnell

http://www.thorprojects.com/

•Seen as TESTING

•Must have a broader view about "baking in" quality

•Testing by itself does not improve software quality. Test results are an indicator of quality, but in and of themselves, they don't improve it. Trying to improve software quality by increasing the amount of testing is like trying to lose weight by weighing yourself more often. What you eat before you step onto the scale determines how much you will weigh, and the software development techniques you use determine how many errors testing will find. If you want to lose weight, don't buy a new scale; change your diet. If you want to improve your software, don't test more; develop better



State of the Industry: State of the Art

- A large gap exists between the state of the industry and the State of the Art
- Defining State of the Art is difficult in an industry with few, generally accepted best practices

“There is no consensus about what practices are best, unless consensus means ‘people I respect also say they like it.’”

- – No Best Practices: How to Think About Methodology, James Bach (www.satisfice.com)

http://www.thorprojects.com/

Story: Diabetes Advantage Program (Roche) http://care.diabetesjournals.org/cgi/content/full/24/6/1079?ijkey=cda10b1a9c2791b15f2e1d1481567ec73e670946

•Published research established care guidelines for patients with diabetes

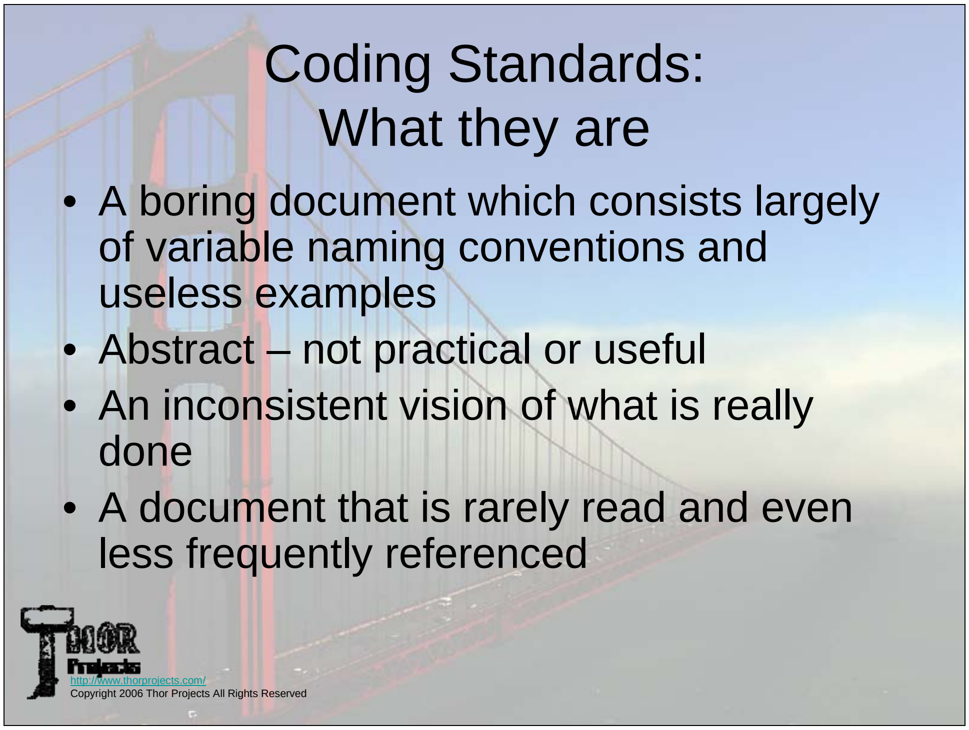
•Most patients with diabetes see primary care physicians

•Primary care physicians were not familiar with the “state of the art” for treating patients with diabetes

•Primary care physicians were not consistently applying “state of the art” practices for treating patients with diabetes

•Information technology was used to guide physicians into using “state of the art” practices for treating their patients with diabetes – without them reading or remembering.

•Clinically significant results

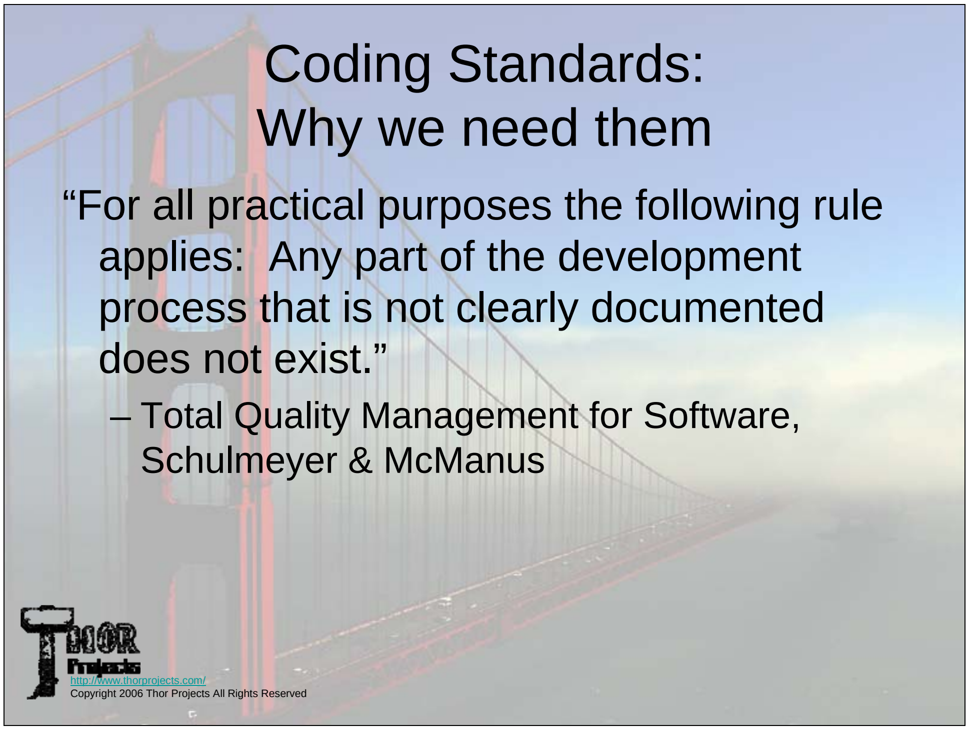


Coding Standards: What they are

- A boring document which consists largely of variable naming conventions and useless examples
- Abstract – not practical or useful
- An inconsistent vision of what is really done
- A document that is rarely read and even less frequently referenced

http://www.thorprojects.com/

•Coding Standards: Creation Process Involvement is important.

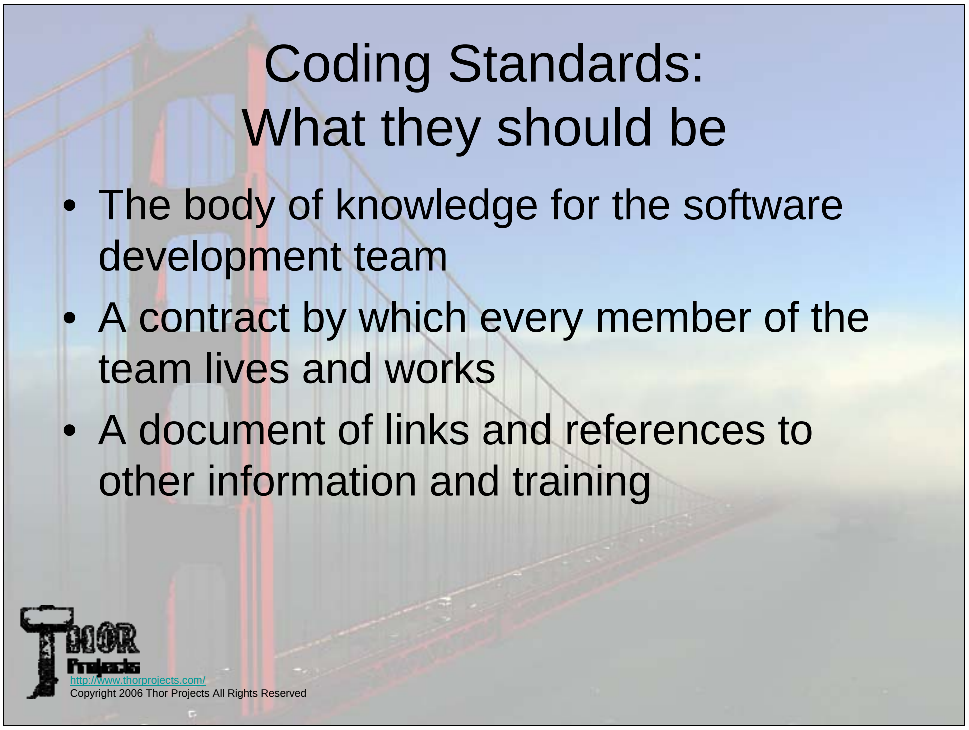


Coding Standards: Why we need them

"For all practical purposes the following rule applies: Any part of the development process that is not clearly documented does not exist."

- Total Quality Management for Software, Schulmeyer & McManus

http://www.thorprojects.com/



Coding Standards: What they should be

- The body of knowledge for the software development team
- A contract by which every member of the team lives and works
- A document of links and references to other information and training

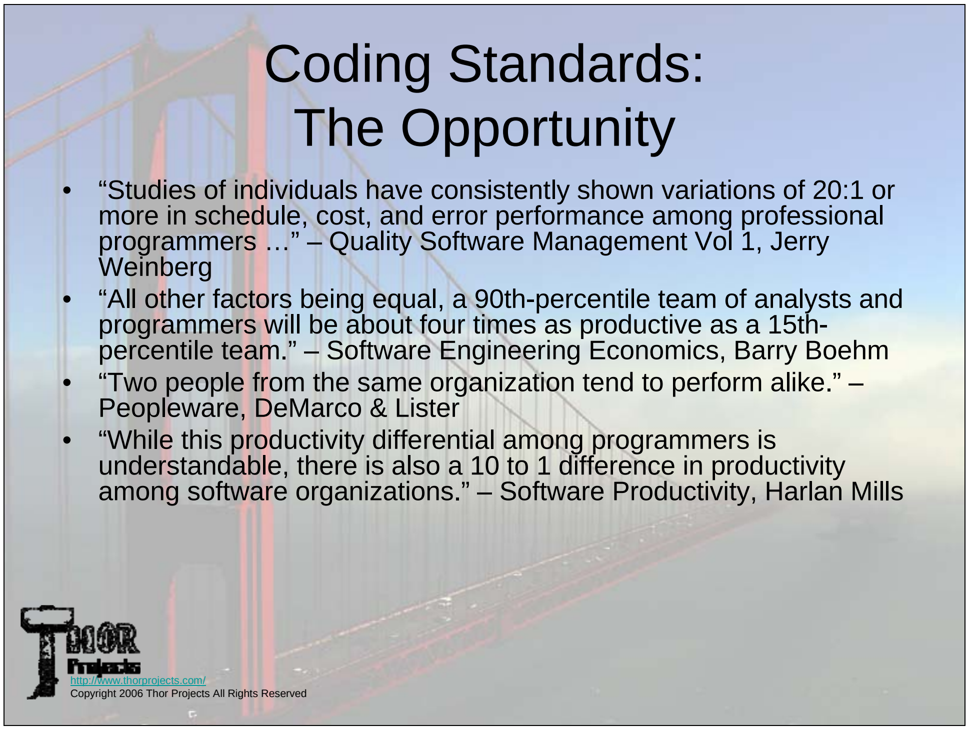
http://www.thorprojects.com/

-REFERENCE: Ground rules Document, applies coding standards to the current project.

-Must be tangible. Not the WHAT but the HOW

> -Comment becomes: Comment to explain why something is being done not what is being done

-Watch for: Inclusion by Reference article

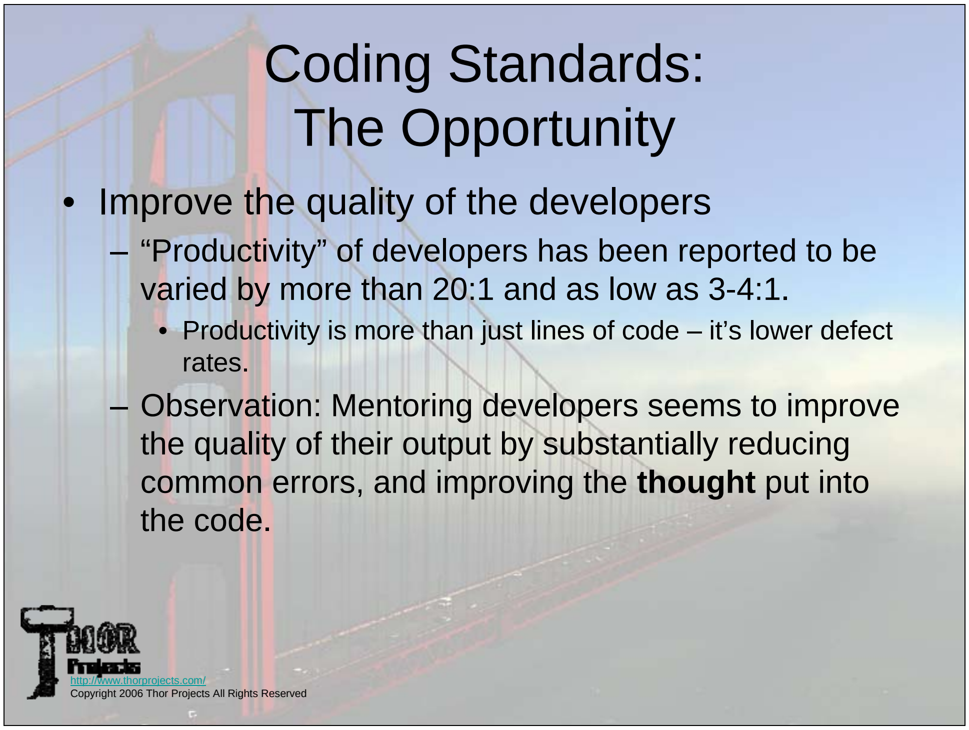


Coding Standards: The Opportunity

- “Studies of individuals have consistently shown variations of 20:1 or more in schedule, cost, and error performance among professional programmers …” – Quality Software Management Vol 1, Jerry Weinberg
- “All other factors being equal, a 90th-percentile team of analysts and programmers will be about four times as productive as a 15th-percentile team.” – Software Engineering Economics, Barry Boehm
- “Two people from the same organization tend to perform alike.” – Peopleware, DeMarco & Lister
- “While this productivity differential among programmers is understandable, there is also a 10 to 1 difference in productivity among software organizations.” – Software Productivity, Harlan Mills

http://www.thorprojects.com/

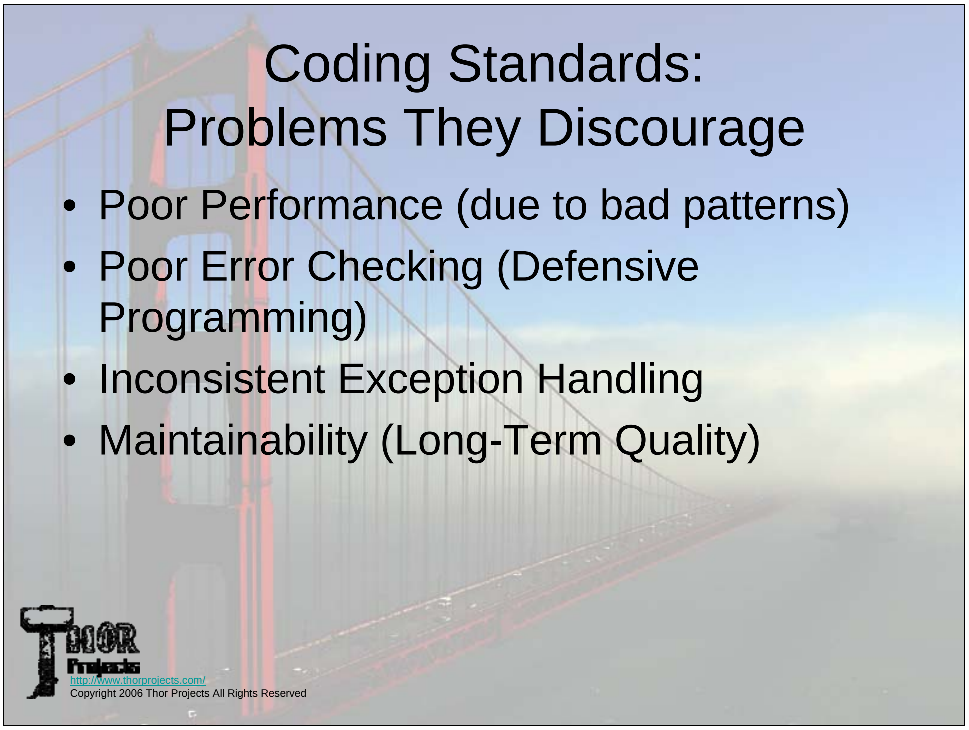
•Individual productivity matters



Coding Standards: The Opportunity

- Improve the quality of the developers
  - “Productivity” of developers has been reported to be varied by more than 20:1 and as low as 3-4:1.
    - Productivity is more than just lines of code – it’s lower defect rates.
  - Observation: Mentoring developers seems to improve the quality of their output by substantially reducing common errors, and improving the **thought** put into the code.

http://www.thorprojects.com/



Coding Standards: Problems They Discourage

- Poor Performance (due to bad patterns)
- Poor Error Checking (Defensive Programming)
- Inconsistent Exception Handling
- Maintainability (Long-Term Quality)

http://www.thorprojects.com/

Code Reviews: Introduction

- Sometimes called…
  - Peer Reviews
  - Software Inspections

http://www.thorprojects.com/

Process Impact (www.processimpact.com) Karl Wiegers Re: Software Requirements

http://www.processimpact.com/articles/two_eyes.pdf

•Inspection

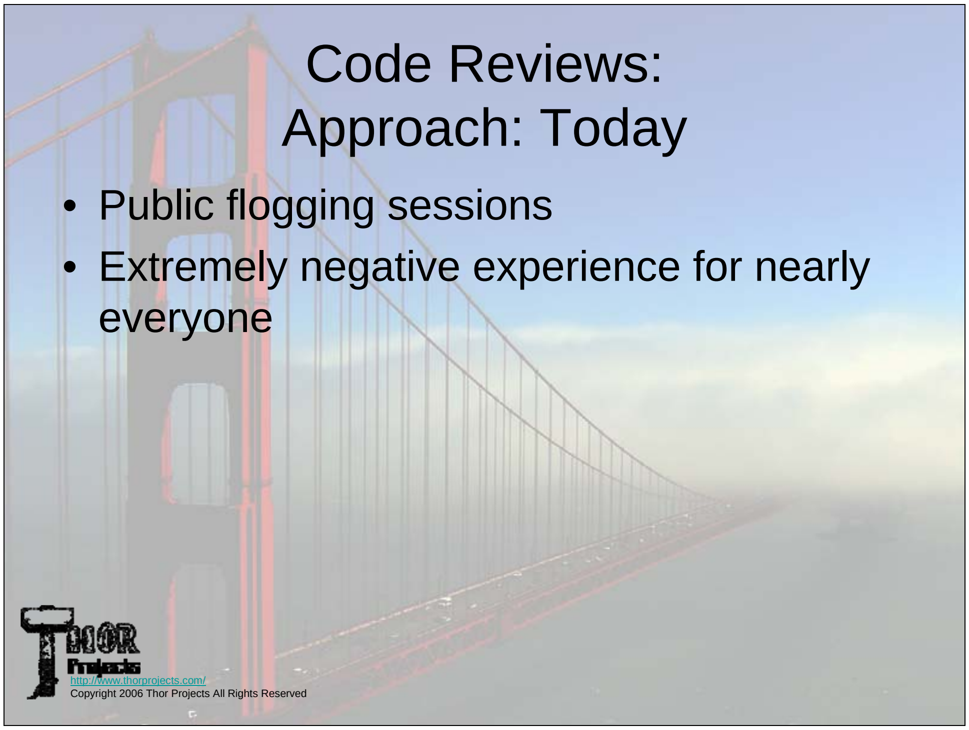
•Team Review

•Walk Through

•Pair Programming

•Peer Deskcheck

•Passaround



Code Reviews: Approach: Today

- Public flogging sessions
- Extremely negative experience for nearly everyone

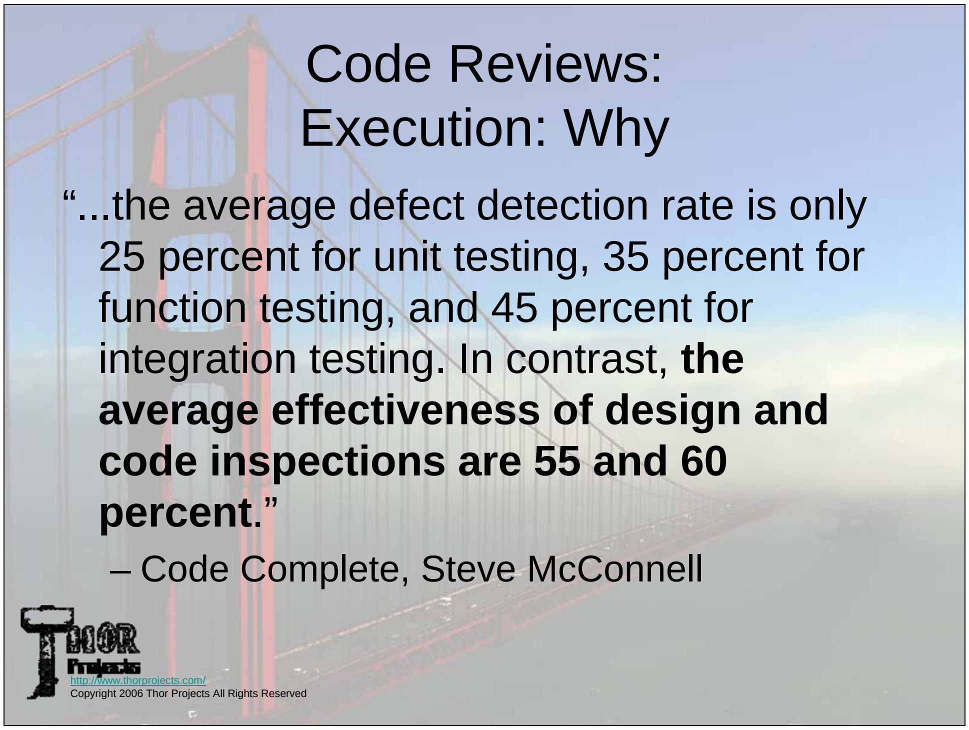
Code Reviews: Approach: Ideal

- Positive Experience
- Learning-based
- Process improvement

http://www.thorprojects.com/

Effective Code Reviews Without the Pain, http://www.developer.com/java/other/article.php/3579756

•Process Improvement – Anything identified in a review which isn’t in the coding standards document should be moved there.



Code Reviews: Execution: Why

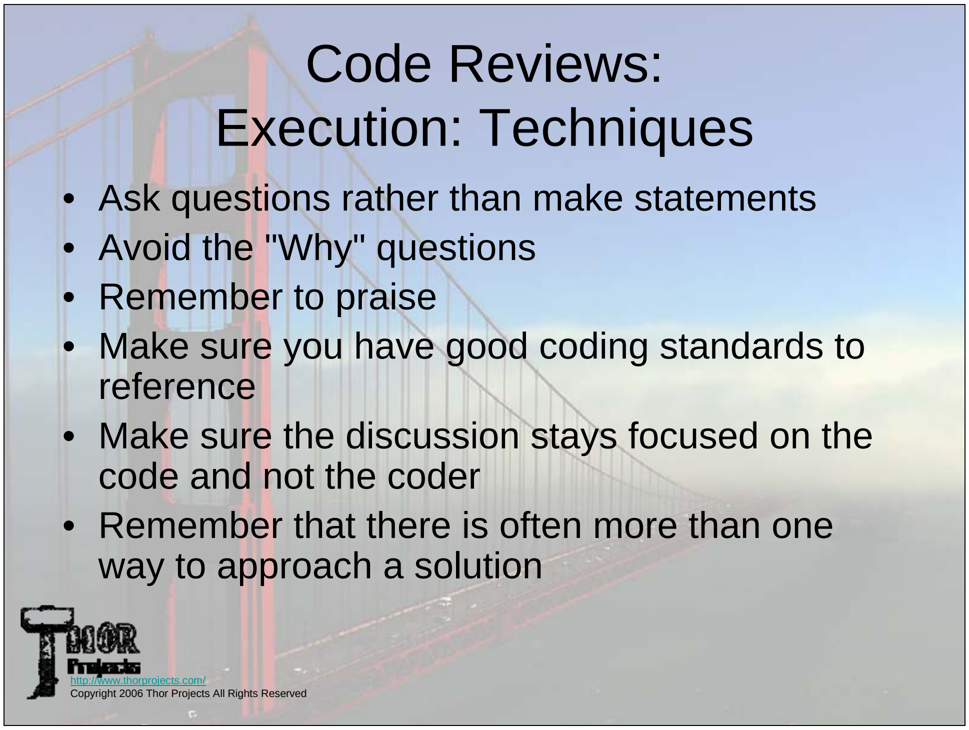
"...the average defect detection rate is only 25 percent for unit testing, 35 percent for function testing, and 45 percent for integration testing. In contrast, **the average effectiveness of design and code inspections are 55 and 60 percent**."

– Code Complete, Steve McConnell

Code Reviews:
Execution: Structure

- Use Proven Techniques
- Create a Lightweight Process
  - Roles
  - Activities
  - Results (Artifacts or Tracking)

THOR
Projects



Code Reviews:
Execution: Techniques

- Ask questions rather than make statements
- Avoid the "Why" questions
- Remember to praise
- Make sure you have good coding standards to reference
- Make sure the discussion stays focused on the code and not the coder
- Remember that there is often more than one way to approach a solution

http://www.thorprojects.com/

Effective Code Reviews Without the Pain, http://www.developer.com/java/other/article.php/3579756

Humanizing Peer Reviews (STQE March/April 2002), Karl Wiegers http://www.processimpact.com

•**Ask questions rather than make statements.** A statement is accusatory. "You didn't follow the standard here" is an attack—whether intentional or not. The question, "What was the reasoning behind the approached you used?" is seeking more information. Obviously, that question can't be said with a sarcastic or condescending tone; but, done correctly, it can often open the developer up to stating their thinking and then asking if there was a better way.

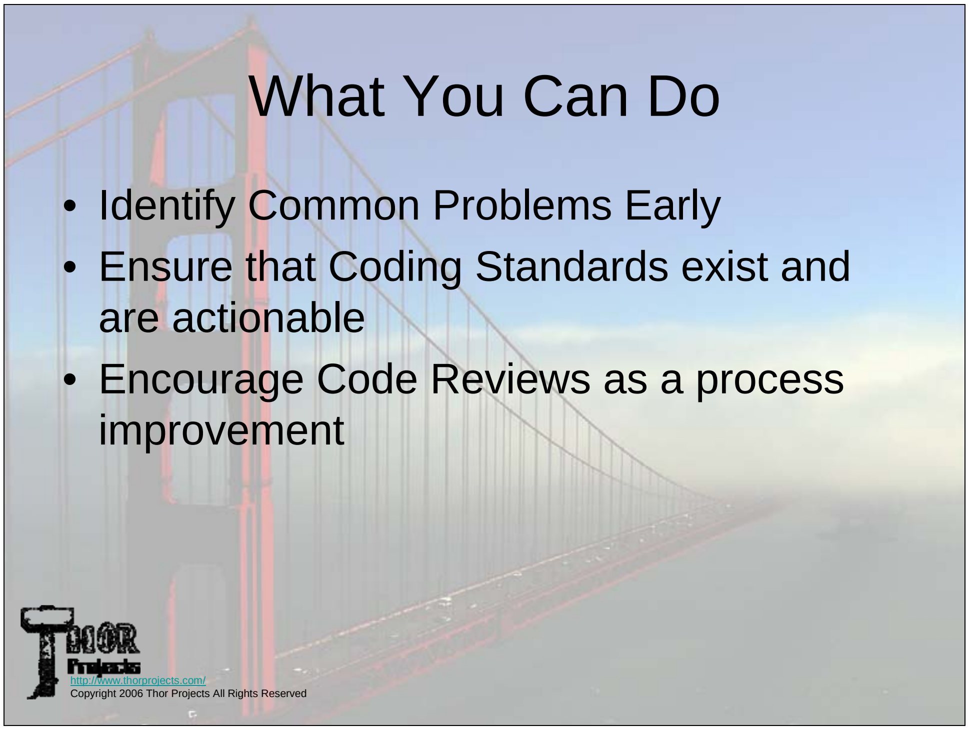
•**Avoid the "Why" questions.** Although extremely difficult at times, avoiding the"Why" questions can substantially improve the mood. Just as a statement is accusatory—so is a why question. Most "Why" questions can be reworded to a question that doesn't include the word "Why" and the results can be dramatic. For example, "Why didn't you follow the standards here..." versus "What was the reasoning behind the deviation from the standards here..."

•**Remember to praise.** The purposes of code reviews are not focused at telling developers how they can improve, and not necessarily that they did a good job. Human nature is such that we want and need to be acknowledged for our successes, not just shown our faults. Because development is necessarily a creative work that developers pour their soul into, it often can be close to their hearts. This makes the need for praise even more critical.

•**Make sure you have good coding standards to reference.** Code reviews find their foundation in the coding standards of the organization. Coding standards are supposed to be the shared agreement that the developers have with one another to produce quality, maintainable code. If you're discussing an item that isn't in your coding standards, you have some work to do to get the item in the coding standards. You should regularly ask yourself whether the item being discussed is in your coding standards.

•**Make sure the discussion stays focused on the code and not the coder.** Staying focused on the code helps keep the process from becoming personal. You're not interested in saying the person is a bad person. Instead, you're looking to generate the best quality code possible.

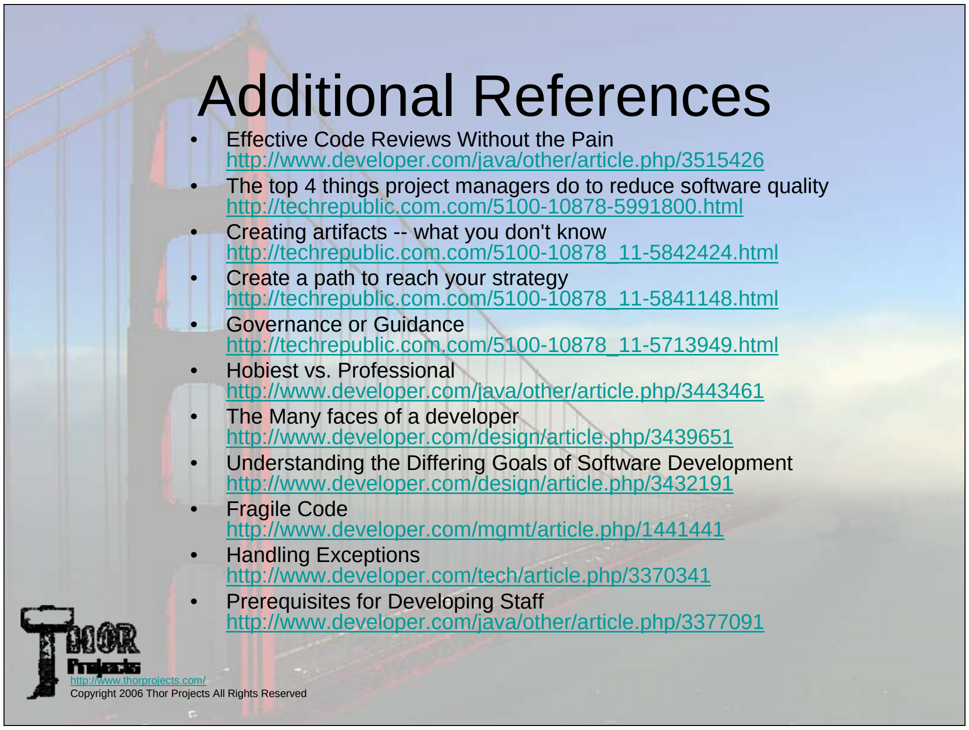
•**Remember that there is often more than one way to approach a solution.** Although the developer might have coded something differently from how you would have, it isn't necessarily wrong. The goal is quality, maintainable code. If it meets those goals and follows the coding standards, that's all you can ask for.



What You Can Do

- Identify Common Problems Early
- Ensure that Coding Standards exist and are actionable
- Encourage Code Reviews as a process improvement

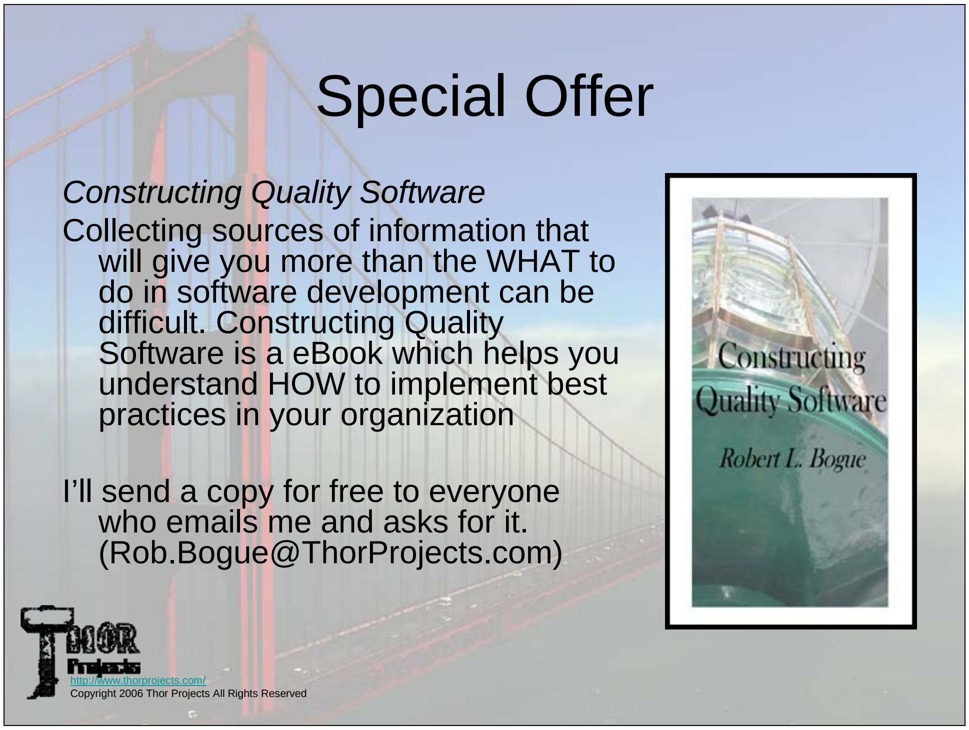
http://www.thorprojects.com/



Additional References

- Effective Code Reviews Without the Pain
  http://www.developer.com/java/other/article.php/3515426
- The top 4 things project managers do to reduce software quality
  http://techrepublic.com.com/5100-10878-5991800.html
- Creating artifacts -- what you don't know
  http://techrepublic.com.com/5100-10878_11-5842424.html
- Create a path to reach your strategy
  http://techrepublic.com.com/5100-10878_11-5841148.html
- Governance or Guidance
  http://techrepublic.com.com/5100-10878_11-5713949.html
- Hobiest vs. Professional
  http://www.developer.com/java/other/article.php/3443461
- The Many faces of a developer
  http://www.developer.com/design/article.php/3439651
- Understanding the Differing Goals of Software Development
  http://www.developer.com/design/article.php/3432191
- Fragile Code
  http://www.developer.com/mgmt/article.php/1441441
- Handling Exceptions
  http://www.developer.com/tech/article.php/3370341
- Prerequisites for Developing Staff
  http://www.developer.com/java/other/article.php/3377091

http://www.thorprojects.com/



Special Offer

Constructing Quality Software

Collecting sources of information that will give you more than the WHAT to do in software development can be difficult. Constructing Quality Software is a eBook which helps you understand HOW to implement best practices in your organization

I'll send a copy for free to everyone who emails me and asks for it. (Rob.Bogue@ThorProjects.com)

http://www.thorprojects.com/