


# Reducing Risk When Changing Legacy Code

---

There are lots of options besides “do more testing”



**Hi, I'm Tina!**  
**I'm a Test Strategist.**





**Reducing Risk: Why I Care**



# What is Legacy Code?

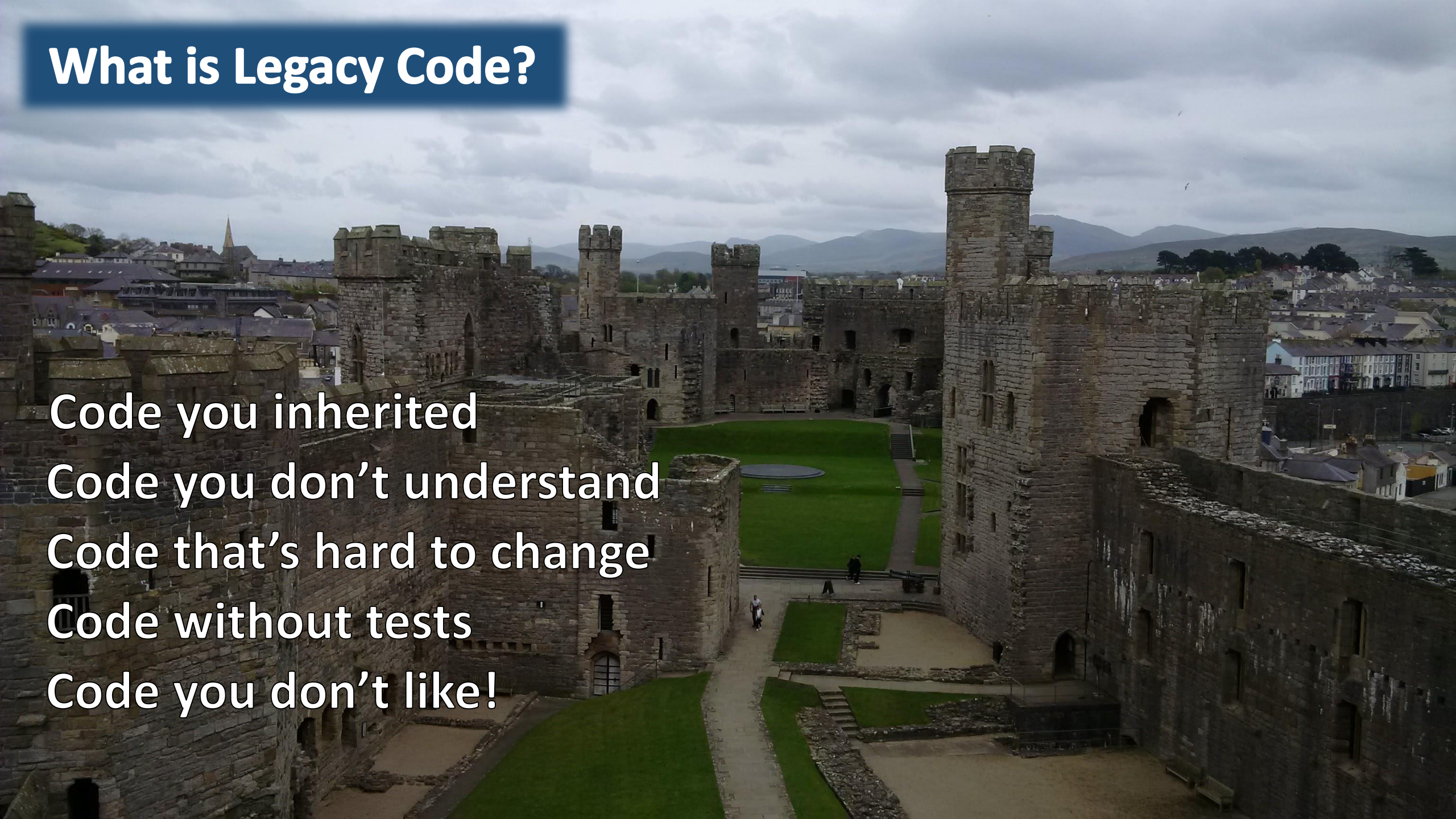
Code you inherited

Code you don't understand

Code that's hard to change

Code without tests

Code you don't like!





# Why is Legacy Code Risky?





A man and a woman are standing in a bar, smiling at the camera. They are both wearing red jackets. The woman is holding a glass of beer, and the man is holding a bottle of beer. Behind them are shelves filled with many different bottles of liquor. A large, ornate chandelier hangs from the ceiling. The text "Step 1: Do More Testing" is overlaid on the image in a blue box.

## Step 1: Do More Testing



USFS

AMC

## BOOTT SPUR LINK

← TUCKERMAN RAVINE TRAIL 0.6

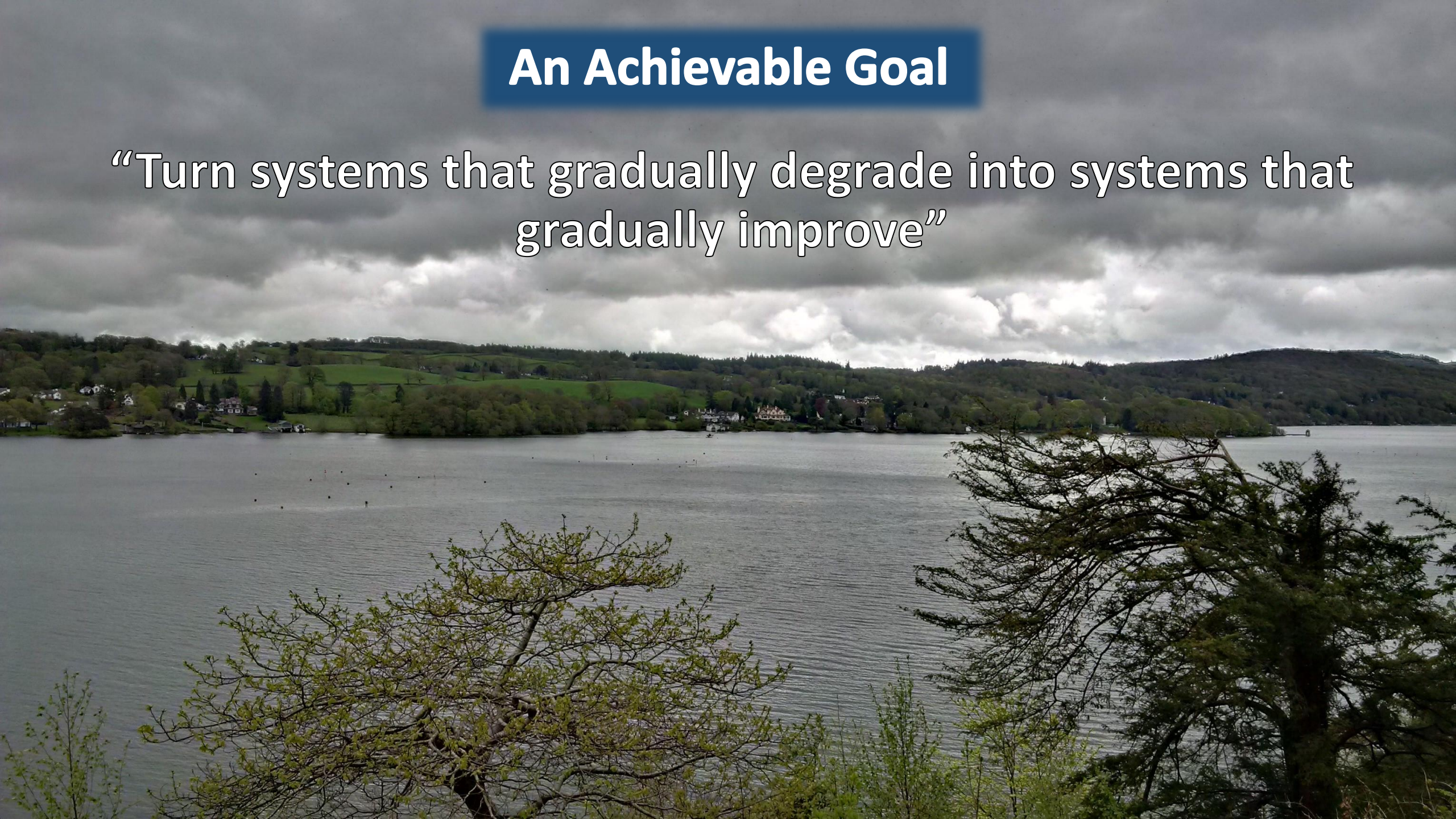
← HERMIT LAKE SHELTERS 0.6

**Step 2: Consider Other Options**



# An Achievable Goal

“Turn systems that gradually degrade into systems that gradually improve”





**Code Stewardship**  
**Knowledge Management**  
**Test Coverage Analysis**  
**Team Shadowing**  
**Culture Change**

# **Reducing Risk When Changing Legacy Code: 5 Ideas**



**Code Stewardship**  
**Knowledge Management**  
**Test Coverage Analysis**  
**Team Shadowing**  
**Culture Change**

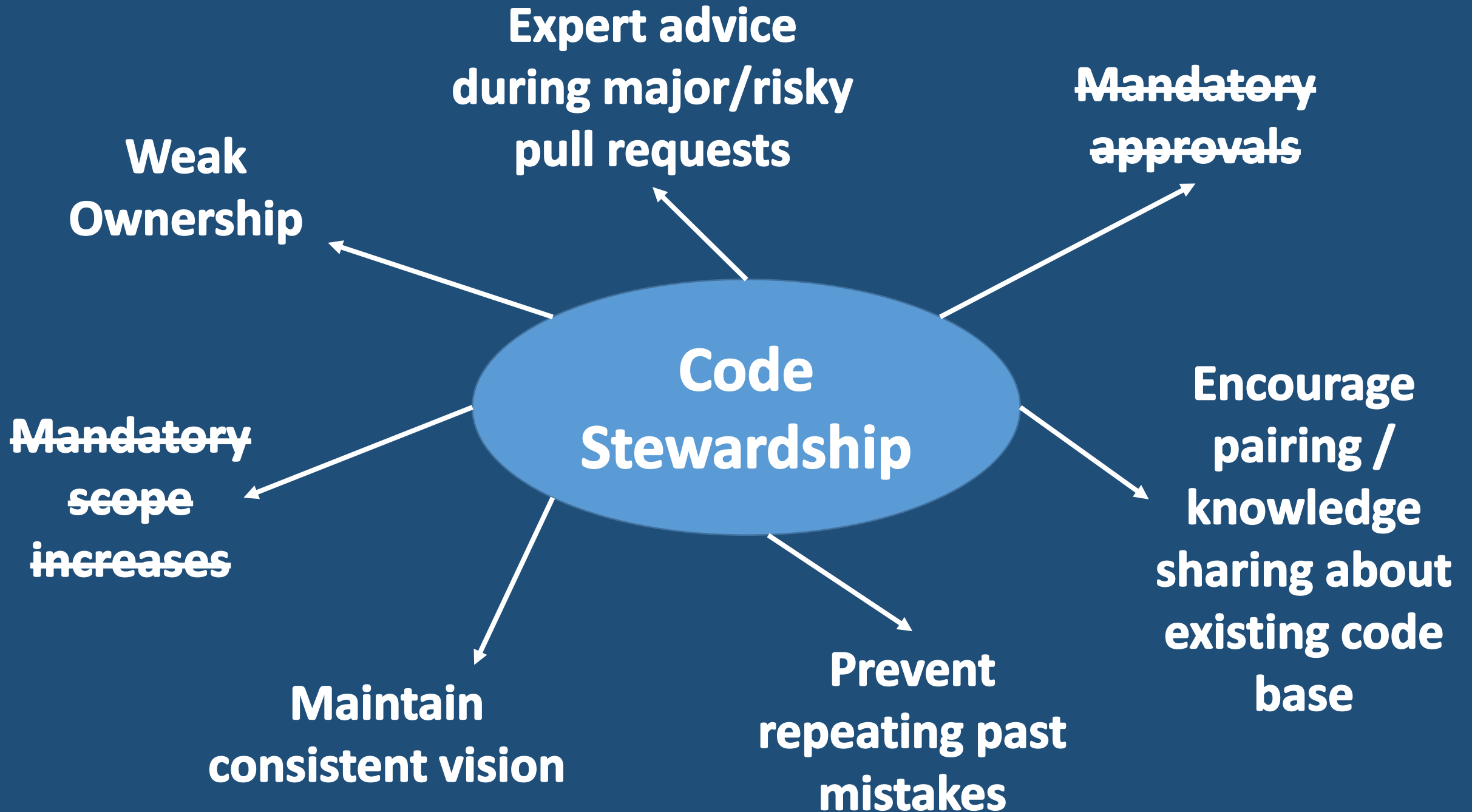
# Reducing Risk When Changing Legacy Code: Idea #1



**“Module owners are expected to take responsibility for the modules they own and keep an eye on changes made by other people.”**


*Martin Fowler*







## Edit Pull Request

Destination  master ▾

Update the destination branch for the pull request

Title \* Calexander/MultiGetSupport

Description This also broke some stuff in LE. Companion PR:  
<https://git.dev.d2l/projects/CORE/repos/le/pull-requests/4655/overview>



Preview

Reviewers

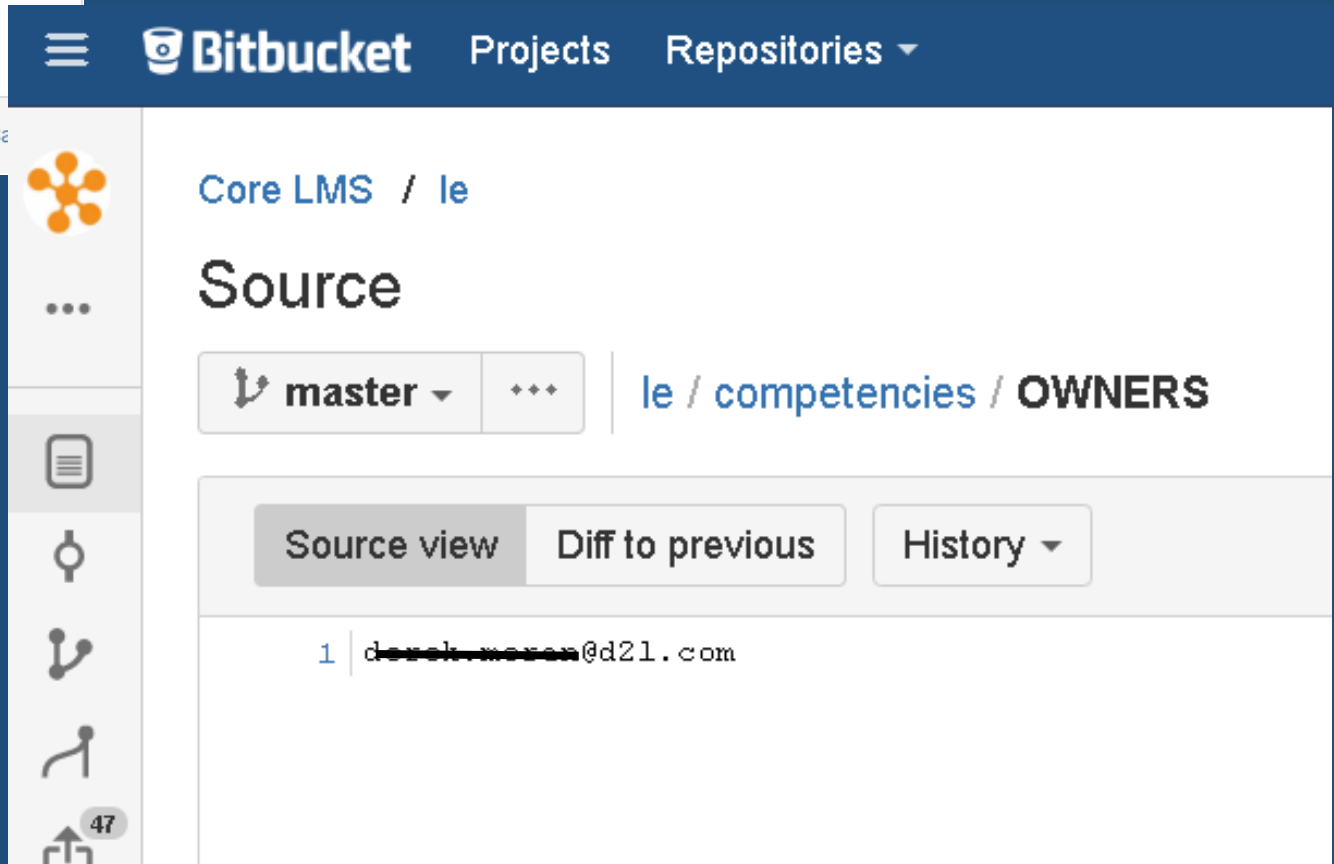
Suggestions: Dave Lockhart, Daryl McMillan

Save

Ca

# Open source plugin for Bitbucket Native solution for Github

Makes finding experts  
easier and safer



The image shows the Bitbucket web interface. At the top is a dark blue navigation bar with the Bitbucket logo, 'Projects', and 'Repositories' with a dropdown arrow. Below this is a sidebar with icons for repository actions. The main content area shows the 'Source' view for the 'le' repository under 'Core LMS'. It includes a 'master' branch selector, a breadcrumb 'le / competencies / OWNERS', and tabs for 'Source view', 'Diff to previous', and 'History'. The code view shows a single line: '1 | davek.warren@d2l.com'. A sidebar on the left contains icons for repository management, including a commit icon with a '47' badge.





Actively recruit stewards


Use data to identify candidates



Core LMS / Ip

## Source

 master ▾  Ip / framework / tools / D2L.LP.Tools.UnitTests / Profile / ProfileCardManagement / **Domain** /

---

 ..

 OWNERS	Proposing Justin as owner for Profile Card Management unit tests
 ProfileCardManagementSettingsManagerTest.cs	DE25504: Remove local const

Low vs high quality owners

Limit scope, focus on risk areas



# **I Want To Try Code Stewardship!**

- ✓ **Find a way to suggest stewards as reviewers**
- ✓ **Prioritize by identifying risk areas**
- ✓ **Some owners are better than none**



**Code Stewardship**  
**Knowledge Management**  
**Test Coverage Analysis**  
**Team Shadowing**  
**Culture Change**

# Reducing Risk When Changing Legacy Code: Idea #2



**“In nearly every legacy system, what the system does is more important than what it is supposed to do.”**

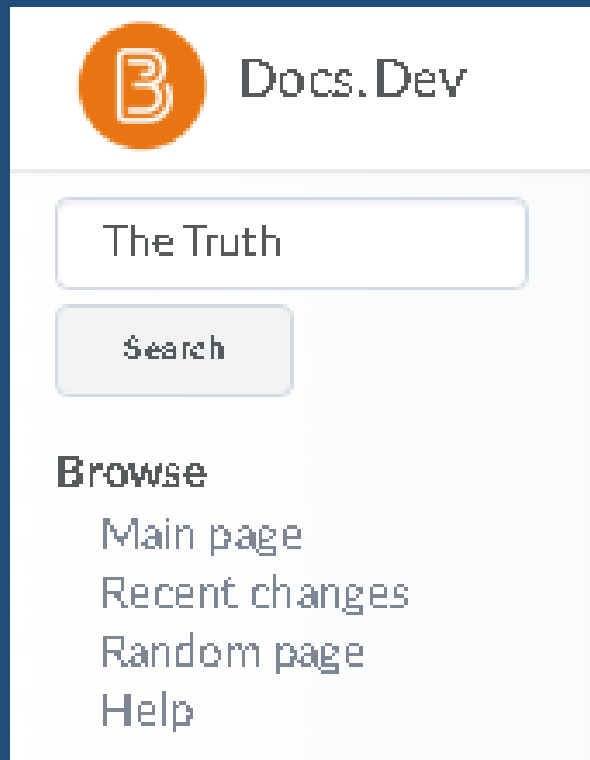
*Michael C. Feathers*







# Don't expect perfection



## Do lots of things

Contents [hide]

- 1 Overview
- 2 Architecture Diagram
- 3 Technical Details
- 4 Setup
  - 4.1 Production setup
  - 4.2 Local Setup
- 5 Feature Flags
- 6 Monitoring
  - 6.1 Health checks
  - 6.2 Metrics
- 7 Scalability
- 8 Demos
- 9 Useful Links
- 10 HELP EVERYTHING IS BROKEN AND IT'S YOUR FAULT

## Consider discoverability, searchability, and maintainability

# **I Want To Try Knowledge Management!**

- ✓ **Document important information**
- ✓ **Share knowledge in lots of ways**
- ✓ **Supplement with human interaction**



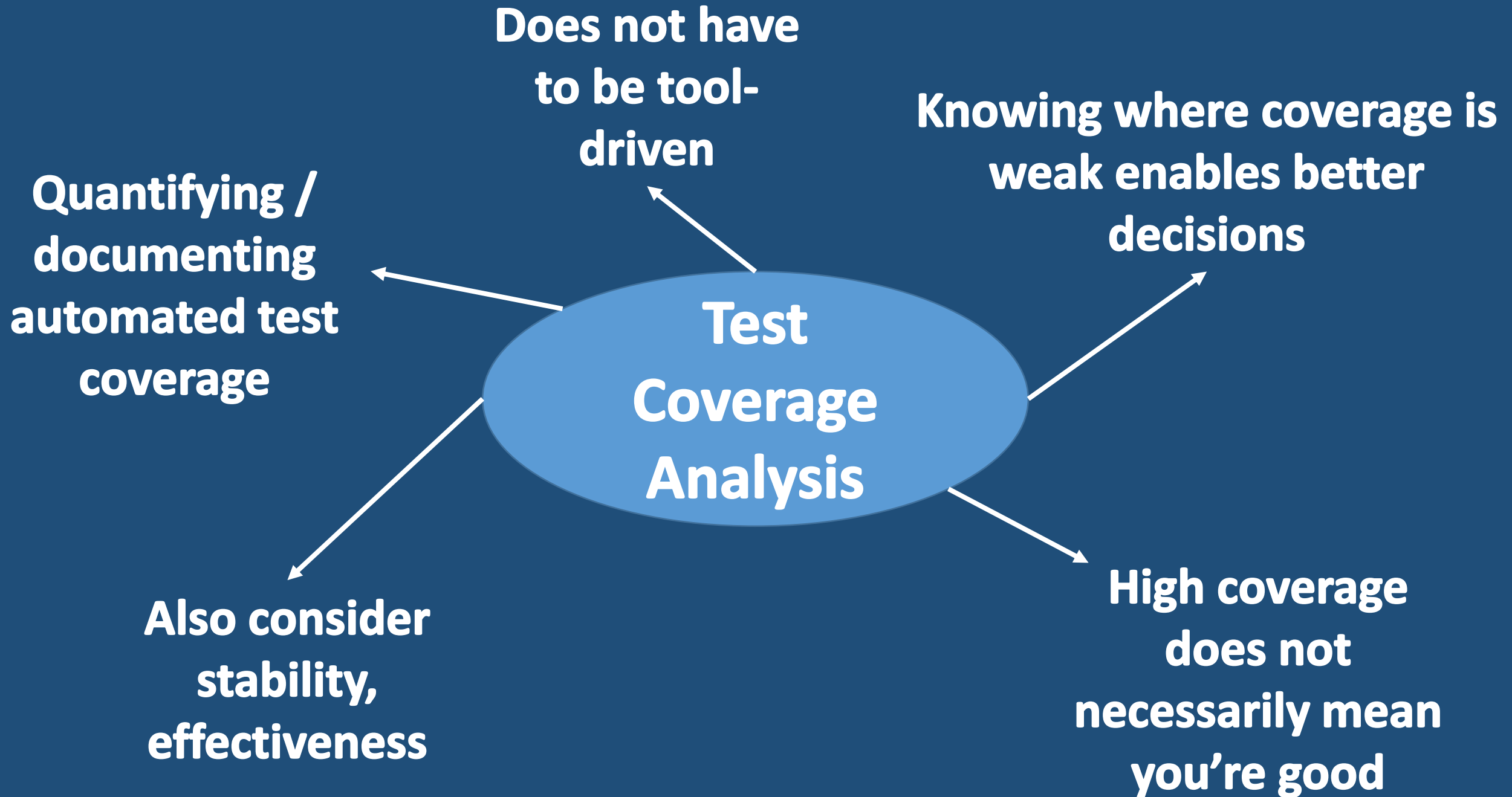
**Code Stewardship**  
**Knowledge Management**  
**Test Coverage Analysis**  
**Team Shadowing**  
**Culture Change**

# **Reducing Risk When Changing Legacy Code: Idea #3**

**“Test coverage is a useful tool for finding  
untested parts of a codebase.  
Test coverage is of little use as a numeric  
statement of how good your tests are.”**

*Martin Fowler*





## SYSTEM LEVEL AUTOMATED TESTS

What Do We Actually Want?

D2L.com

Conversation  
starter, not to-  
do list

### 4. Self Aware Automation

- Tests have proven ability to detect regressions (e.g. through mutation testing or another strategy)
- Problematic tests automatically removed from suite until they can be repaired, and new tests have to demonstrate stability/efficacy before being added to pipeline
- Test failures automatically notify relevant stakeholders (original test author, most recent test editor, teams dependent on the related functionality) with specific and actionable error messages

### 3. Resilient Automation

- Tests pass reliably, regardless of product configuration or sequence of tests.
- Flaky tests are removed manually from test suites, and re-introduced after improvements are made
- $\geq 90\%$  of test failures are due to product issues (i.e., not test/environment/framework problems)
- Test coverage is  $\geq 80\%$
- Test execution time is  $\leq 10$  mins for commit build/unit tests, and  $\leq 1$  hour for everything else in the CD pipeline (integration and system level tests)

### 2. Automatic Automation

- Build pipeline includes: automated test execution, execution time measurement, and test coverage measurement
- Test coverage trend is increasing (i.e., keeping pace with adding new tests as new code is written, plus adding tests in previously un-covered areas); or,  $\geq 80\%$  overall

### 1. Manual Automation

- Automated tests exist, but do not run as part of build process (e.g. no CI infrastructure, or tests take too long)
- Test coverage data and recent automated test results can be provided on demand
- Test coverage is not decreasing (i.e., keeping pace with adding new tests as new code is written)

### 0. No Automation

- No automated tests exist

Automation  
maturity is a lot  
more than a  
number



# I Want To Try Test Coverage Analysis!

- ✓ Decide what you want out of it
- ✓ Start with something cheap
- ✓ Focus on knowledge rather than change

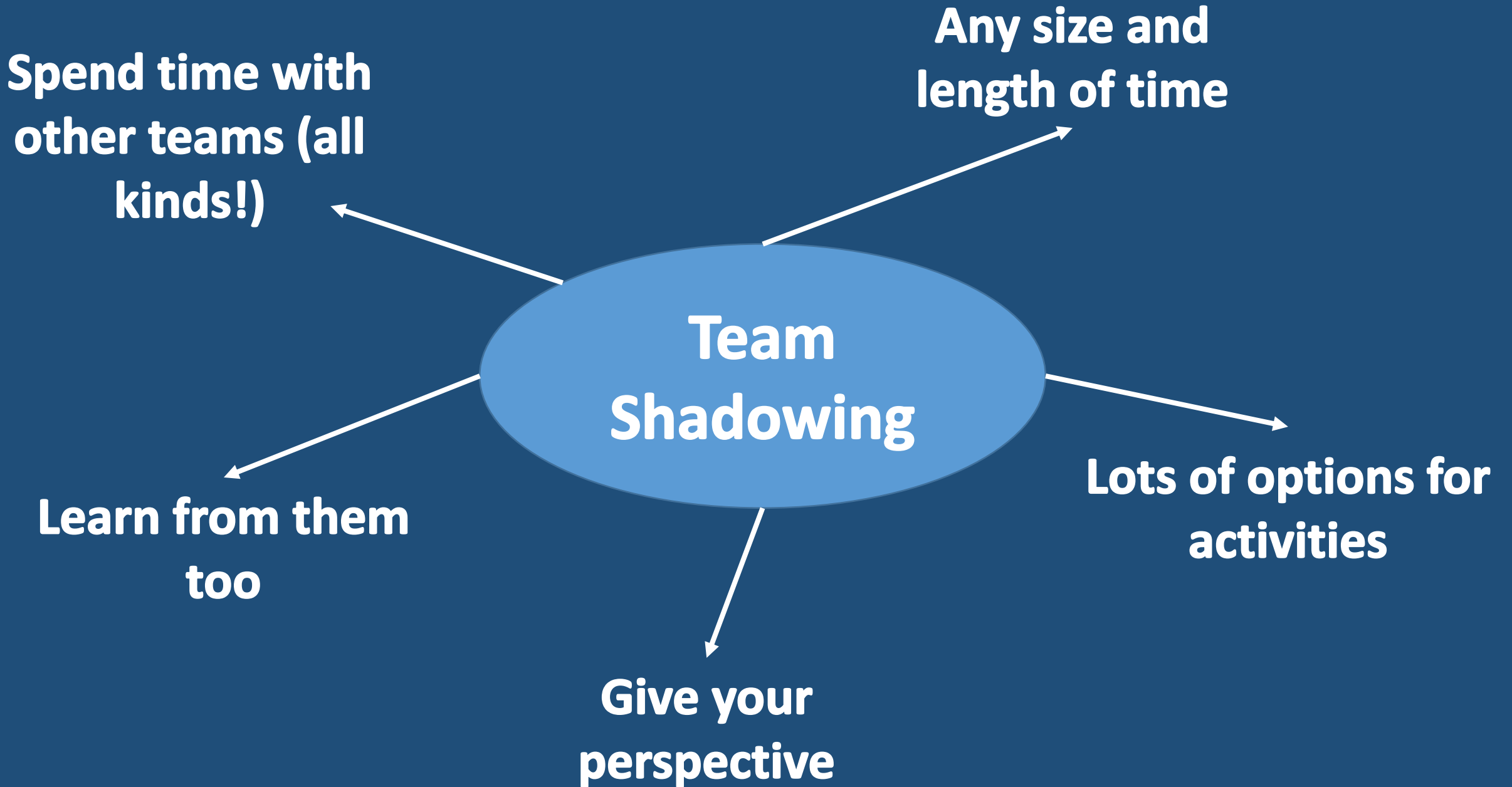
**Code Stewardship**  
**Knowledge Management**  
**Test Coverage Analysis**  
**Team Shadowing**  
**Culture Change**

# **Reducing Risk When Changing Legacy Code: Idea #4**



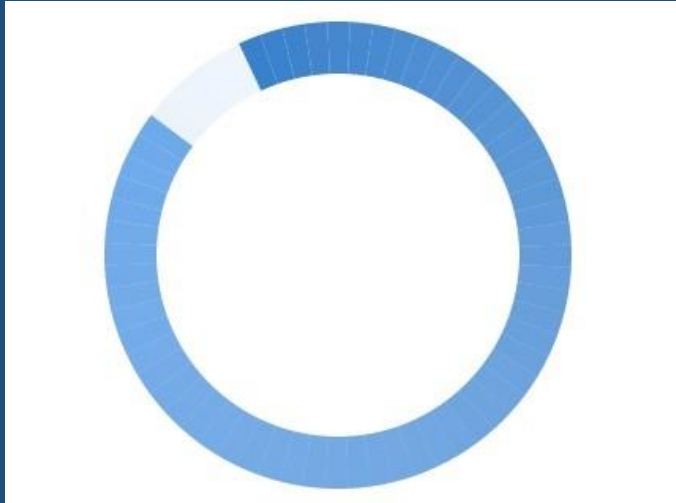
**“I'm surprised every single time by what I learn, and can then bring back to my 'home' team - even if all I'm doing is attending another team's stand-up.”**

*Me*





**Prevents re-  
inventing the wheel**



**Prevents pulling  
in incompatible  
directions**



**Promotes awareness of other  
potential stakeholders**

# **I Want To Try Team Shadowing!**

- ✓ **Identify teams you don't know well**
- ✓ **Ask if you can listen in on some meetings**
- ✓ **Invite others to shadow your team**

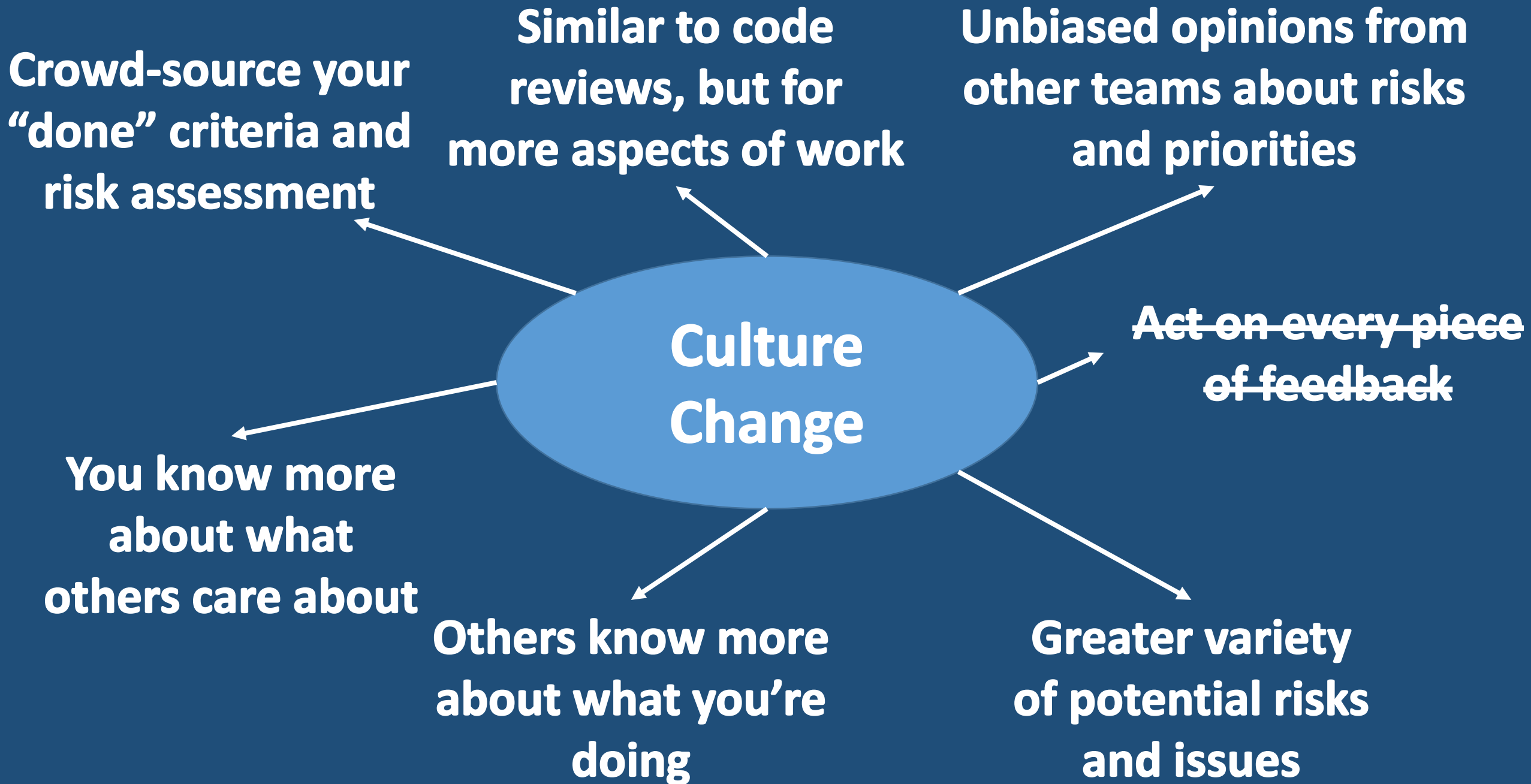


Code Stewardship  
Knowledge Management  
Test Coverage Analysis  
Team Shadowing  
Culture Change

# Reducing Risk When Changing Legacy Code: Idea #5

**“How do we start consistently acting like we value quality above deadlines instead of just saying that we do?”**

*James Spere*







**Get feedback from  
many sources**

**Highlight most  
controversial bits**

**Make it easy to  
give feedback**

**Show that input is  
taken seriously**

**Celebrate valuing  
quality over deadlines**



# **I Want To Try Culture Change!**

- ✓ **Make a list of stakeholders**
- ✓ **Prefer informal / voluntary over mandated**
- ✓ **Find ways to celebrate**

**Code Stewardship**  
**Knowledge Management**  
**Test Coverage Analysis**  
**Team Shadowing**  
**Culture Change**

# **Reducing Risk When Changing Legacy Code: 5 Ideas**



A scenic view of a mountain range during autumn. The foreground is filled with trees showing vibrant yellow and orange foliage. In the background, misty, layered mountain peaks are visible under a grey, overcast sky. The overall atmosphere is serene and slightly somber due to the weather.

**Changing Legacy Code is Risky**



# Use Many Strategies to Reduce Risk







**Start Somewhere, Start Small**





**Measure and Adjust**





**The Saranac Lake 6er Bell**

Only those who finish climbing the 6er peaks—McKenzie Mt., Ampersand Mt., Scarface Mt., St. Regis Mt., Haystack Mt., and Mt. Baker—may ring this bell. Tradition has it that each finisher shall ring the bell 6 times, once for each peak, and good luck shall go to the ringer.

**THE KIWASSA CURSE**  
Beware, for those who ring the bell—even once—having not climbed all 6er mountains, good luck is not guaranteed as the Kiwassa Curse is upon all pretenders to the title and honor of Saranac Lake 6er.

Set an Achievable Goal



Constructive  
debates

Are We There Yet?



Projects actually  
finish

Specific risks  
understood

Engaged team



**Thank You**



**@fletchertinam**



# References

- **Martin Fowler - Code Ownership**
  - <https://martinfowler.com/bliki/CodeOwnership.html>
- **Martin Fowler - Test Coverage**
  - <https://martinfowler.com/bliki/TestCoverage.html>
- **James Spere - Valuing Learning**
  - <https://www.linkedin.com/pulse/valuing-learning-james-spere>
- **Chromium - Code Stewardship**
  - [https://chromium.googlesource.com/chromium/src/+/\\_master/docs/code\\_reviews.md](https://chromium.googlesource.com/chromium/src/+/_master/docs/code_reviews.md)
- **Agile @ Lego - Alignment Day**
  - <https://crisp.se/wp-content/uploads/2016/12/Agile@Lego.pdf>
- **Working Effectively With Legacy Code** (Michael C. Feathers)
- **How to Measure Anything: Finding the Value of “Intangibles” in Business, Third Ed.** (Douglas W. Hubbard)