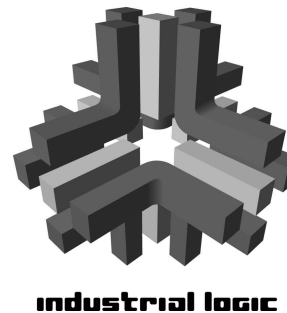


10 Tips for Successful Agile Transitions

Joshua Kerievsky
joshua@industriallogic.com
QCON, San Francisco
November, 2007



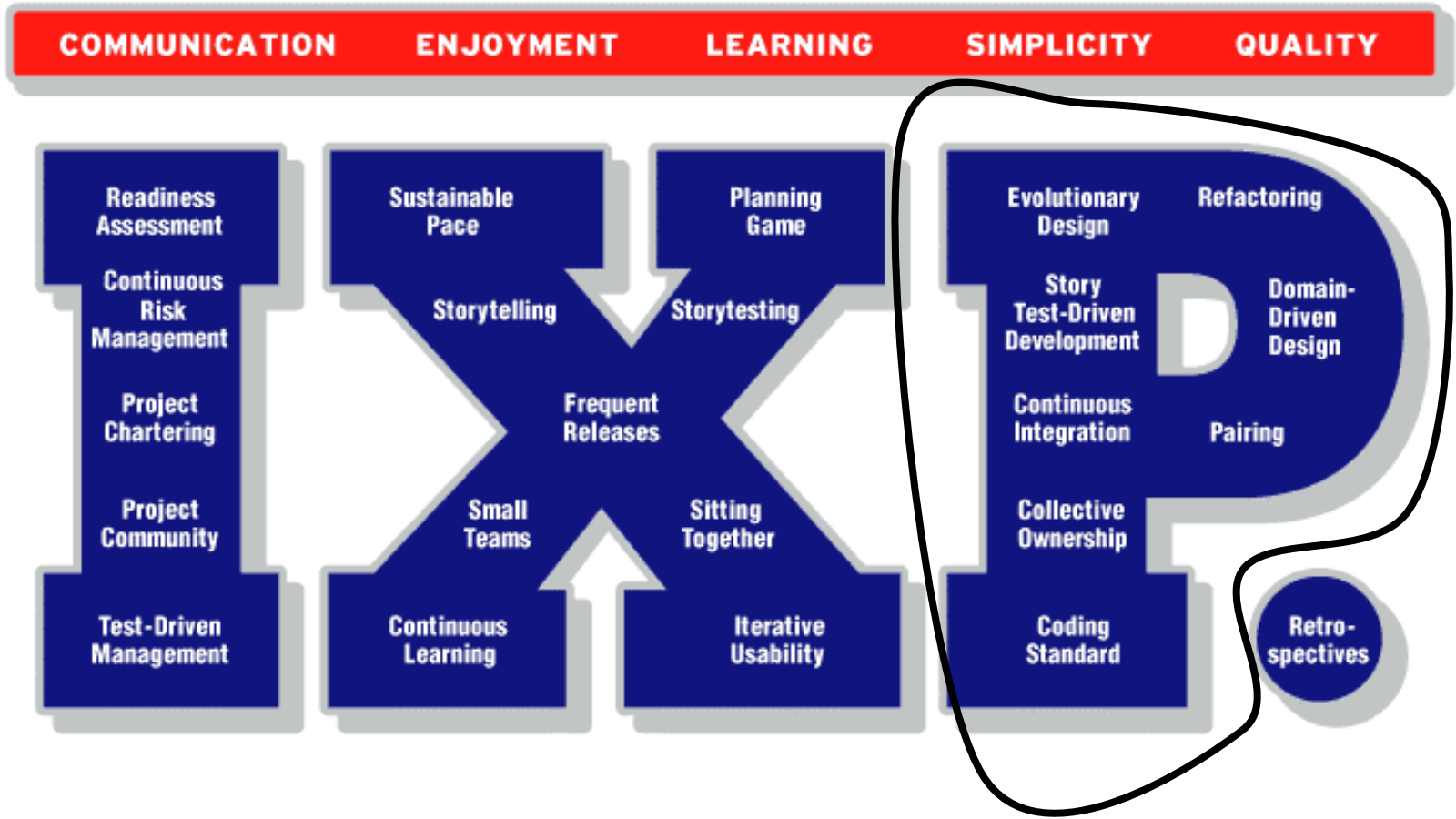
Copyright 2007, Industrial Logic, Inc.
All Rights Reserved.



**Tip #1: Begin All
Transitions with a
Readiness Assessment**

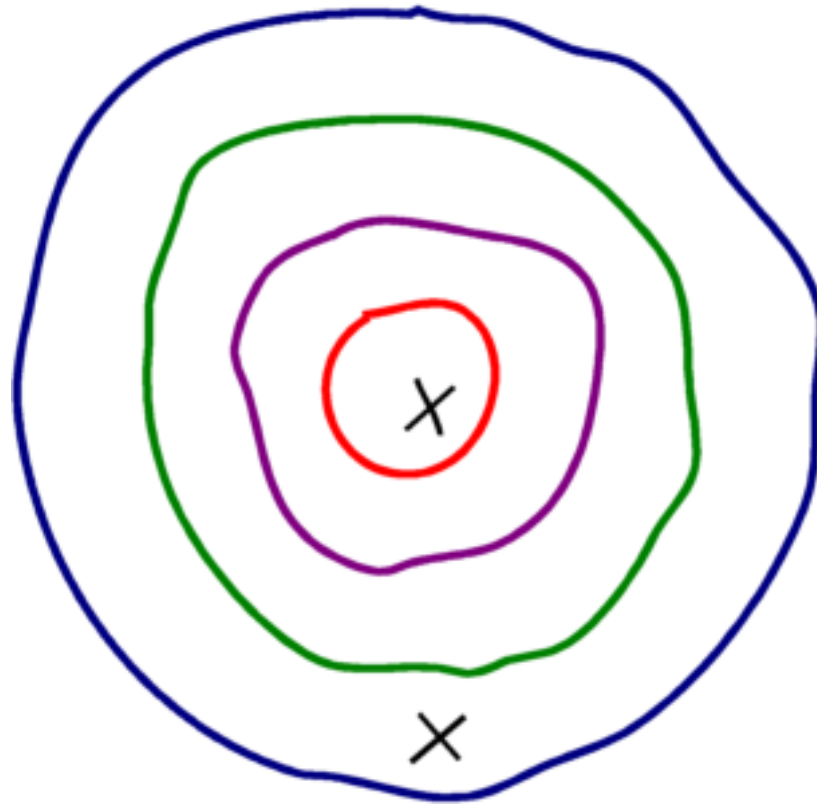
**Tip #2 – Leverage
The Power Of
Non-Technical AND
Technical Practices**





Copyright 2007, Industrial Logic, Inc.
 All Rights Reserved.

Huh?!?!



Copyright 2007, Industrial Logic, Inc.
All Rights Reserved.



**Tip #3: Evolve By
Learning A Little Of
Everything**



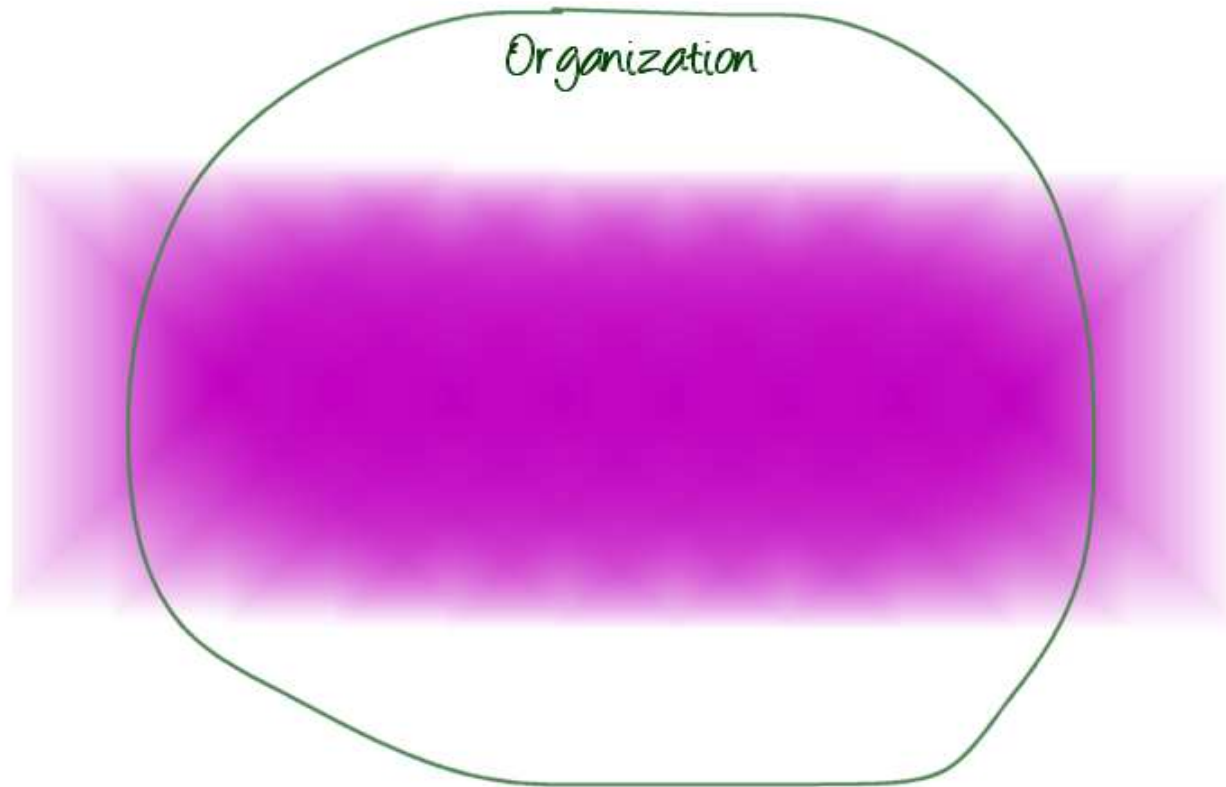
Copyright 2007, Industrial Logic, Inc.
All Rights Reserved.

Whole Enchilada Transition

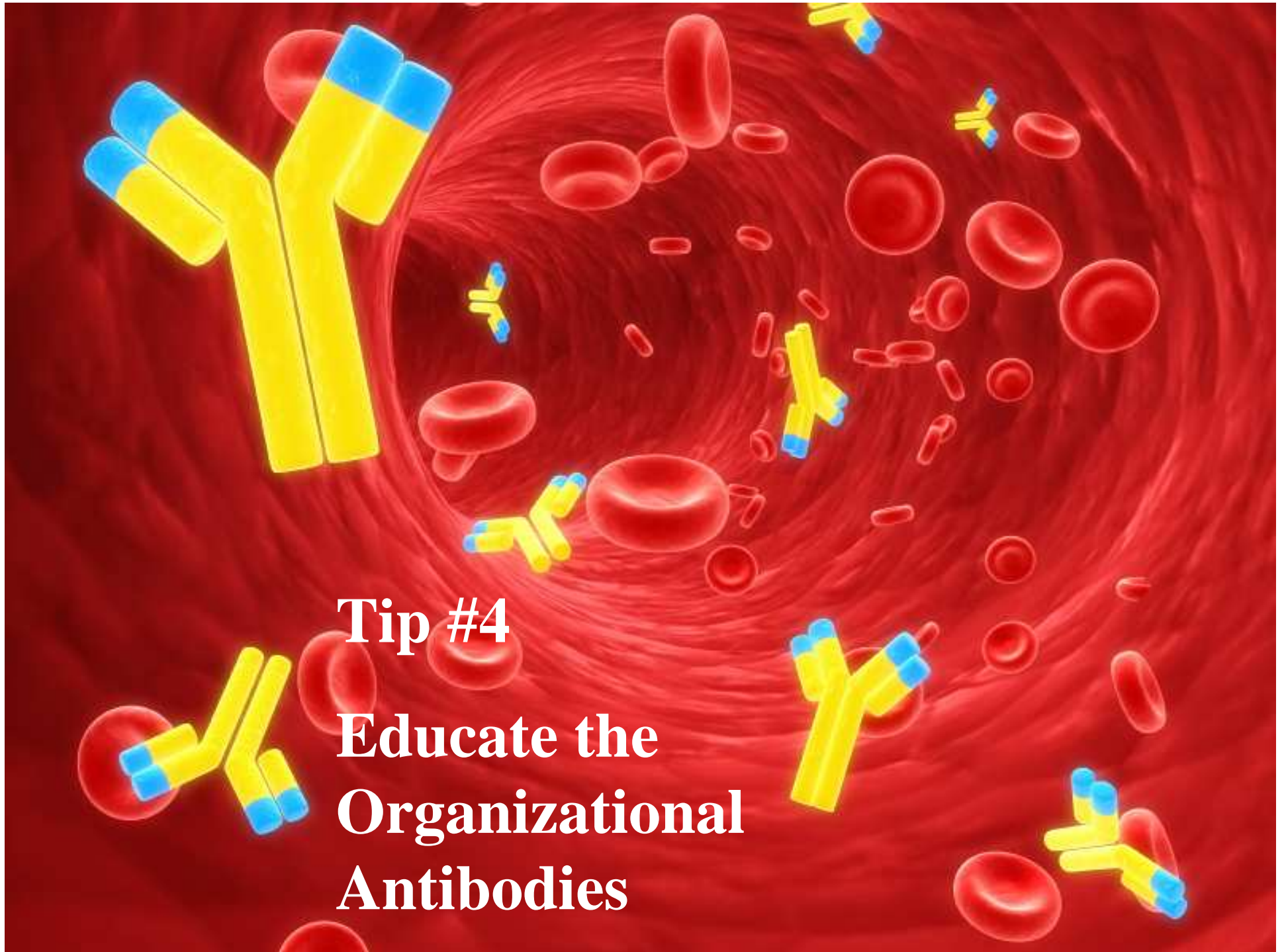


Copyright 2007, Industrial Logic, Inc.
All Rights Reserved.

Broad Brushstroke Transition



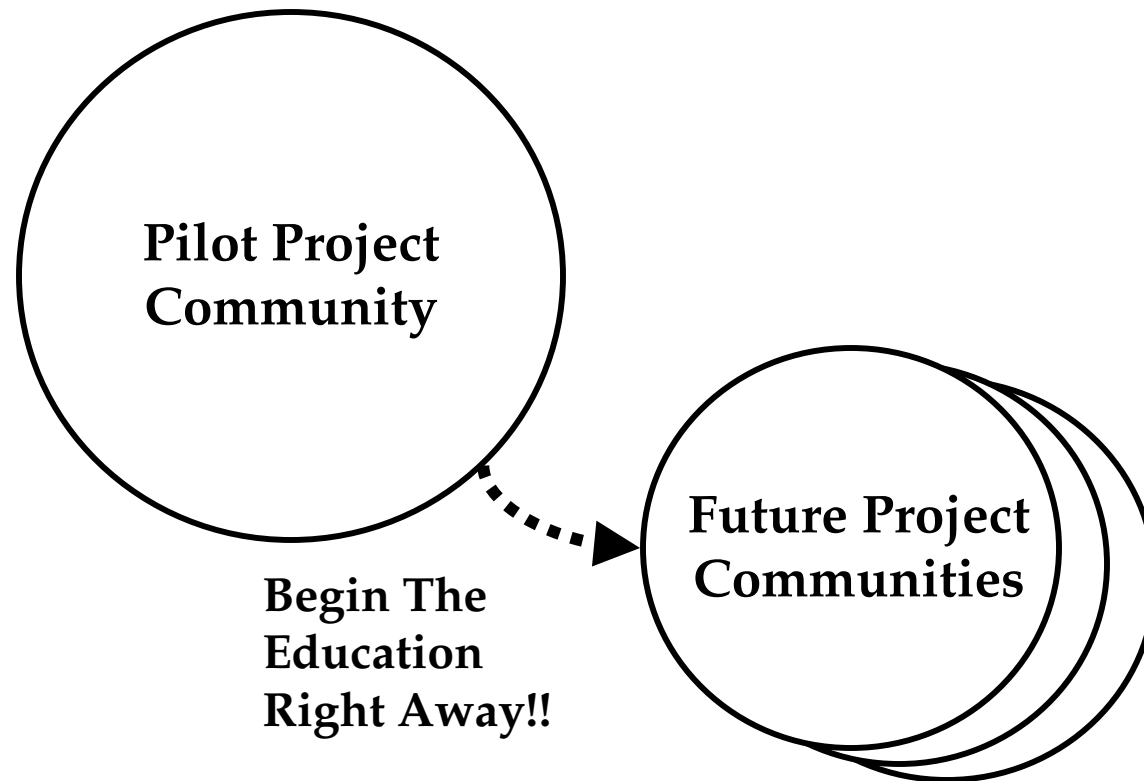
Copyright 2007, Industrial Logic, Inc.
All Rights Reserved.



Tip #4

**Educate the
Organizational
Antibodies**

Educating The Antibodies



How Would You Sell Pair-Programming?

- Don't sell it!
- Sell the risks of Solo Programming
 - Tunnel Vision (Frog-In-A-Well) Syndrome
 - Less Productivity
 - Less Knowledge Transfer
 - Longer Times Fixing Defects
 - Less Code Re-Use
 - Poorer testing
 - Etc..

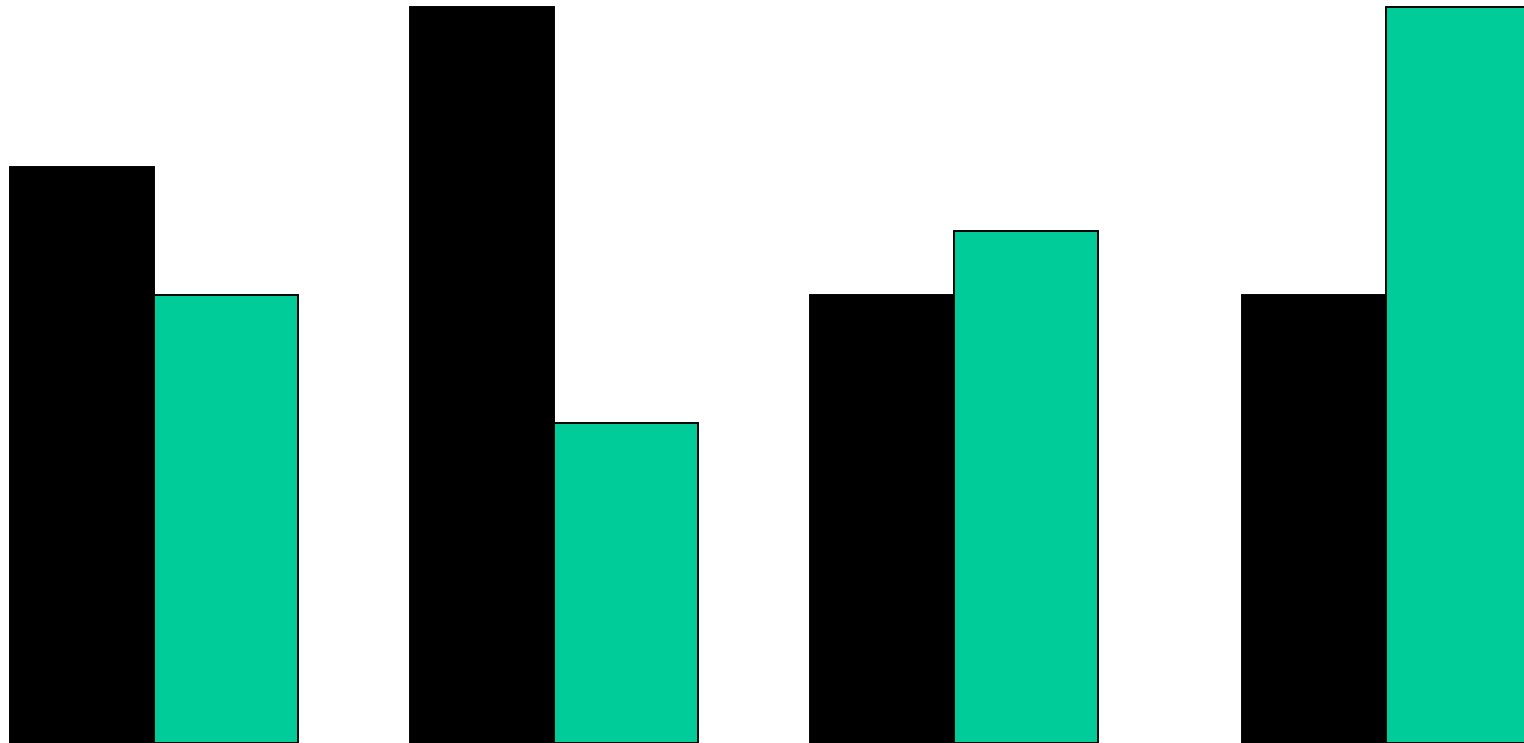
Removing Risks



Copyright 2007, Industrial Logic, Inc.
All Rights Reserved.

Tip #5 Business Trumps Process

(most, but not all of the time)



Copyright 2007, Industrial Logic, Inc.
All Rights Reserved.

Tip #6 Engage The Entire Organization

Management

Executives & Board Members,
Product & Product Line
Management, Sales & Marketing
Management, Project & Functional
Management, QA Management

Support

Coaches, Software Services,
Facilities, Human Resources,
Legal, Auditors, Finance,
3rd party vendors

Customers

Product and/or Project Manager,
Domain Experts , Analysts (market,
technical, etc.), Testers/QA,
End Users / Beta Sites,
Sales, Marketing & Service Support,
Usability Experts, Technical
Support, Educational Services

Development

Programmers, Technical writers,
Database Designers &
Administrators, Architects

PANEL A

PROJECT COMMUNITY

Customers

- Julie - Product Manager - Hardware
- Lorne - Lead/Application Specialist
- Nadia - MET ID application Specialist
- Nick - Software Product Manager
- Gary Impey - Technical Marketing
- Byron - Product Manager/Marketing
- Dan - Applied Biosystems Software Manager
- Carmoi - real-world user
- Tom (REMO, CARMEN) - user experience
- Shengping - application research

Developers

- Chris - Frameworks/Emergent Coach
- Rob - Frameworks
- Ray - Frameworks
- Shengping - MET ID Developer
- Sidonia - Application Developer
- Hai - Application Developer
- Jim Bohannon - BIOANALYST Developer
- Friedrich - Liaison to ANALYST

More Customers

- Fiona - Product Manager - Hardware
- Tamara - Product Manager - Hardware
- Customer - Christine Dieckman/Merck
- Customer - Gerard Hopfgartner/Univ of Geneva
- Eliot
- Tonia
- Louissetta
- alisia

CONTENT

DIRECTIONAL ADVISORS

COACHES

- JIMS.
- GUNJAN
- JOSH

IT Support

- Heather R.
- Paul
- Victor

PANEL I

PROJECT COMMUNITY

GOLDOWNERS

- KAMRAN - Dir - Quality Assessment + Regulatory Affairs
- ANDREW - Product Test Manager
- JOE - JOINT VENTURE DIRECTOR
- KEV - THE MAAD - Director Product Development
- JOHN - Applied Bio - Software Manager
- SCOTT - Software Manager
- STEVE - Application Specialist
- HEATHERS - Technical Writing/user Experience Manager
- PIERRE - VP, Operations

More Customers

- Yvet - Research Scientist
- Eva - Research Scientist
- Jeff - Metabolism Specialist
- Tsai - Metabolism Specialist

TECH WRITING

- LAURENE - writer

JANET - team lead

FACILITIES

- BRIAN

PRODUCT TEST

- LEANNE - tester

MANAGERS

- RON - Project Manager
- DALIA - Software Lead/Emergent Coach

SOFTWARE QUALITY ASSURANCE

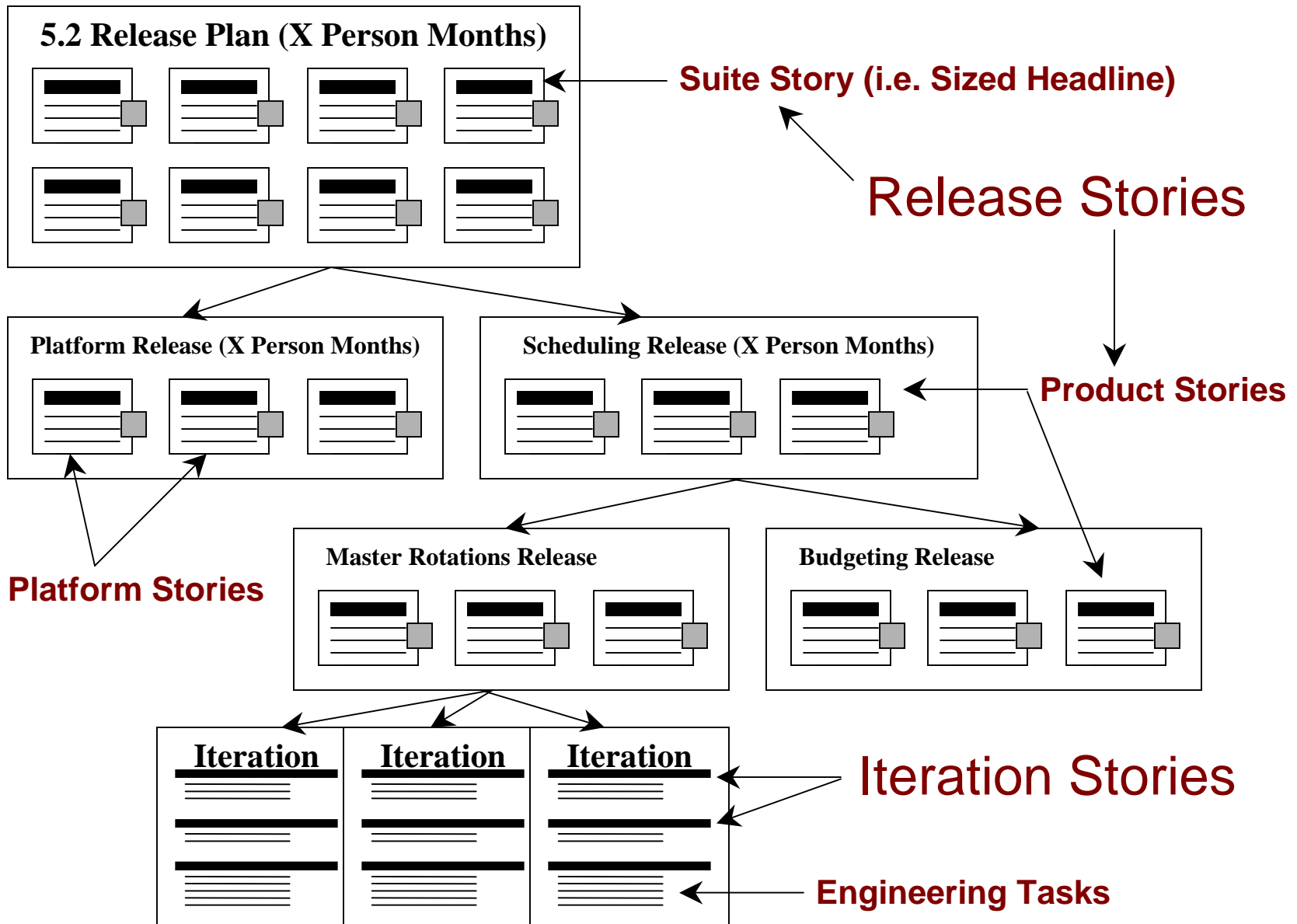
- HARVEY

CUSTOMER SOFTWARE SUPPORT

- Crystal

JIGGLER

III





Marketing

VP

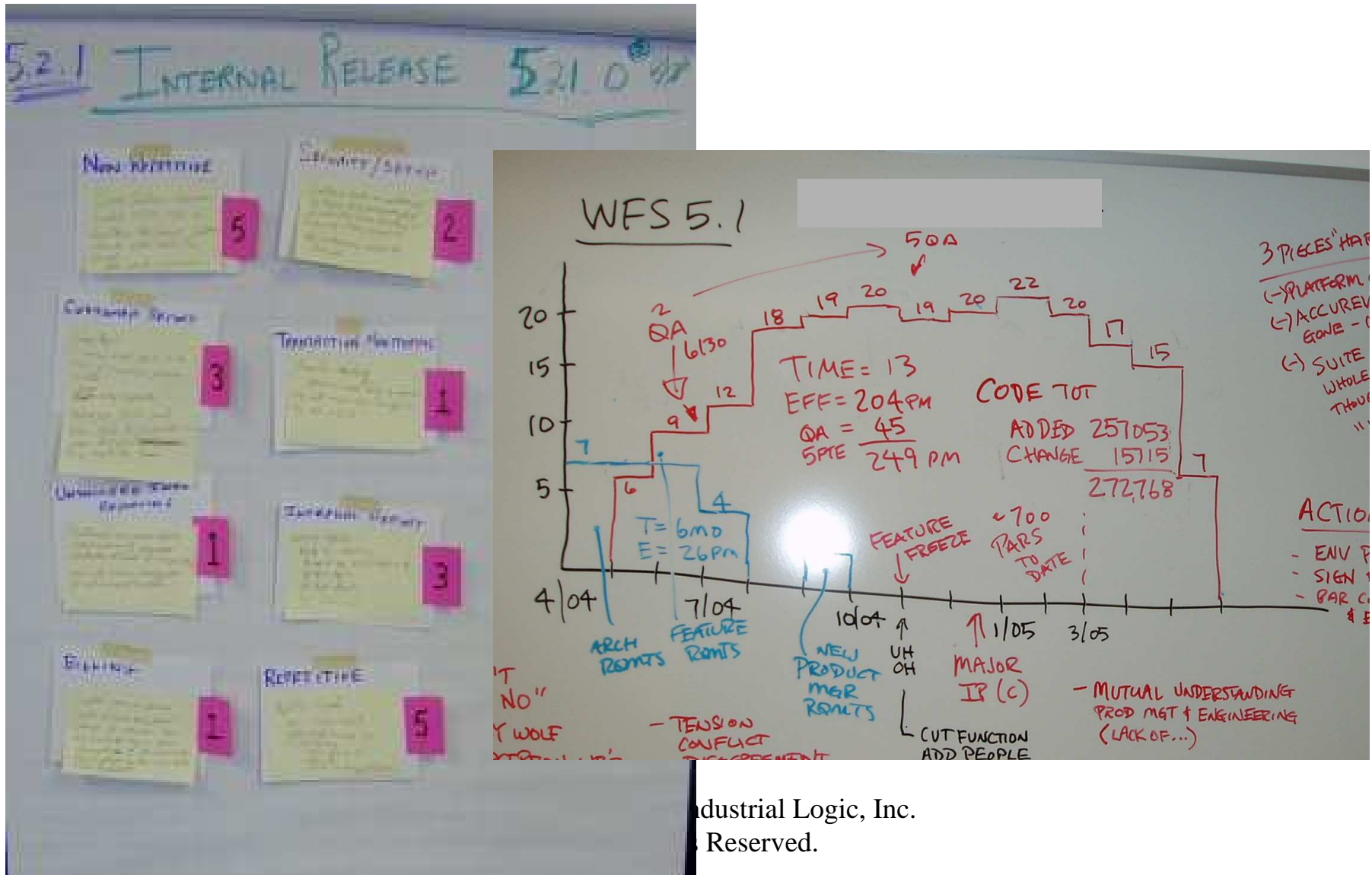
Programmers

Tester

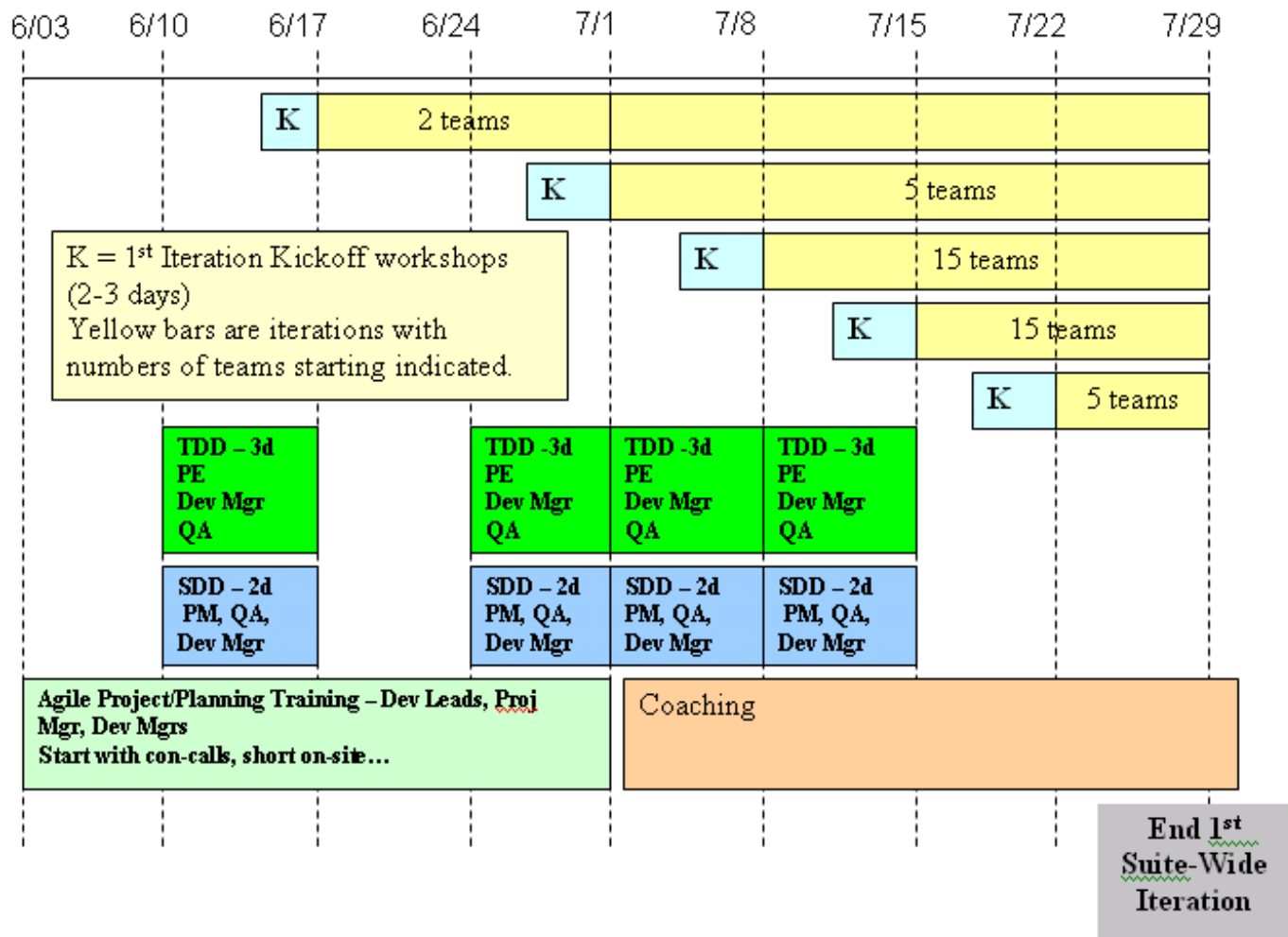
Subject
Matter
Expert

Tip #7

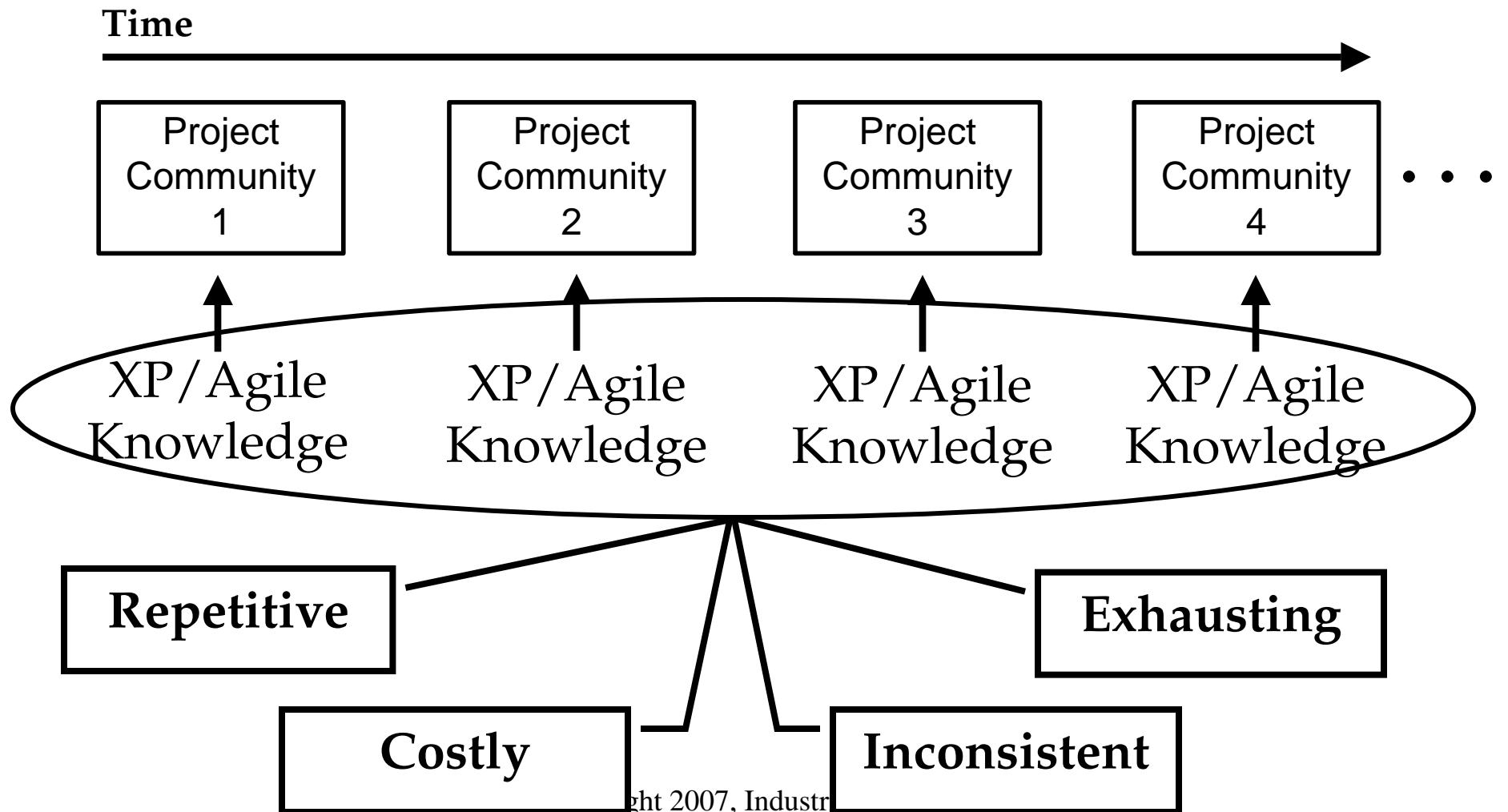
Pick A Big Important Project!



Tip #8 – Handle The Scaling Problems



Serialized Knowledge Transfer



eLearning To The Rescue

The screenshot shows a web interface for 'Industrial Logic'. At the top left is the 'industrial logic' logo, which consists of a red stylized 'IL' icon above the text 'industrial logic'. To the right of the logo is the main heading 'Scale Agility Faster.' in a large, bold, black font. Further right is a login section with the heading 'Login' above two input fields labeled 'User:' and 'Password:'. Below these fields is a 'Log In' button. In the center of the page, below the login section, is the text 'Please login to proceed to your workshop.' Below this text are three workshop cards. The first card is green and titled 'There Must Be 50 Ways To Leave Your Debugger', with a list of topics including 'RECURSION', 'FACTORY', 'TEST-DRIVEN DEVELOPMENT', 'STORYTEST-DRIVEN DEVELOPMENT', 'EVOLUTIONARY DESIGN', 'CODE SMELLS', and 'SOFT SKILLS'. The second card is blue and titled 'Break On Through To The Agile Side', with topics including 'AGILE', 'STORYTELLING', 'RELEASE & ITERATION', 'STORYTEST-DRIVEN DEVELOPMENT', 'RELEASE & RESUME', and 'REPRODUCTION'. The third card is yellow and titled 'Let The Good Designs Roll', with topics including 'DESIGN PATTERNS', 'DESIGN TOOLS', 'PATTERNS OF BEHAVIOR', 'PATTERNS OF DESIGN', 'PROLOGUE', and 'CODE'. Each card features the 'industrial logic' logo and the workshop title at the bottom.

industrial logic

Scale Agility Faster.

Login

User:

Password:

Log In

Please login to proceed to your workshop.

There Must Be 50 Ways To Leave Your Debugger

- RECURSION
- FACTORY
- TEST-DRIVEN DEVELOPMENT
- STORYTEST-DRIVEN DEVELOPMENT
- EVOLUTIONARY DESIGN
- CODE SMELLS
- SOFT SKILLS

THE TESTING & REFACTORING WORKSHOP

Break On Through To The Agile Side

- AGILE
- STORYTELLING
- RELEASE & ITERATION
- STORYTEST-DRIVEN DEVELOPMENT
- RELEASE & RESUME
- REPRODUCTION

THE EXTREME PROGRAMMING WORKSHOP

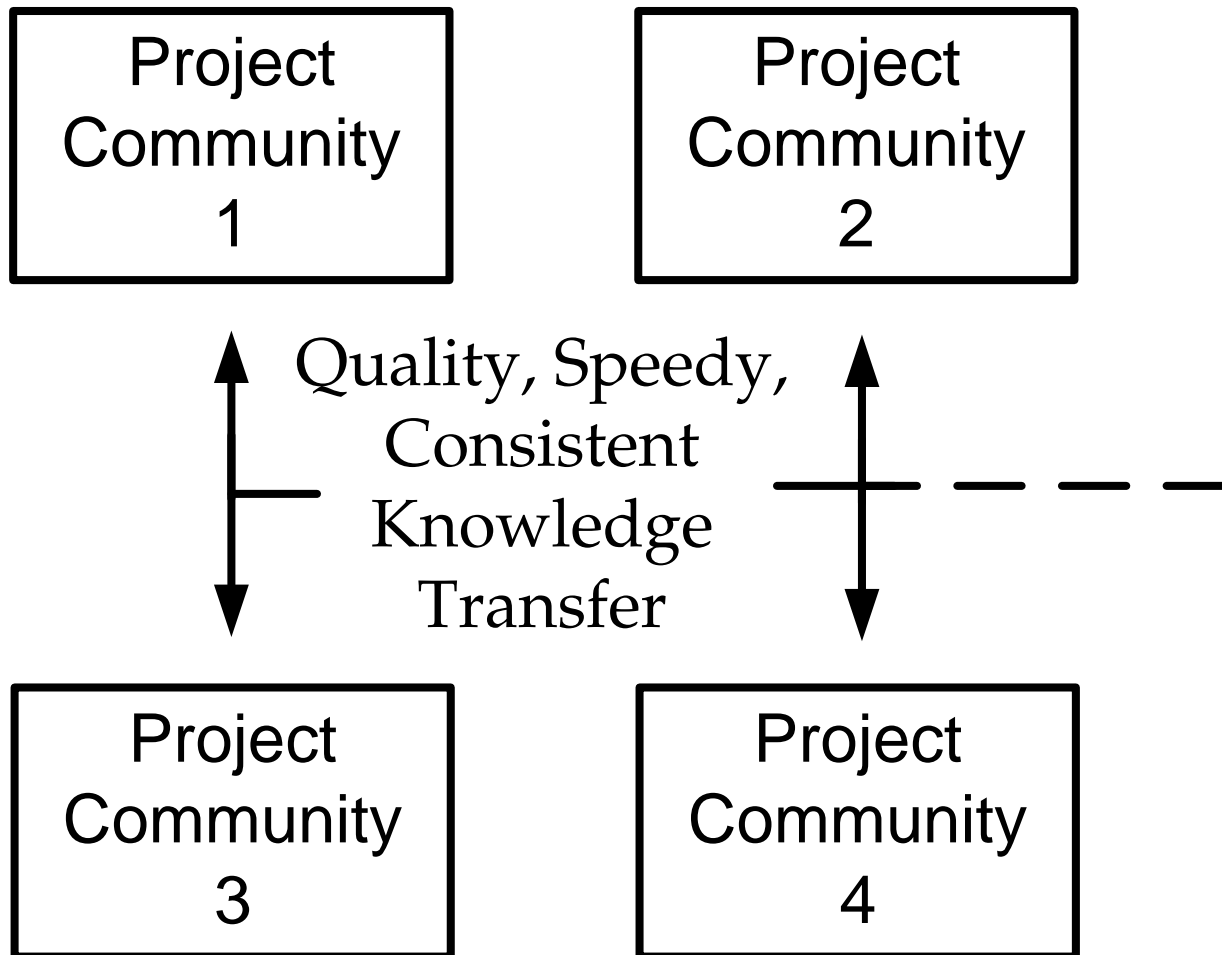
Let The Good Designs Roll

- DESIGN PATTERNS
- DESIGN TOOLS
- PATTERNS OF BEHAVIOR
- PATTERNS OF DESIGN
- PROLOGUE
- CODE

THE DESIGN PATTERNS WORKSHOP

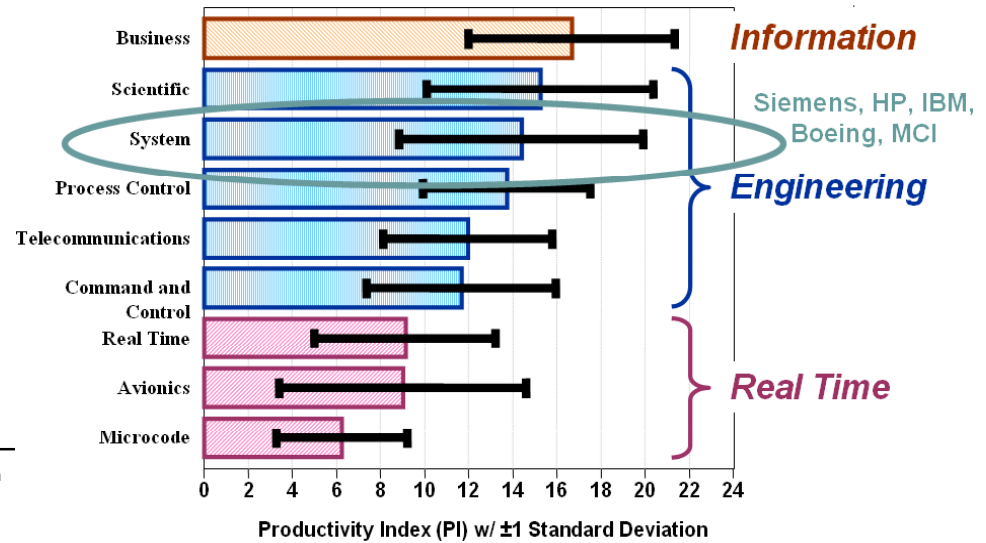
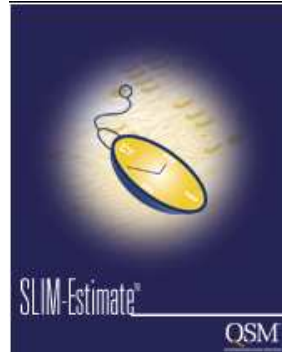
All Rights Reserved.

Parallel Knowledge Transfer





Tip #9 Gather Metrics!

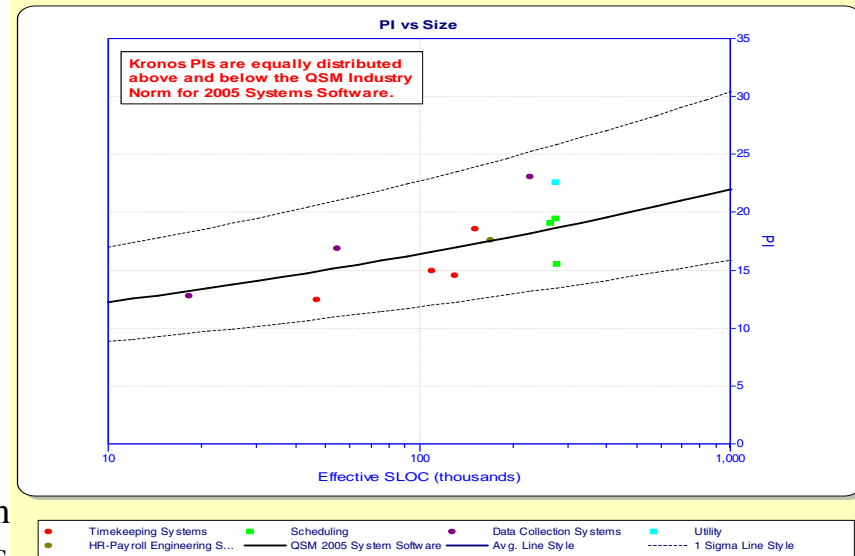


$$PI = \frac{SIZE}{TIME \cdot EFFORT}$$

$$Condor PI = \frac{376,022 \text{ SLOC}}{24_{\text{months}} * 347_{\text{person months}}} = 17$$

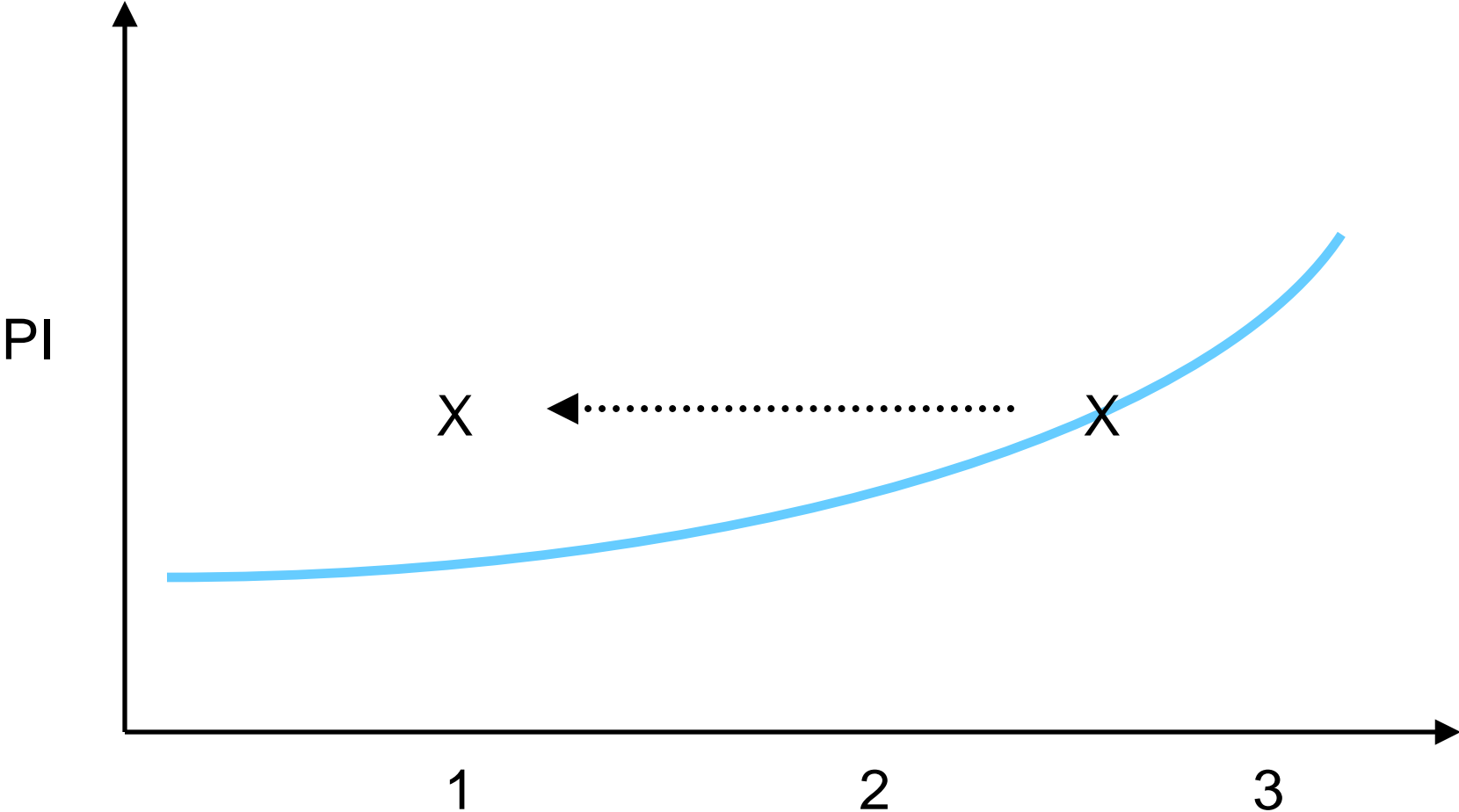
PI = measure of process productivity and development complexity

Larry Putnam: Software Metrics Guru & Pioneer.
Author of 5 Core Metrics.



Copyright 2007, In
All Rights Reserved.

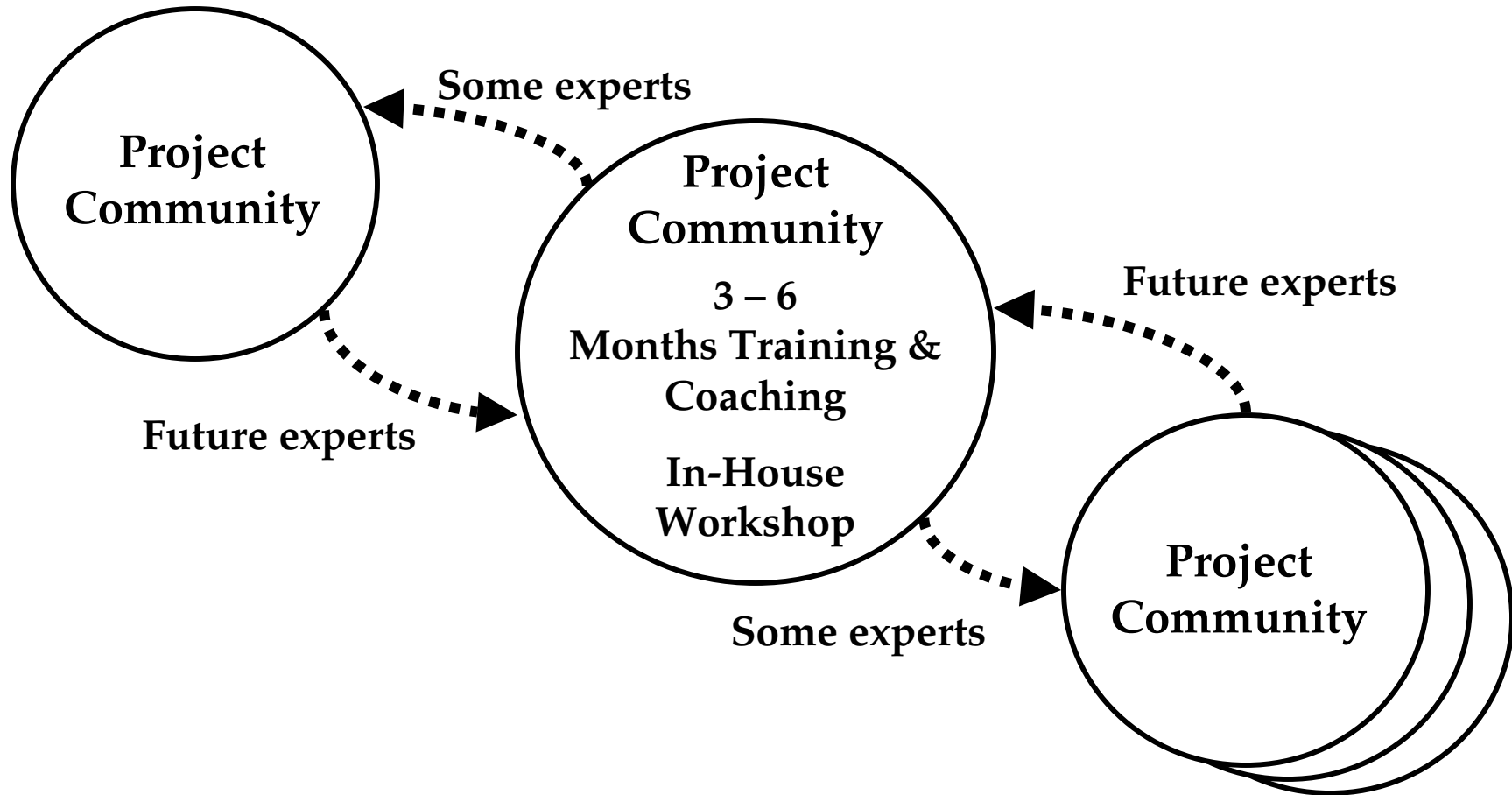
XP/Agile Yields Results Faster



Years

Copyright 2007, Industrial Logic, Inc.
All Rights Reserved.

In-House Workshop: A Lower Cost Strategy

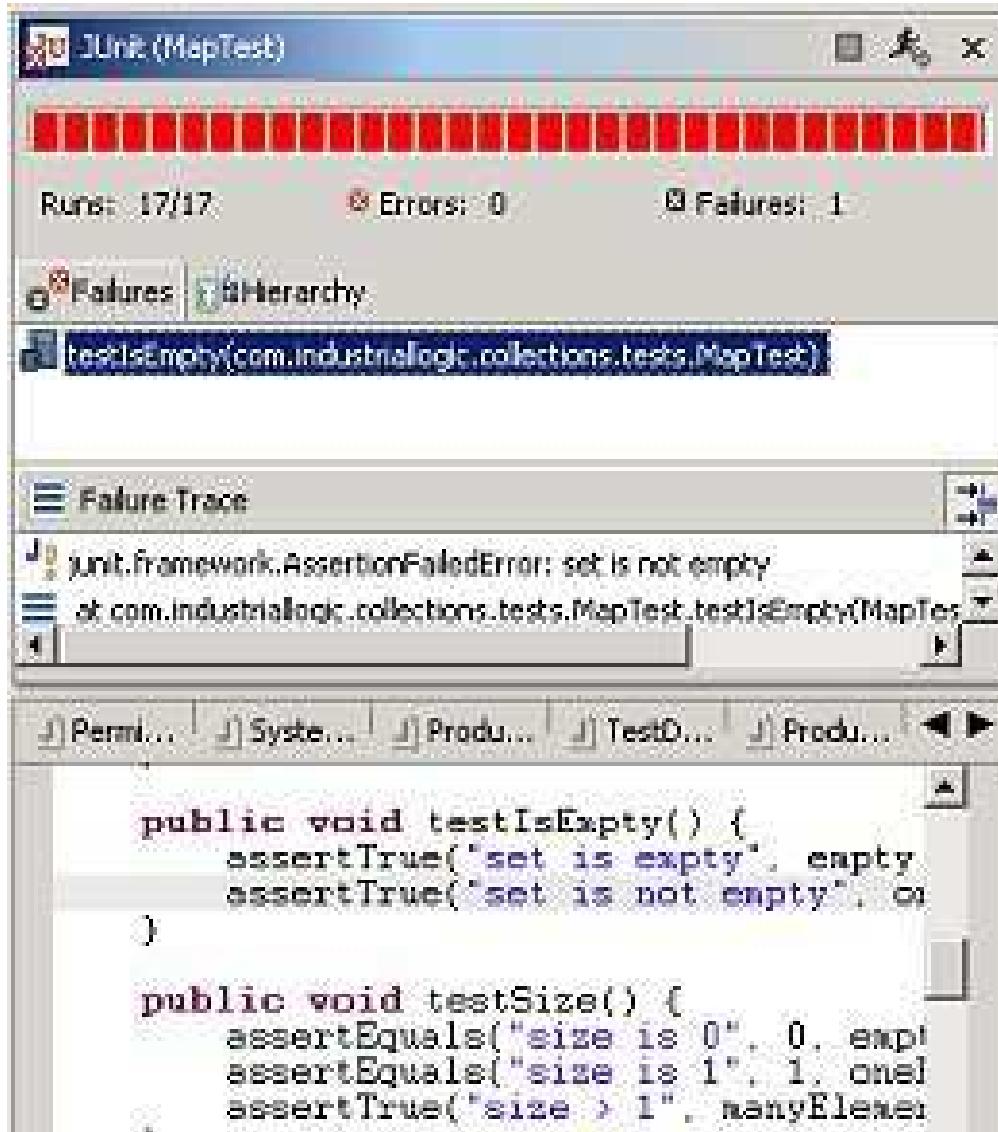


XP/Agile Productivity Improvement

	Previous Performance	Current Performance	Percentage Improvement
Cost	\$2.8 Million	\$1.1 Million	61%
Schedule	18 Months	13.5 Months	24%
Defects	2,270	381	83%
Staffing	18	11	39%

Thanks to Michael Mah of QSMA and Jim Highsmith of the Cutter Consortium
Copyright 2007, Industrial Logic, Inc.
All Rights Reserved.

Tip #10 - Fail Fast!



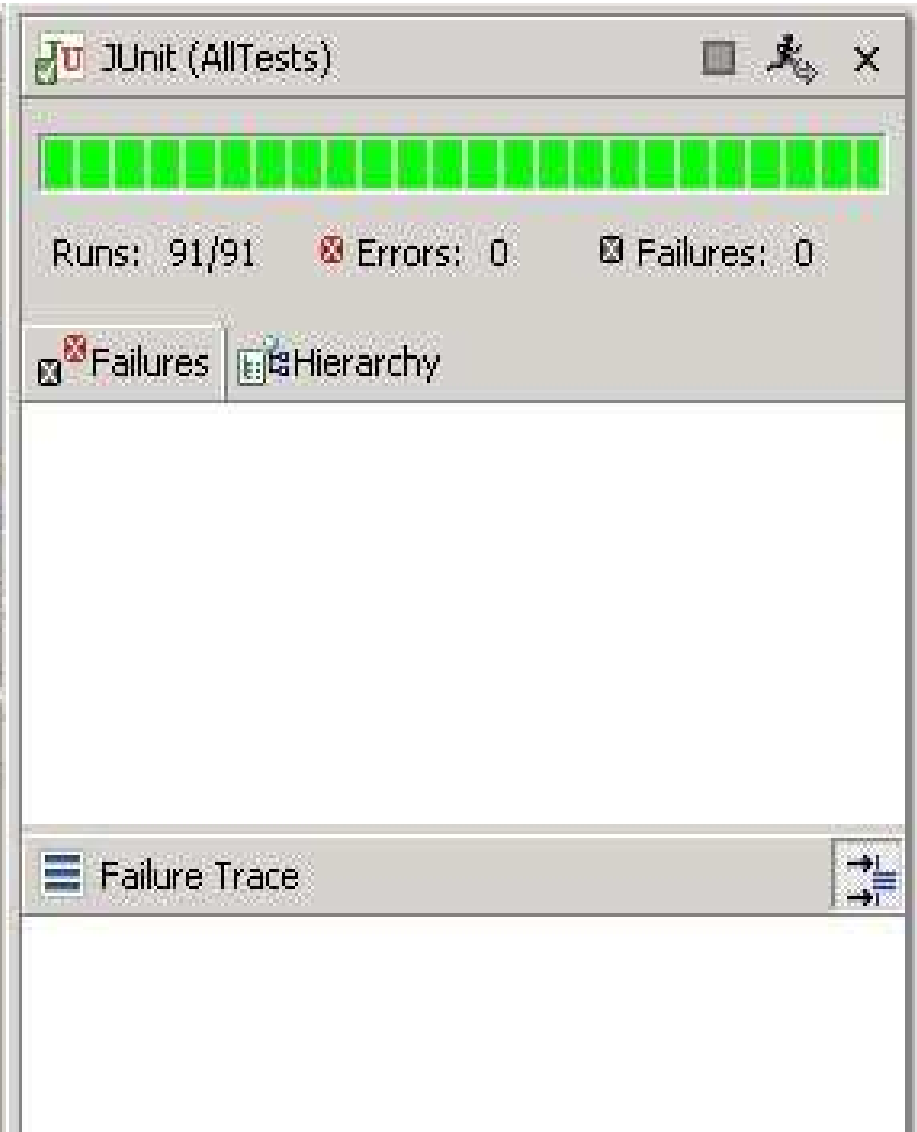
The screenshot shows the JUnit (MapTest) window. The progress bar is red, indicating a failure. The status bar shows "Runs: 17/17", "Errors: 0", and "Failures: 1". The "Failures" tab is selected, showing the failure message: "testIsEmpty(com.industriallogic.collections.tests.MapTest)". The "Failure Trace" section shows the following stack trace:

```
JUnit.framework.AssertionFailedError: set is not empty
    at com.industriallogic.collections.tests.MapTest.testIsEmpty(MapTest.java:10)
```

The code editor shows the following code:

```
public void testIsEmpty() {
    assertTrue("set is empty", empty);
    assertTrue("set is not empty", on);
}

public void testSize() {
    assertEquals("size is 0", 0, empty);
    assertEquals("size is 1", 1, one);
    assertTrue("size > 1", manyElements);
}
```



The screenshot shows the JUnit (AllTests) window. The progress bar is green, indicating a successful run. The status bar shows "Runs: 91/91", "Errors: 0", and "Failures: 0". The "Failures" tab is selected, showing no failures. The "Failure Trace" section is empty.

Failing Fast

FIT brings domain experts, analysts, testers and programmers into a test-driven development cycle that yields executable specifications.

FIT is a monumental achievement in software development because it utterly succeeds in uniting the actions of those who analyze, specify, test and program software systems.

FIT is an innovative tool that succeeds wildly in uniting the actions of those who analyze, specify, test and program software systems.

By guiding how domain experts, analysts, testers and programmers collaborate on the creation and validation of simple, tabular-based examples of business rules, FIT has fundamentally changes the way quality software gets built.

By providing a simple, effective method for creating and automating tabular examples of business rules, FIT has significantly improved how domain experts, analysts, testers and programmers collaborate to produce quality software.

THANK YOU!

Copyright 2007, Industrial Logic, Inc.
All Rights Reserved.