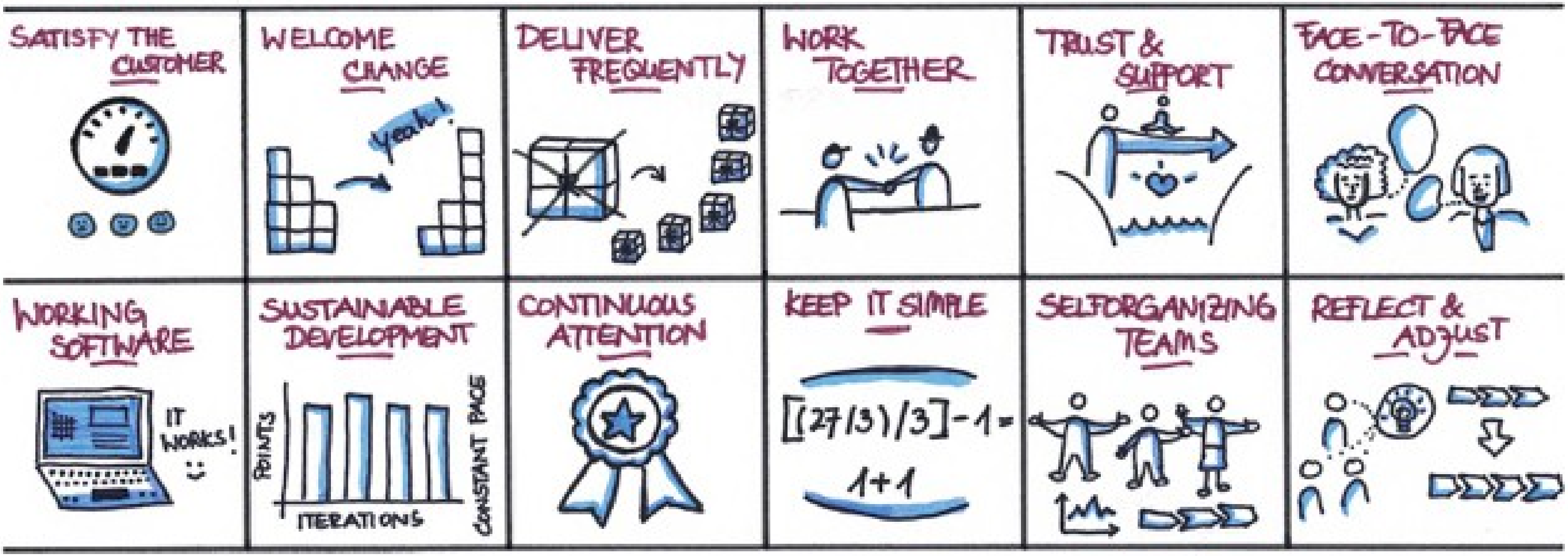
Why Organizational Agility Depends on Technical Excellence!

Marcel Baumann, tangly llc

Agile Manifesto Principles



Agile Manifesto Principles

Continuous attention to **technical excellence** and **good design** enhances agility.

Working software is the primary measure of progress.

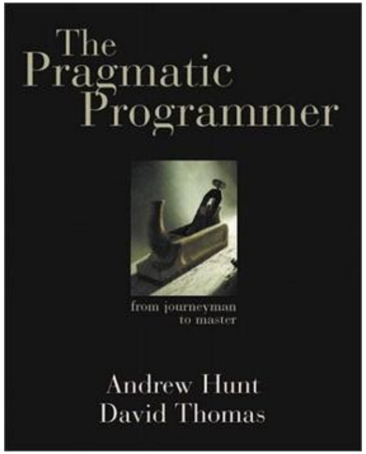
Build *projects* around motivated individuals. **Give them the environment and support they need**, and trust them to get the job done.

The best architectures, requirements, and designs emerge from **self-organizing teams**.

Historical Triggers

- The Pragmatic Programmer – 1999, 20th anniversary edition 2019
- Flaccid Scrum – Martin Fowler January 2009
- Software Craftsmanship Manifesto - 2009

- Dichotomy Agile / Scrum vs Developers

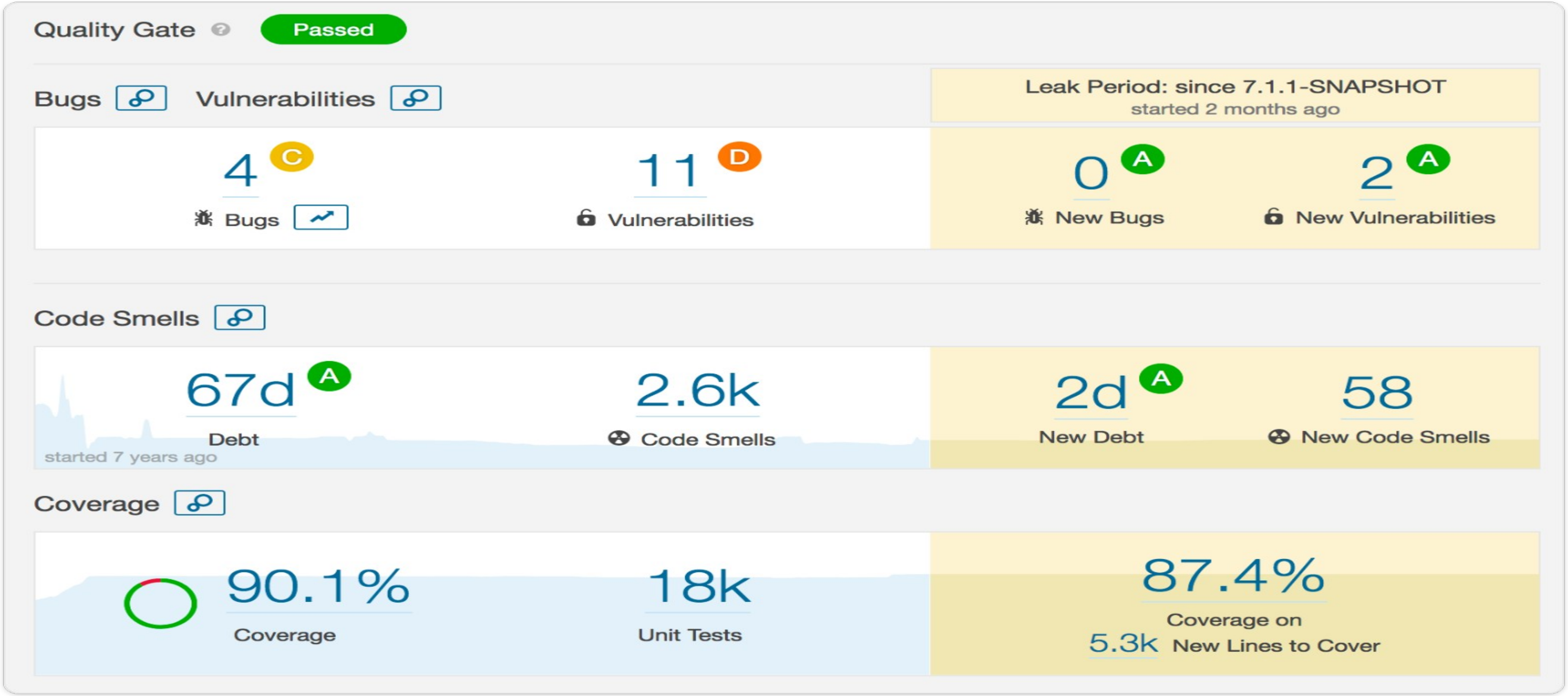


The Pragmatic Programmer

from journeyman to master

Andrew Hunt
David Thomas

How Can You Bootstrap?



Quality Gate Passed

Bugs Vulnerabilities

Leak Period: since 7.1.1-SNAPSHOT
started 2 months ago

4 C
Bugs

11 D
Vulnerabilities

0 A
New Bugs

2 A
New Vulnerabilities

Code Smells

67d A
Debt
started 7 years ago

2.6k
Code Smells

2d A
New Debt

58
New Code Smells

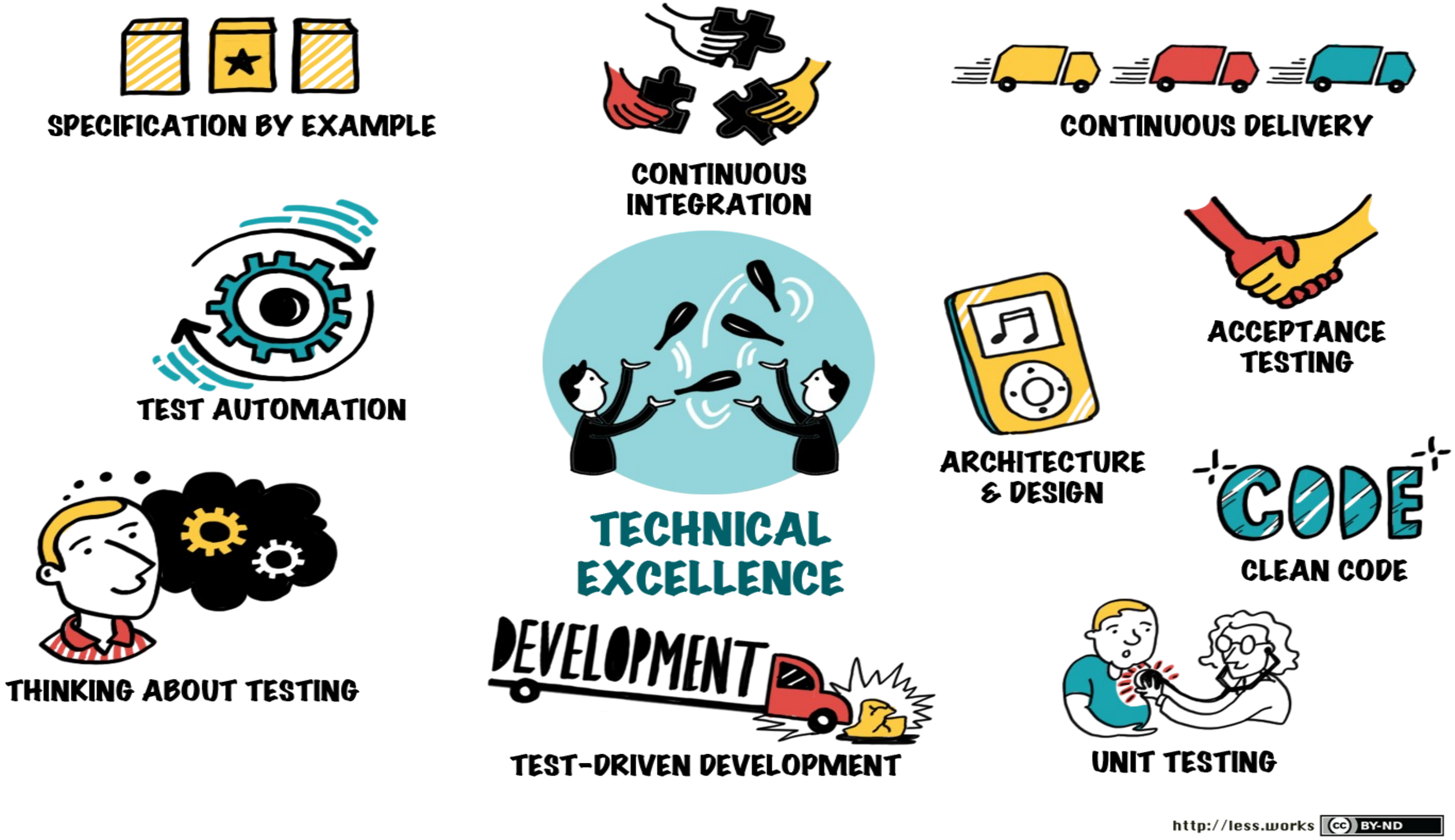
Coverage

90.1%
Coverage

18k
Unit Tests

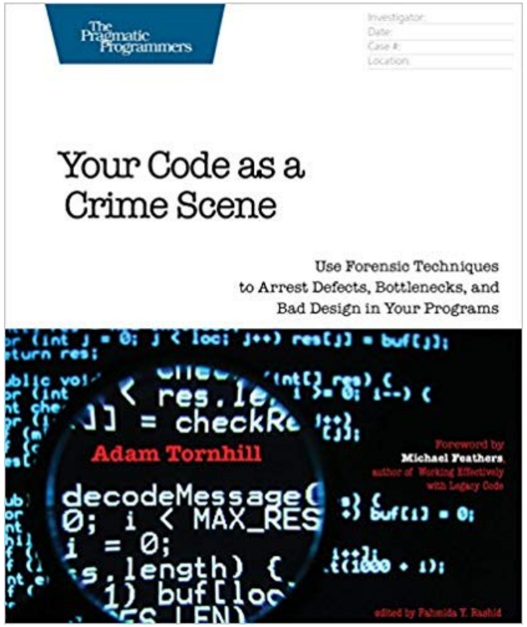
87.4%
Coverage on
5.3k New Lines to Cover

Which Practices are Helpful?



Good Practices

- Pair programing – Mob programming
- Coding dojos, technical meetings
- Git and your code as a crime scene
- Zero bug policy
- Pragmatic craftsmanship
- *Learning and training*

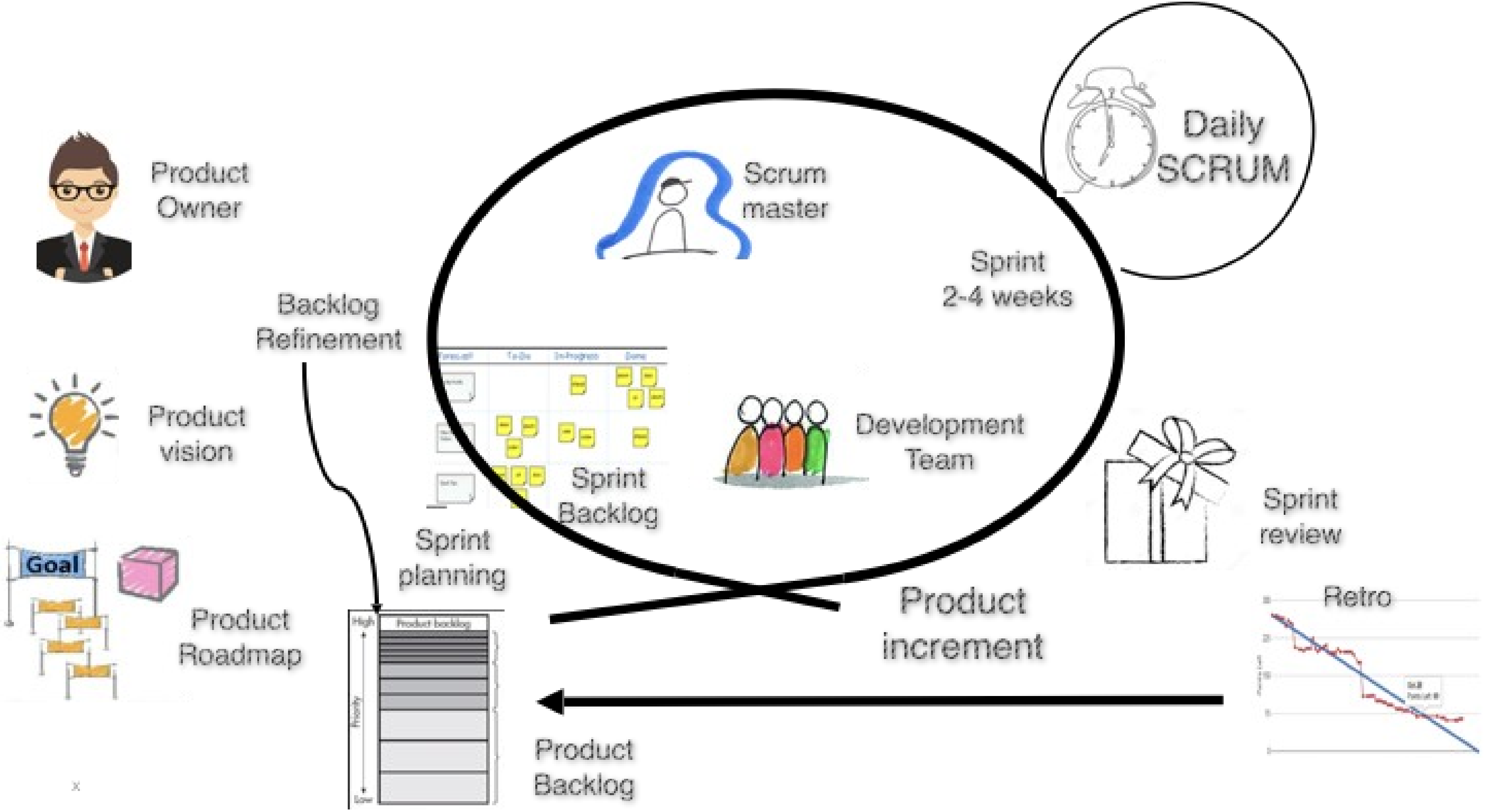

The Pragmatic Programmers
Your Code as a Crime Scene
Use Forensic Techniques to Arrest Defects, Bottlenecks, and Bad Design in Your Programs
Adam Tornhill
Foreword by Michael Feathers author of Working Effectively with Legacy Code
edited by Fahmida Y. Rashid

Which Challenges?

- Your team focus is on **feature delivery** and ignore quality
- Organization is not aware of **technical debt**
- Your organization is a **laggard**
  - SVN vs git, manual testing vs. TDD and ATDD
  - Do not touch a running system
  - No monitoring or logging
  - Old JDK and libraries

- *If your organization is a laggard, your cloud move will fail*

Scrum



Product Owner

Product vision

Product Roadmap

Backlog Refinement

Scrum master

Sprint 2-4 weeks

Daily SCRUM

Development Team

Sprint Backlog

Sprint planning

Sprint review

Product increment

Retro

Product Backlog

Scrum and XP

- Certified Scrum Developer
  - Scrum Alliance and Scrum.org
- Extreme Programming is acknowledged
  - Not really integrated *(2020-02-01)*
    - PSM-I 285’830, PSM-II 5’923, PSM-III 780
    - PSD 11’1324
      (ratio 1 PSD/26 PSM, which is strange given that there are several developers in one team and only one Scrum Master, expected 8 PDS/ 1 PSM – a difference of factor **200**)

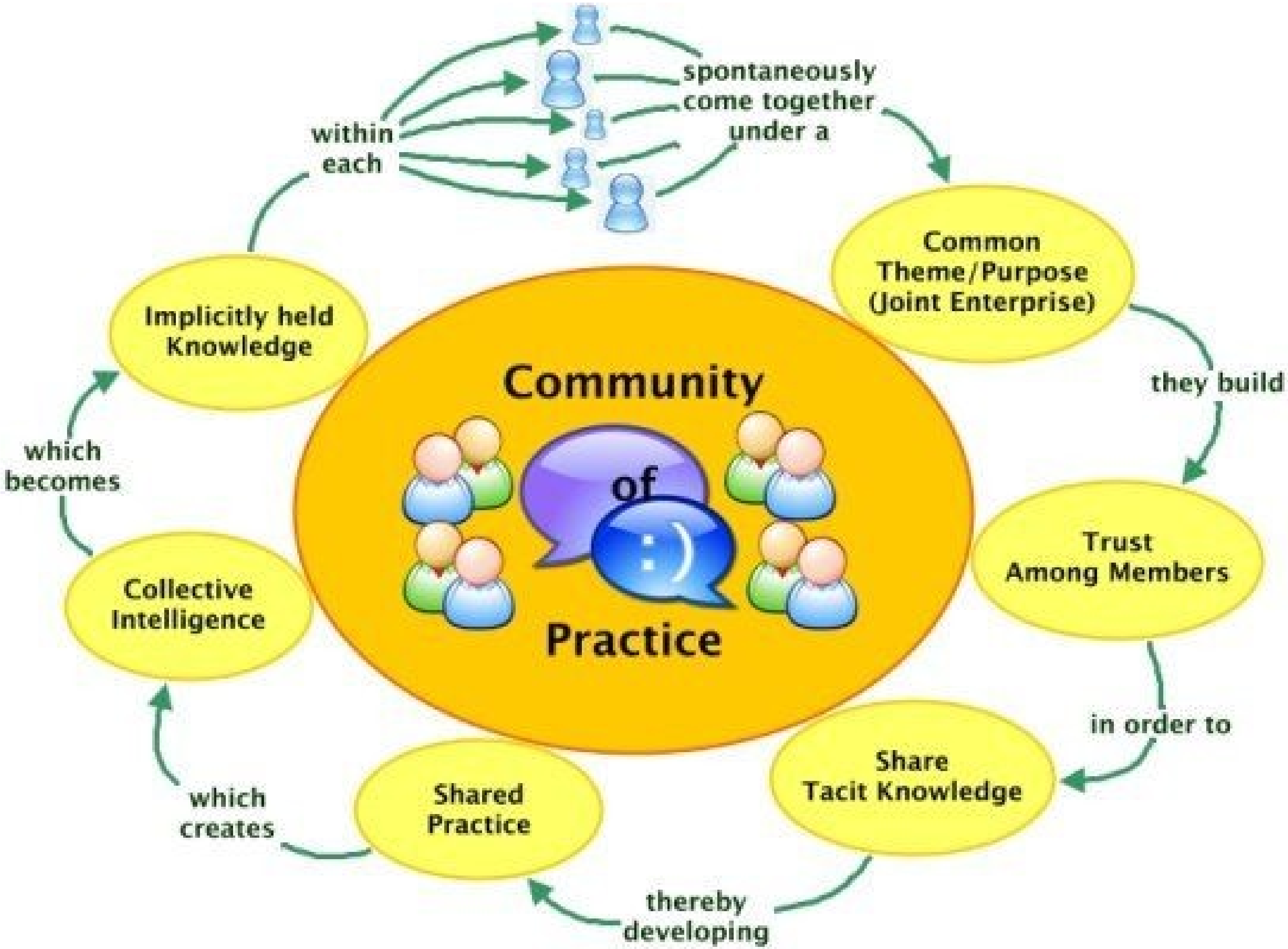
Scrum Master Role

- Scrum is a management framework to develop products
- Scrum master is a **teacher**, **coach** and facilitator
  - You can only teach what you know!
  - Scrum master shall not be just framework police!
  - *Remember* **working software** is the primary measure of progress

Scrum Master Team Responsibilities

The Scrum Master serves the Development Team in several ways, including:

- Coaching the Development Team in self-organization and **cross-functionality,**
- **Helping the Development Team to create high-value products,**
- **Removing impediments to the Development Team’s progress,**
- Facilitating Scrum events as requested or needed,
- Coaching the Development Team in organizational environments in which Scrum is not yet fully adopted and understood.

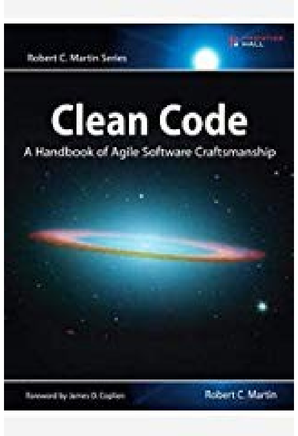

Community
of
:)
Practice
spontaneously
come together
under a
Common
Theme/Purpose
(Joint Enterprise)
they build
Trust
Among Members
in order to
Share
Tacit Knowledge
thereby
developing
Shared
Practice
which
creates
Collective
Intelligence
which
becomes
Implicitly held
Knowledge
within
each

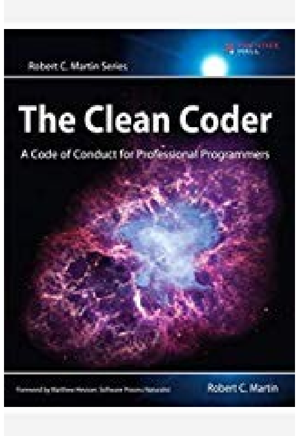
Scaling Approaches

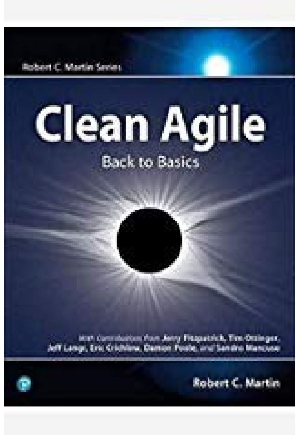
- LeSS
- Nexus and Scrum@Scale
- Spotify Model
- SAFe

Clean Movement

- Robert C. Martin clean code movement is known by all modern software developers
- Very direct, no frills approach

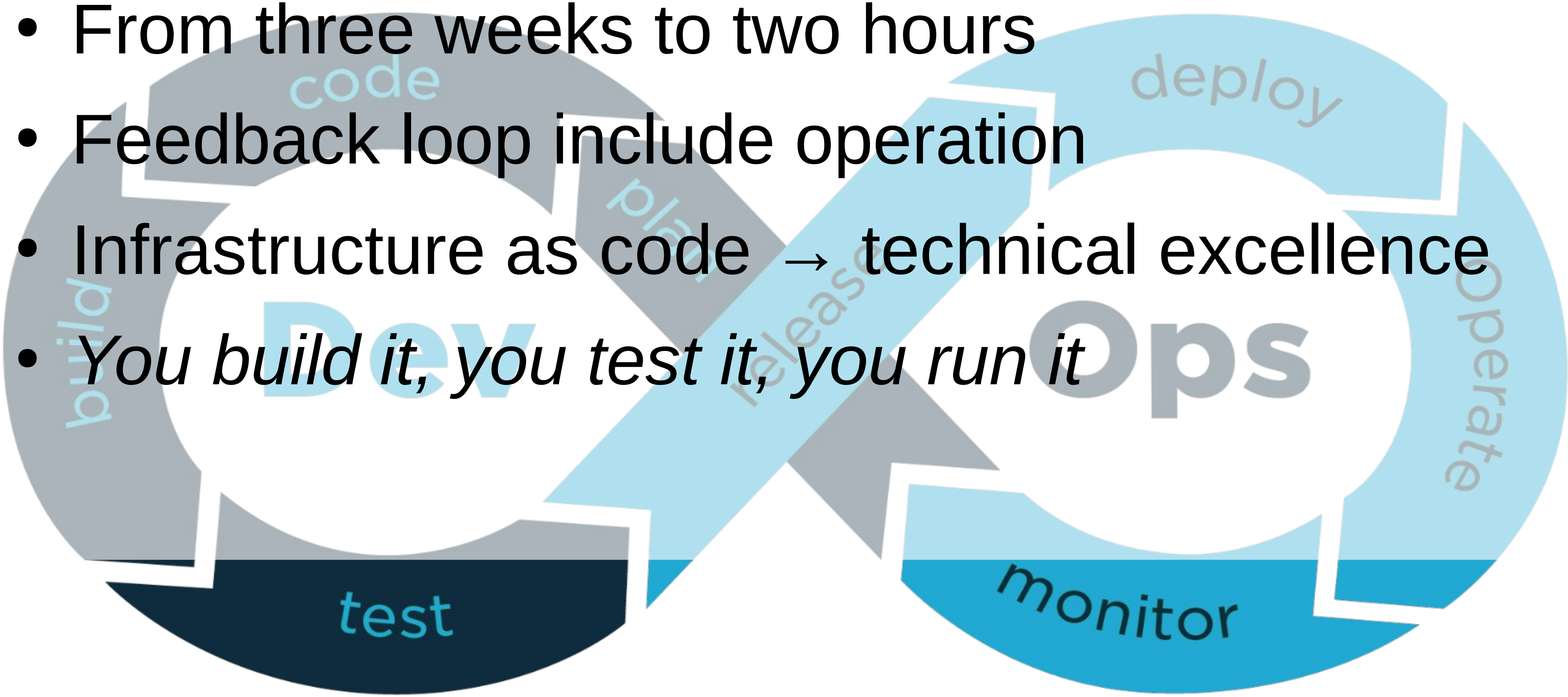

Robert C. Martin Series
Clean Code
A Handbook of Agile Software Craftsmanship
Foreword by James O. Coplien
Robert C. Martin




Robert C. Martin Series
Clean Agile
Back to Basics
With Contributions from Jerry Fitzpatrick, Tim Ottinger, Jeff Langr, Eric Crichlow, Damon Poole, and Sandro Mancuso
Robert C. Martin

DevOps Movement

- From three weeks to two hours
- Feedback loop include operation
- Infrastructure as code → technical excellence
- *You build it, you test it, you run it*



Findings

- Scrum or Kanban frameworks do **not** guarantee good software product
- You need **technical excellence** to succeed
- Scrum Master is **co-responsible** for technical excellence

Discussion

- Links to blogs
- tangly blog
- Questions and Answers