Keynote:
Technical Agile Coaching With the Samman Method

Emily Bache
Technical Agile Coaching
With the Samman method

Emily Bache
Emily Bache

Technical Agile Coach

Conference Speaker

Based in Gothenburg, Sweden

Author of

“Technical Agile Coaching with the Samman method”

@emilybache

emily.bache@proagile.se
Samman Coaching Outcomes

5. Have you seen these outcomes from the coaching?

More Details

- Improved teamwork, better collaboration
- More likely to write unit tests
- More likely to work in small increments, committing code more often
- More likely to improve design and refactor safely
- More likely to use tests to drive development
- More likely to design readable, maintainable test cases
What is important to you in your daily work as a software developer?
What is important to you in your daily work as a software developer?

Human Needs
- Salary
- Coffee
- Water
- Peace and quiet

Teamwork
- Diversity in collaboration management
- Positive collaboration
- Amazing colleagues

Learning and Professional Development
- Learning and constantly growing
- Teaching and mentoring others
- Safe to fail and learn

Making progress and achieving a Flow state
- Doing one thing at a time
- Well-organized project
- Little “Process”

Meaningful relationship with end users / customers
- Team collaboration
- Openness
- Psychological safety
- Supportive colleagues

Shared coding rules and Architecture
- Good code and coding environment
- Good coding environment
- Good architecture
- A learning organisation

Building something important
- Clear goals and objectives
- Contributing to something important
- Meeting users (no proxy)

Proper tooling
- Tools I know and understand
- Good tooling
- Good testing

By Emily Bache. Released under a Creative Commons Share Alike license.
XP Team

Traditional Team

@emilybache
Do you see a need to get better at any of these technical practices?

Mentimeter survey
Accelerate Research

- Many years of research
- Respected authors
High Performing Behaviours

Generative & Safe Culture

Empowered Teams without Handovers

Use the Cloud

Lean Product Management

Continuous Delivery

Modular Architecture

THE SCIENCE OF DEVOPS ACCELERATE
Building and Scaling High Performing Technology Organizations
Nicole Forsgren, PhD
Jez Humble and Gene Kim
Software Today is Built by Teams

- Psychological Safety
- Cross-functional
- Empowered

Empowered Teams

- The team can:
- Make large-scale design changes
  - Without detailed coordination outside the team
- Deploy on-demand
- Do their own testing

Accelerate Research Results

- Continuous Delivery (CD)
  - Less Burnout
  - Less Deployment Pain
  - Reach Organizational Goals

Technical Practices

Accelerate Research Results

- Automated deployment
- Continuous Integration (CI)
- Test-Driven Development
- Trunk-based development
- Loosely-coupled architecture
- Database agility

Continuous Delivery (CD)

- Reach Organizational Goals
- Less Deployment Pain
- Less Burnout

Test-Driven Development

Red

Refactor

Green
Demo

Leap Years
Test-Driven Development

Red

Refactor

Green
Feedback loops

http://wiki.c2.com/?TestDrivenDevelopment
Accelerate Research Results

- Automated deployment
- Continuous Integration (CI)
- Test-Driven Development
- Trunk-based development
- Loosely-coupled architecture
- Database agility

Continuous Delivery (CD)

Reach Organizational Goals
Less Deployment Pain
Less Burnout

Continuous Integration

commit → Build & unit test → Deployable Component → system test → load test → acceptance test → Deploy to production

Pipeline is green to this point at least daily
Trunk-Based Development

Release 12.1

Trunk (aka Master, Main)

Release 12.2

Short-lived Feature Branches

https://trunkbaseddevelopment.com/
Continuous Integration & Trunk based dev & TDD

Everyone has *almost* the same code on their local machines

Shared view of status
XP Team

Traditional Team
Do you already work this way?

Mentimeter survey
A new way of skiing

Flickr Simon Hannford

Flickr nonanet
Which of these learning activities are you already doing?

Mentimeter survey
Which of these training activities have you successfully used to learn TDD?

Mentimeter survey
Coding Dojos

Dojo == the place you go to learn
A Coding Dojo

- About 5-15 coders
- 2-3 hours each week/month
Pair Coaching with Llewellyn Falco
Technical Agile Coaching with the Samman method

By Emily Bache
Foreword by Kent Beck

- Coaching method for working with software development teams
- Improve the way we write software
- “Samman” means “together”

Book published Jan 2021
Samman Technical Coaching Areas

- Better Unit Tests
- Continuous Integration
- Safe Refactoring
- Iterative and Incremental Design
- Legacy Code Rescue
- Levelling up the whole team
The Samman Method

- Learning Hours
- Ensemble Working
Learning Hour

- Like a short, frequent coding dojo
- Planned by a teacher/coach
Training from the back of the room

- Active learning techniques
- Hands-on, experiential
- Like a skiing lesson!
Learning hours have a “4C” structure

- Connect

Which of these learning activities are you already doing?

- Book club: 0
- Instructor-led training courses for 2-5 days at a time: 1
- Online on-demand training videos: 1
- Frequent pair programming: 0
- Brown bag lunch-n-learn: 0
- Practicing on code katas and exercises: 0
Learning hours have a “4C” structure

- Connect
- Concept

Learning Hour

- Like a short, frequent coding dojo
- Planned by a teacher/coach
Learning hours have a “4C” structure

- Connect
- Concept
- Concrete

Design some training that uses the 4C model.
Sample Learning Hour Outline

Misconceptions about Refactoring

In this session we talk about what refactoring is and why we do it. We work on improving some names to improve readability.

Session Outline

- 15 min connect: refactoring warm-up questions
- 5 min concept: When should you refactor
- 30 min do: pairs refactor Tennis3
- 5 min reflect: review warm-up questions

https://sammancoaching.org/learning_hours/refactoring/misconceptions.html
Learning hours have a “4C” structure

- Connect
- Concept
- Concrete
- Conclusions

Write a comment in the group chat summarizing what you’ve learned about ‘Training from the back of the room’
Interact with the material

- Mentimeter
- Miro board
- Printed Paper Handouts that you write on
I will read out any comments I like

Conclusions, Summary, Evaluation, Celebration
The Samman Method

- Learning Hours
- Ensemble Working
A new way of skiing
TDD in a code kata

TDD “in real life”

Flickr Ruth Hartnup
Ensemble Working

- Whole team
- Realistic task
- Practicing techniques rather than making progress
“All the brilliant people working on the same thing, at the same time, in the same space, on the same computer.”
– Woody Zuill

Mob Programming
Book recommendation:

https://leanpub.com/mobprogramming
Ensemble Roles

- Typist
- Navigator
- Team member
- Rotate roles often
Demo

Mars rover ensemble
No decisions at the keyboard

- Typist enters code into the computer
- Navigator & ensemble decide *what* code
Working Agreement

- We treat everyone with kindness, consideration and respect
Technical Coach joins the ensemble
Describe characteristics of good technical leaders (architects)

Mentimeter survey
Coaching Behaviours

Working with the team
Agile Coaching Competency Framework

Agile-Lean Practitioner

Professional Coaching

Mentoring

Teaching

Technical Mastery

Business Mastery

Transformation Mastery

Facilitating

See more at Resources on AgileCoachingInstitute.com

©2015 Agile Coaching Institute

Agile Coaching Competency Framework is Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.
Teaching

TDD

Ideas to remember concepts to retain

- Write test first
- Then implement functionally
- Many small steps instead of 1 large
- Always write the right test
- 1 step at a time
- Don’t change test before change in code
- Write code to cover test cases
- Don’t change test when you change code
- Incrementally add code make them pass

Golden Rule of TDD

Do not write any production code until you have a failing test that requires it

@emilybache
Mentoring

\[ IDT = 1 \]

\[ \text{Time Qu} = \frac{10}{45} \]

\[ \text{Time Qu Mech.} = \text{CIP' / DTP} \]

\[ t + 10 \]
Facilitating
Professional Coaching
Describe characteristics of good technical leaders (architects)
Samman Technical Coaching

- Whole team
- 2 hours Ensemble programming per day
- 1 learning hour per day
- 10 coaching days at a time
- Deliberate breaks between coaching
Local or Remote
Breaks between coaching blocks

- Example: Four teams in rotation
- Each team has 10 day coaching block then about 4 weeks break before next one
Team Chartering

Each team is unique, so before the coaching begins:

- **Workshop 1**: Coach learns about the team’s situation
- **Workshop 2**: Collaboratively agree topics and goals for the coaching
Samman Coaching Goals

- Teams and organizations improve their technical practices
- Learning new skills with your team becomes a normal part of your job
- Changed behaviours continue after the coach has left
The world needs more technical coaches
Do you think technical coaching is something for you personally?

Mentimeter survey
The world needs more Technical Agile Coaches!
Samman Technical Coaching

Ensemble Working

Learning Hour

Test → Code → Integrate → Deploy → Release → Steer