Chapter Summaries

Sprinting In Place: Why Your Agile Team Isn't Agile and What You Can Do About It

By Jeffrey Fredrick and Douglas Squirrel

Introduction

We define "agile" and suggest that this book is *not* for those whose teams are delivering quickly and smoothly, but for everyone else in charge of or working with a team that is following agile practises but is nevertheless not functioning well. We briefly introduce the Agile Manifesto, whose values form our main chapter headings, and the idea of "Action Science", the academic theory that informs many of the techniques we describe.

How Not To Use This Book

This very short chapter aims to put the reader in the appropriate frame of mind for the rest of the book, by describing how she should *not* approach the ideas we introduce. For example, if the reader is looking for someone else to blame, trying to "win" politically or personally, looking to save face and avoid embarrassment for herself and others, or aiming for others to change behaviour without herself changing, she is unlikely to get much from our book. We suggest, only partly tongue in cheek, that she repeat to herself at intervals while reading, "It's my fault."

Individuals and interactions over processes and tools

Agile software development done wrong can seem soulless - it's all about grooming the backlog and maintaining the kanban board, not about people and their messy, inconsistent desires. The answer to this is right there at the beginning of the Agile Manifesto - vigorous, frequent communication between people is what we need. We introduce the Action Science idea of "double-loop learning" and show how readers can apply it to agile activities like retrospectives and root-cause analyses to get more effective communication and problem-solving in their teams; then we describe how communication techniques like the Core Protocols and the Coleman Blitz can help a team accelerate adoption of these ideas.

Working software over comprehensive documentation

One of the biggest problems with software development is the lack of *valid information* - about the software itself, or the state of the build, or the morale of the team. We may turn to extensive documentation (interface specs, timesheets, burndown charts, and more) to gain comfort, but this information is far too often out of date and misleading. Toyota managers use the phrase "Genchi Genbutsu" or "see for yourself" to refer to the alternative approach of getting valid information from the source; we describe several methods for doing this through conversations with your team members, including the Ladder of Inference, also known as "test-driven development for people". As a bonus, once you get in the habit of exchanging reliable information, you build trust and everyone can act more efficiently, without the need for extensive documents or signoff rituals.

Customer collaboration over contract negotiation

In this chapter we tell several stories about building relationships as a route to improved delivery and agility for teams we've worked with. We point out that relationship-building is a high-investment, high-reward alternative, often overlooked in favour of agile "ceremonies" that don't by themselves address problems like mistrust between team members or unclear goal-setting by management. We give step-by-step guides for using techniques including relationship mapping, coherence busting, and the Mirror Principle, each of which is designed to help the reader and her team improve relationships with others inside and outside the team.

Responding to change over following a plan

A comprehensive management study at Google found that "psychological safety", or the ability to take risks and be vulnerable without repercussions, was the number one contributor to team success. We think psychological safety is particularly important in helping a team respond to internal or external change, so we spend this chapter explaining how to talk to people in your team in a way that encourages this level of safety and gets you the creative thinking and valid information you need to perform better and respond in an agile way. Approaches we suggest include "say what you see", "ask 'why' to get unexpected answers", and "outlaw 'should' and 'best practise'".

Knotty Problems

In this chapter we suggest some failsafe approaches - things to try when the ideas in the rest of the book don't go far enough. How do we increase communication when the parties are on different continents or won't speak to each other? How do we collaborate when distant managers impose contract-style signoff rituals? How do we create psychological safety when people are being fired right and left? Our techniques include starting small ("when and how long should the first phone call be?") and finding the "economic sponsor" who can be an ally for cultural changes that have a direct effect on the bottom line.

It's Your Fault: An Action Science Story

Throughout the book we've emphasized that improvement in team agility will only come through action by the reader - it really is her fault, which is in fact a liberating frame since it means the reader can take steps to improve her situation. By way of conclusion, in this chapter we tell a short fictional story about a team whose leader accepts responsibility for change and applies a variety of our techniques to achieve substantial gains in communication, relationship quality, and agility.

Illustration [end papers or middle section or similar]

One week to improvement - a step-by-step five-day schedule for applying techniques from the book, with cross-references to the relevant sections.