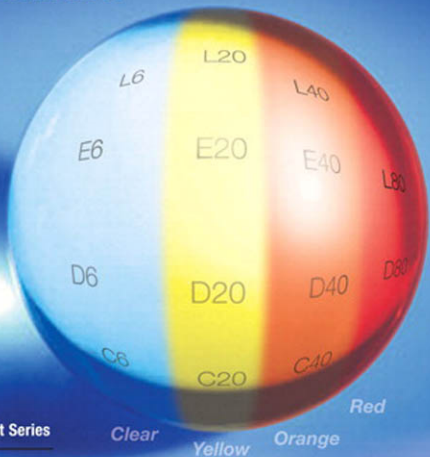




Crystal Clear

A Human-Powered Methodology for Small Teams

Alistair Cockburn



Agile Software Development Series

Alistair Cockburn and Jim Highsmith,
Series Editors

“Crystal Clear is beyond ‘Agile’ since I think that Crystal Clear is Agile + Flexible + Practical. ‘Flexible and Practical’ is important for us. This is not theoretical methodology but practical methodology that lots of people are doing. When I read *Crystal Clear*, I felt that this was easy to understand. This book helps us from software process hell to successful software development through practical examples and useful samples.”

—*Basaki Satoshi, Schlumberger*

“Alistair Cockburn shows how small teams can be highly effective at developing fit-for-purpose software by following a few basic software development practices and by creating proper team dynamics. These small teams can be much more effective and predictable than much larger teams that follow overly bureaucratic and prescriptive development processes.”

—*Todd Little, Sr. Development Manager, Landmark Graphics*

“I find Cockburn’s writings on agile methods enlightening: He describes ‘how to do,’ of course, but also how to tell whether you’re doing it right, to reach into the feeling of the project. This particular book’s value is that actual project experiences leading to and confirming the principles and practices are so...well...clearly presented.”

—*Scott Duncan, ASQ Software Division Standards Chair and representative to the US SC7 TAG and IEEE S2ESC Executive Committee and Management Board and Chair of IEEE Working Group 1648 on agile methods*

“*Crystal Clear* identifies principles that work not only for software development, but also for any results-centric activities. Dr. Cockburn follows these principles with concrete, practical examples of how to apply the principles to real situations and roles and to resolve real issues.”

—*Niel Nickolaissen, COO, Deseret Book*

“All the successful projects I’ve been involved with or have observed over the past 19 or so years have had many of the same characteristics as described in *Crystal Clear* (even the big projects). And many of the failed projects failed because they missed something—such as expert end-user involvement or accessibility throughout the project. The final story was a great read. Here was a project that in my opinion was an overwhelming success—high productivity, high quality, delivery, happy customer, and the fact that the team would do it again. The differing styles in each chapter kept it interesting. I started reading it and couldn’t put it down, and by the end, I just had to say ‘Wow!’”

—*Ron Holliday, Director, Fidelity Management Research*

“*Crystal Clear* gives me both essential and practical ways of creating and navigating motivated software development teams. At the same time, it increases the quality of engineering life!”

—*Kenji Hiranabe, Project Manager, Eiwa System Management, Inc.*

Crystal Clear: A Human-Powered Methodology for Small Teams

Table of Contents

Contents

Preface

Chapter 1 Explained (View from the Outside)

Chapter 2 Applied (The Seven Properties)

Property 1. Frequent Delivery

Property 2. Reflective Improvement

Property 3. Osmotic Communication

Property 4. Personal Safety

Property 5. Focus

Property 6. Easy Access to Expert Users

Property 7. Technical Environment with Automated Tests,
Configuration Management, and Frequent Integration

Evidence: Collaboration across Organizational Boundaries

Reflection on the Properties

Chapter 3 In Practice (Strategies and Techniques)

The Strategies

Strategy 1. Exploratory 360°

Strategy 2. Early Victory

Strategy 3. Walking Skeleton

Strategy 4. Incremental Rearchitecture

Strategy 5. Information Radiators

The Techniques

Table of Contents

Technique 1. Methodology Shaping

Technique 2. Reflection Workshop

Technique 3. Blitz Planning

Technique 4. Delphi Estimation Using Expertise Rankings

Technique 5. Daily Stand-up Meetings

Technique 6. Essential Interaction Design (Jeff Patton)

Technique 7. Process Miniature

Technique 8. Side-by-Side Programming

Technique 9. Burn Charts

Reflection about the Strategies and Techniques

Chapter 4 Explored (The Process)

The Project Cycle

The Delivery Cycle

The Iteration Cycle

The Integration Cycle

The Week and the Day

The Development Episode

Reflection about the Process

Chapter 5 Examined (The Work Products)

The Roles and Their Work Products

Roles: Sponsor, Expert User, Lead Designer,
Designer-Programmer, Business Expert, Coordinator, Tester,
Writer

A Note about the Project Samples

Sponsor: Mission Statement with Trade-off Priorities

Team: Team Structure and Conventions

Team: Reflection Workshop Results

Table of Contents

Coordinator: Project Map, Release Plan, Project Status, Risk List,
Iteration Plan and Status, Viewing Schedule

Coordinator: Project Map

Coordinator: Release Plan

Coordinator: Project Status

Coordinator: Risk List

Coordinator: Iteration Plan & Iteration Status

Coordinator: Viewing Schedule

Business Expert and Expert User: Actor-Goal List

Business Expert: Requirements File

Business Expert and Expert User: Use Cases

Expert User: User Role Model

Designer-Programmers: Screen Drafts, System Architecture,
Source Code, Common Domain Model, Design Sketches and
Notes

Designer-Programmer: Screen Drafts

Lead Designer: System Architecture

Designer-Programmer: Common Domain Model

Designer-Programmer: Source Code and Delivery Package

Designer-Programmer: Design Notes

Designer-Programmer: Tests

Tester: Bug Report

Writer: Help Text, User Manual, and Training Manual

Reflection about the Work Products

Chapter 6 Misunderstood (Common Mistakes)

We colocated and ran two-week iterationswhy did we fail?

Two developers are separated by a hallway and a locked door.

Table of Contents

We have this big infrastructure to deliver first.

Our first delivery is a demo of the data tables.

No user is available, but we have a test engineer joining us next week.

One developer refuses to discuss his design or show his code to the rest.

The users want all of the function delivered to their desks at one time...

We have some milestones less than a use case and some bigger.

We wrote down a basic concept and design of the system. We all sit together, so that should be good enough.

Who owns the code?

Can we let our test engineer write our tests? How do we regression test the GUI?

What is the optimal iteration length?

Chapter 7 Questioned (Frequently Asked)

Question 1. What is the grounding for Crystal?

Question 2. What is the Crystal family?

Question 3. What kind of methodology description is this?

Question 4. What is the summary sheet for Crystal Clear?

Question 5. Why the different chapter formats?

Question 6. Where is Crystal Clear in the pantheon of methodologies?

Question 7. What about the CMM(I)?

Question 8. What about UML and architecture?

Question 9. Why aim only for the safety zone? Cant we do better?

Table of Contents

Question 10. What about distributed teams?

Question 11. What about larger teams?

Question 12. What about fixed-price and fixed-scope projects?

Question 13. How can I rate how agile or how crystal we
are?

Question 14. How do I get started?

Chapter 8 Tested (A Case Study)

The Field Report

The Auditors Report

Reflection on the Field and Audit Reports

Chapter 9 Distilled (The Short Version)

References

Index