Patterns for the People

@KevlinHenney

kevlin@curbralan.com

PATTERN - ORIENTED
SOFTWARE ARCHITECTURE
A SYSTEM OF PATTERNS
Frank Buschmann
Regine Meunier
Hans Rohnert
Peter Sommerlad
Michael Stal
WILEY
Douglas Schmidt
Michael Stal
Hans Rohnert
Frank Buschmann
PATTERN-ORIENTED SOFTWARE ARCHITECTURE
Volume 2
Patterns for Concurrent and Networked Objects
WILEY
WILEY SERIES IN SOFTWARE DESIGN PATTERNS
Michael Kircher
Prashant Jain
PATTERN-ORIENTED SOFTWARE ARCHITECTURE
Volume 3
Patterns for Resource Management
WILEY SERIES IN SOFTWARE DESIGN PATTERNS
WILEY SERIES IN SOFTWARE DESIGN PATTERNS
PATTERN-ORIENTED SOFTWARE ARCHITECTURE
A Pattern Language for Distributed Computing
Volume 4
Frank Buschmann
Kevlin Henney
Douglas C. Schmidt
WILEY SERIES IN SOFTWARE DESIGN PATTERNS
PATTERN-ORIENTED SOFTWARE ARCHITECTURE
On Patterns and Pattern Languages
Volume 5
Frank Buschmann
Kevlin Henney
Douglas C. Schmidt

I don't make stupid mistakes.
Only very, very clever ones.

John Peel

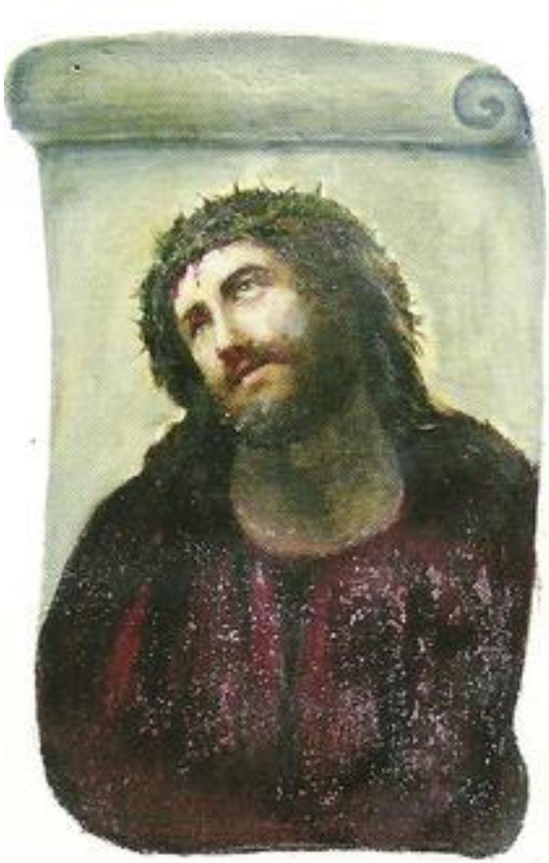
Failure is a far better teacher than success.

Philip Delves Broughton

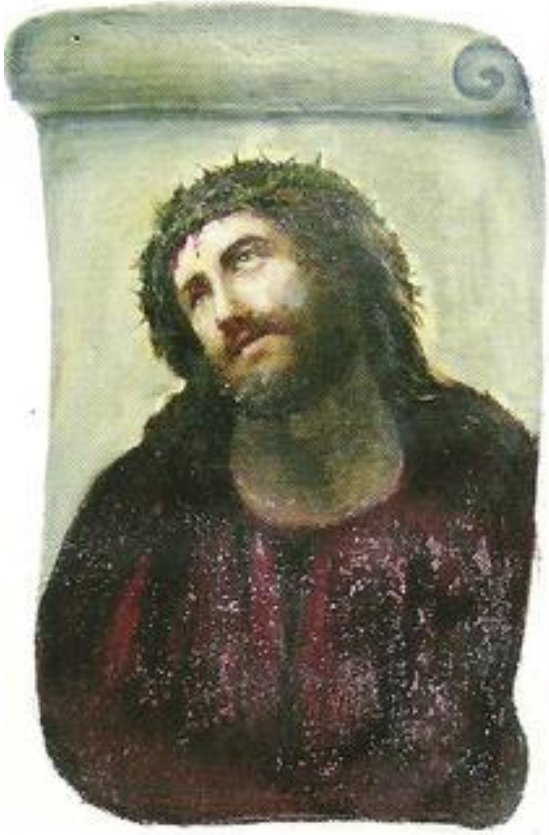
http://www.ft.com/cms/s/0/f33f5508-f010-11e0-bc9d-00144feab49a.html

If you want to learn how to build a house, build a house. Don't ask anybody, just build a house.

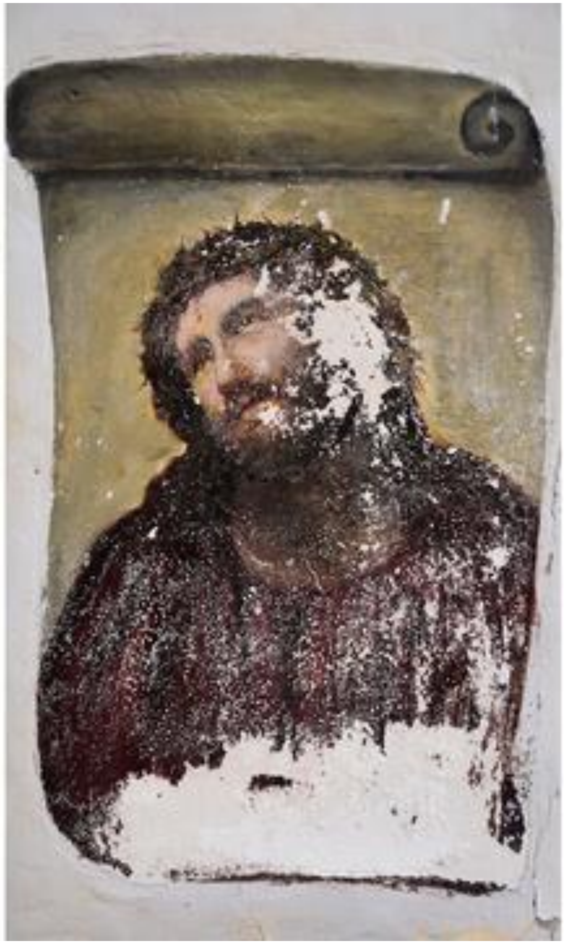
Christopher Walken

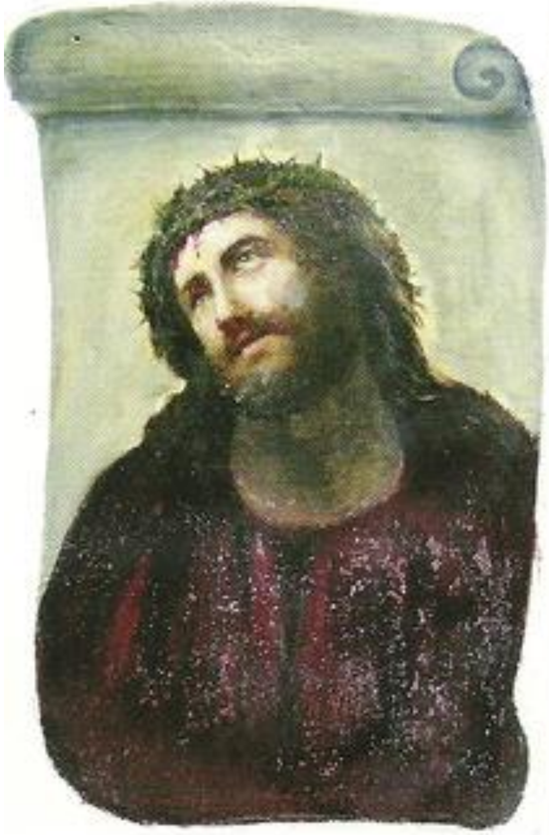


Ecce Homo de Elías García Martínez.

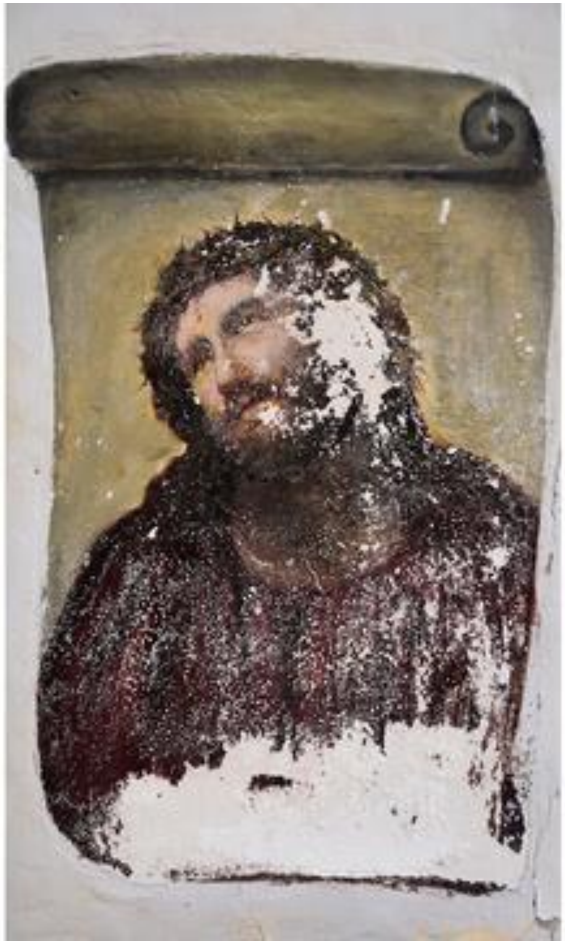


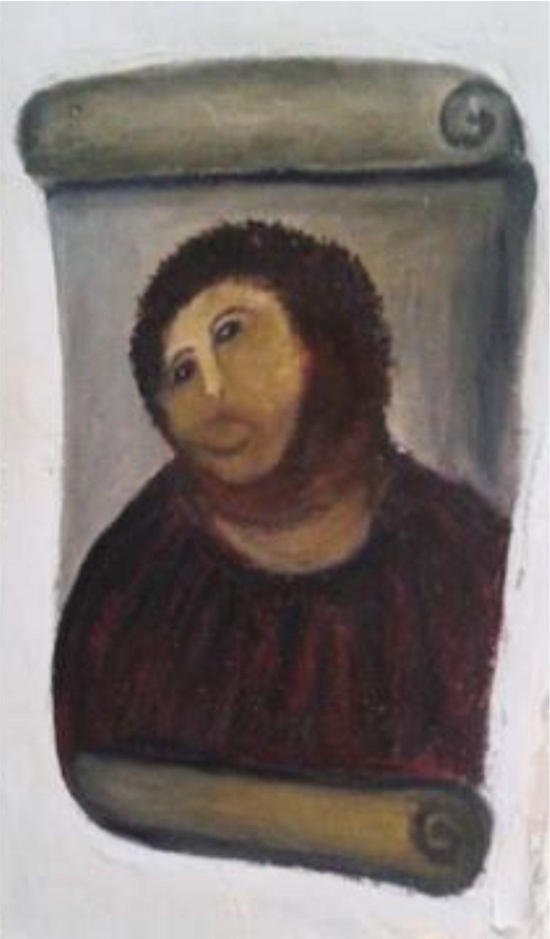
Ecce Homo de Elías García Martínez.





Ecce Homo de Elías García Martínez.





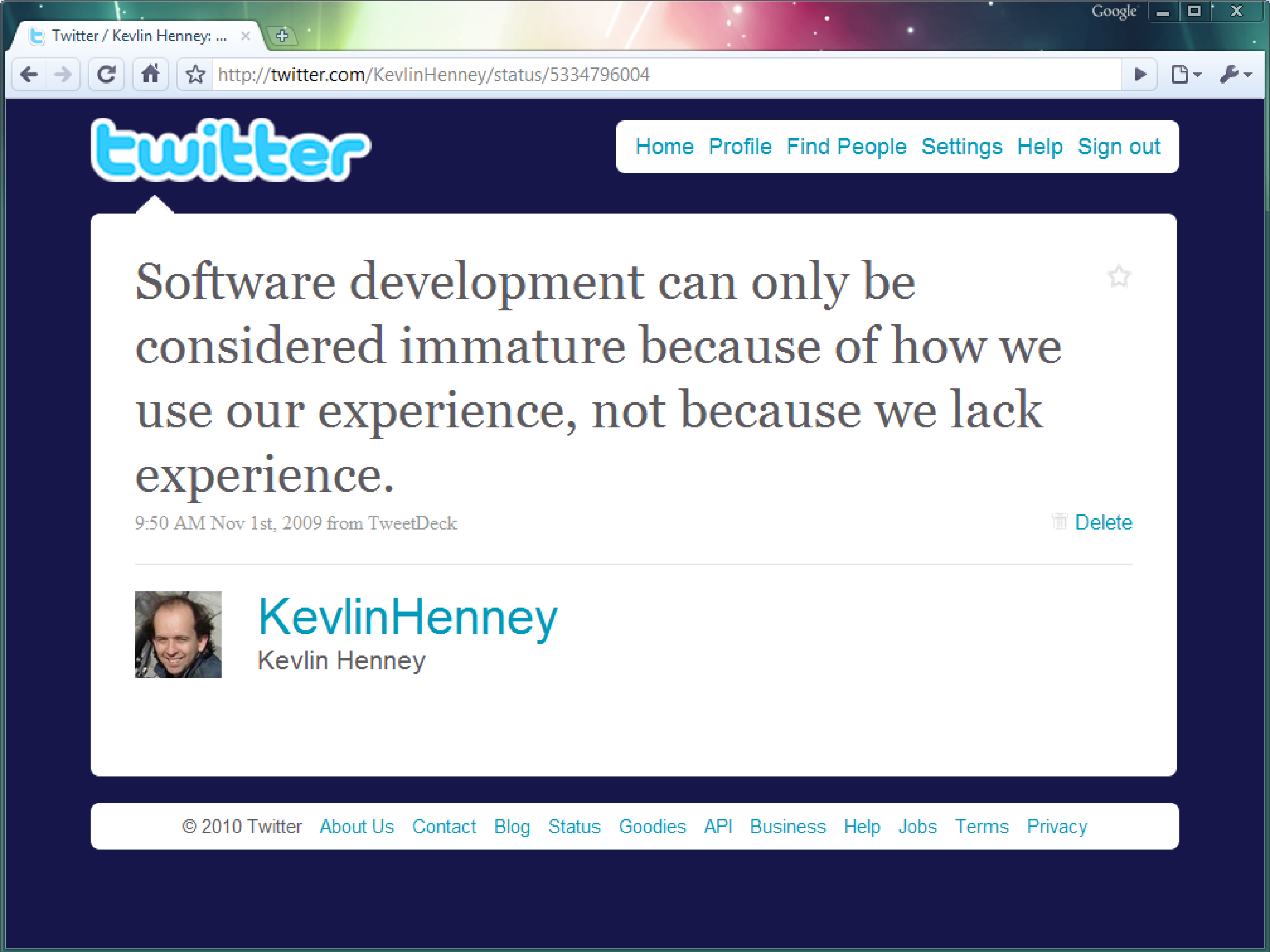
Programming is difficult business. It should never be undertaken in ignorance.

Douglas Crockford

JavaScript: The Good Parts

What experience and history teach is that nations and governments have never learned anything from history.

Georg Wilhelm Friedrich Hegel


Twitter / Kevlin Henney: ...
Google
http://twitter.com/KevlinHenney/status/5334796004
twitter
Home
Profile
Find People
Settings
Help
Sign out

Software development can only be considered immature because of how we use our experience, not because we lack experience.

9:50 AM Nov 1st, 2009 from TweetDeck

KevlinHenney

Kevlin Henney

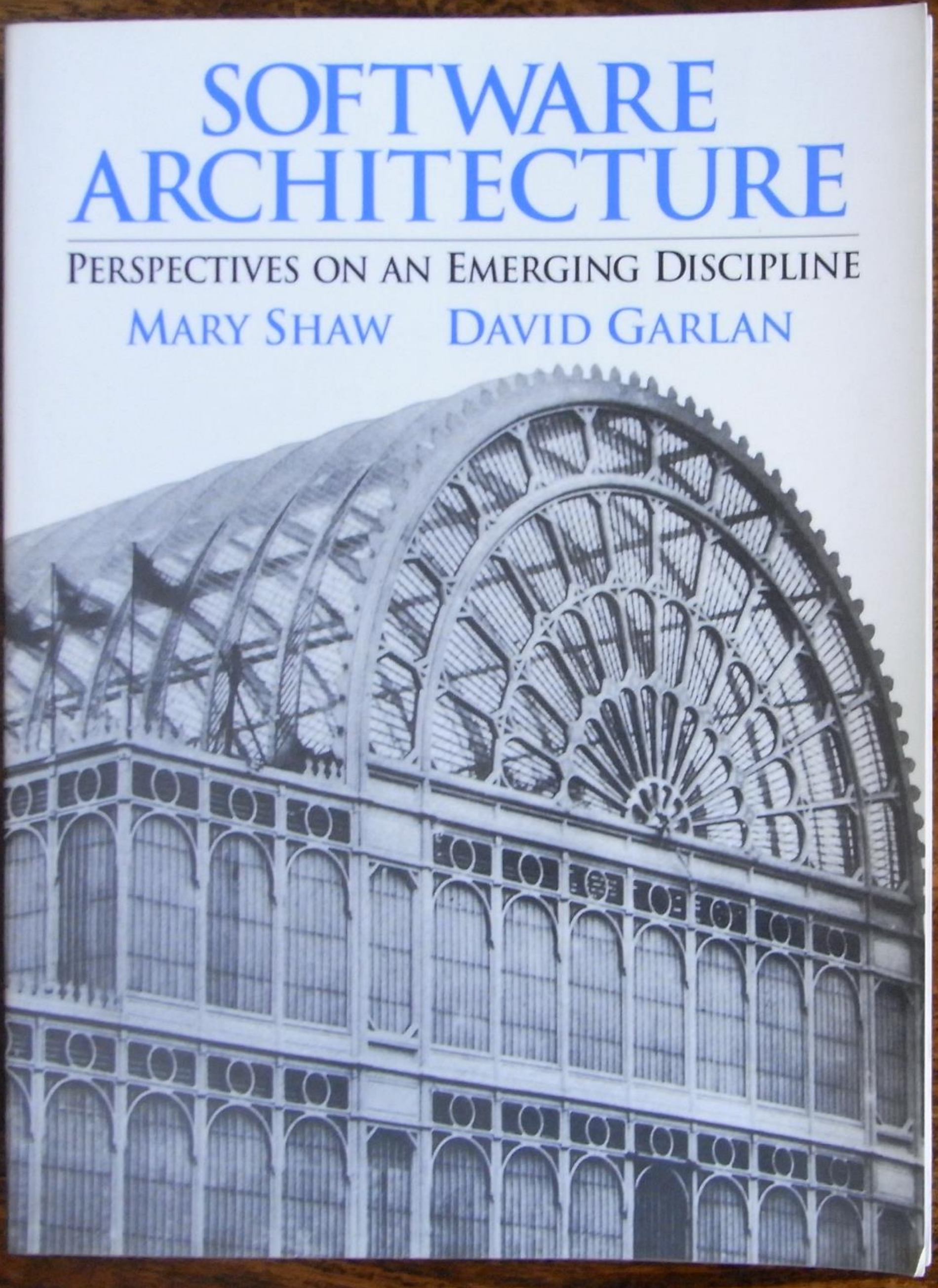
© 2010 Twitter

Wise men profit more from fools than fools from wise men; for the wise men shun the mistakes of fools, but fools do not imitate the successes of the wise.

Cato the Elder

Mark Pagel at the University of Reading, UK, doubts that hominins before *Homo sapiens* had what it takes to innovate and exchange ideas, even if they wanted to. He draws a comparison with chimps, which can make crude stone tools but lack technological progress. They mostly learn by trial and error, he says, whereas we learn by watching each other, and we know when something is worth copying.

http://www.newscientist.com/article/mg21328571.400-puzzles-of-evolution-why-was-technological-development-so-slow.html



SOFTWARE ARCHITECTURE

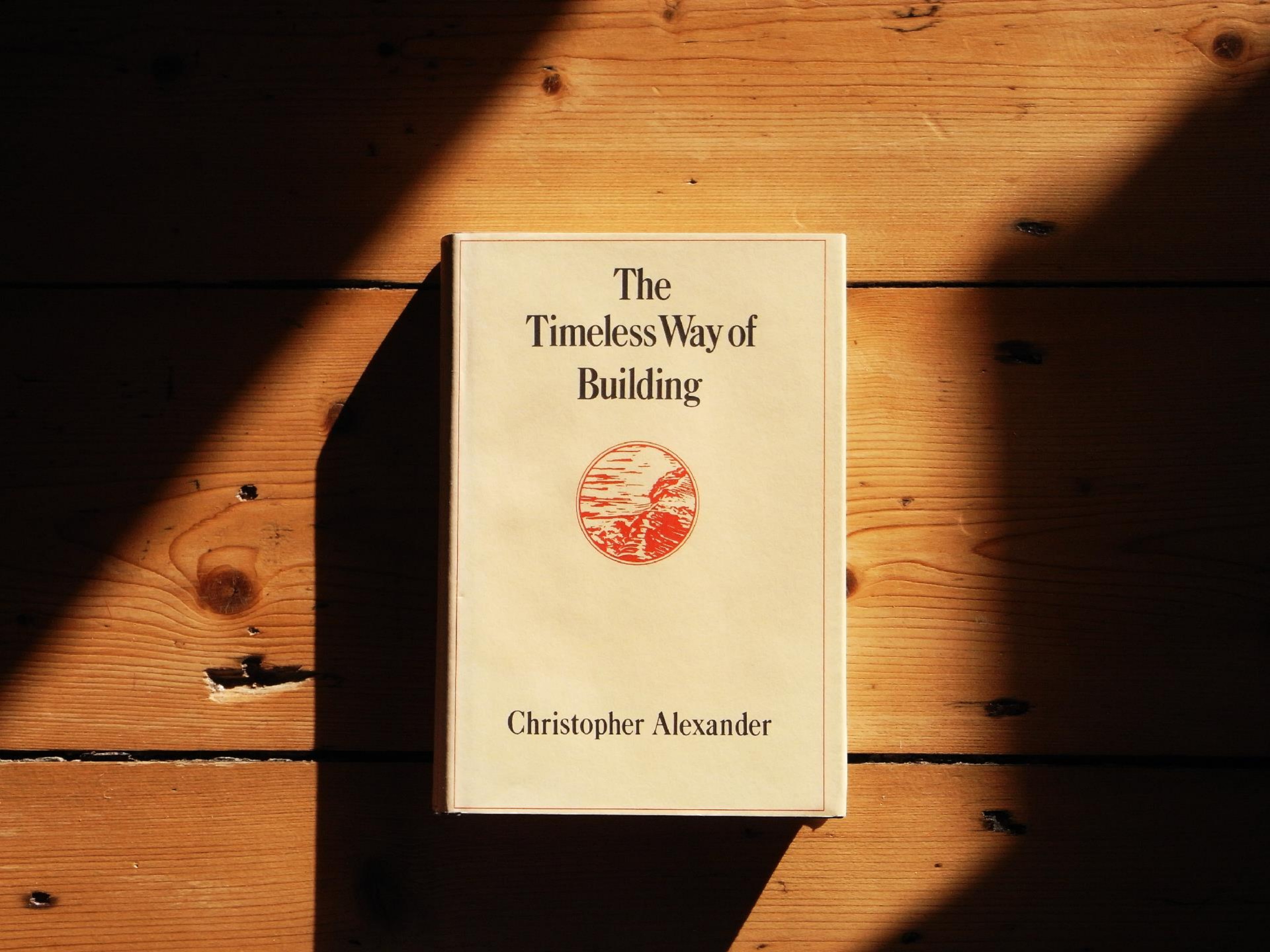
PERSPECTIVES ON AN EMERGING DISCIPLINE

MARY SHAW DAVID GARLAN

One of the hallmarks of architectural design is the use of idiomatic patterns of system organization. Many of these patterns — or architectural styles — have been developed over the years as system designers recognized the value of specific organizational principles and structures for certain classes of software.

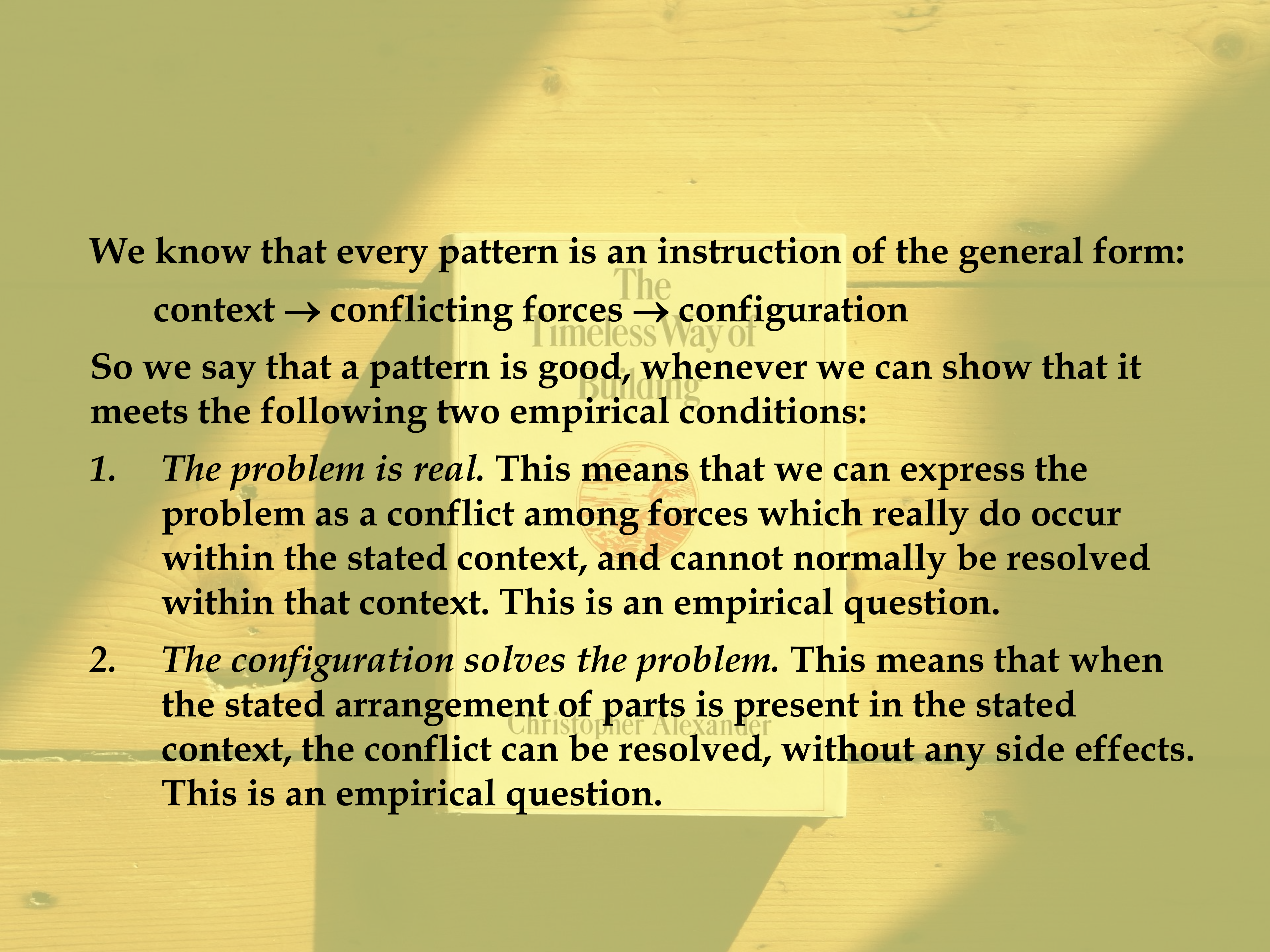
面向模式的软件架构 卷5 模式与模式语言 Frank Buschmann Kevlin Henney Douglas C. Schmidt 人民邮电出版社
面向模式的软件架构 卷4 分布式计算的模式语言 Frank Buschmann Kevlin Henney Douglas C. Schmidt 人民邮电出版社
面向模式的软件架构 卷5 模式与模式语言 Frank Buschmann Kevlin Henney Douglas C. Schmidt 人民邮电出版社
PATTERNS FOR PARALLEL SOFTWARE DESIGN
WHERE CODE AND CONTENT MEET
Buschmann Meunier Rohnert Sommerlad Stal PATTERN-ORIENTED SOFTWARE ARCHITECTURE A SYSTEM OF PATTERNS WILEY
Schmidt Stal Rohnert Buschmann Volume 2 PATTERN-ORIENTED SOFTWARE ARCHITECTURE WILEY
Kircher Jain PATTERN-ORIENTED SOFTWARE ARCHITECTURE Vol 3 WILEY
Buschmann Henney Schmidt Volume 4 PATTERN-ORIENTED SOFTWARE ARCHITECTURE WILEY
Buschmann Henney Schmidt Volume 5 PATTERN-ORIENTED SOFTWARE ARCHITECTURE WILEY
Schumacher Fernandez-Buglioni Hybertson Buschmann Sommerlad SECURITY PATTERNS Integrating Security and Systems Engineering WILEY
Dyson Longshaw ARCHITECTING ENTERPRISE SOLUTIONS WILEY
Borchers A PATTERN APPROACH TO INTERACTION DESIGN WILEY
Völter Kircher Zdun REMOTING PATTERNS WILEY
Völter Schmid Wolff SERVER COMPONENT PATTERNS Component Infrastructures Illustrated with EJB WILEY
PATTERN LANGUAGES OF PROGRAM DESIGN EDITED BY COPLIEN SCHMIDT ADDISON WESLEY
PATTERN LANGUAGES OF PROGRAM DESIGN 2 VLISSIDES COPLIEN • KERTH ADDISON WESLEY
SOFTWARE PATTERNS SERIES Pattern Languages of Program Design 3 MARTIN / RIEHLE / BUSCHMANN ADDISON WESLEY
SOFTWARE PATTERNS SERIES Pattern Languages of Program Design 4 HARRISON / FOOTE / ROHNERT Addison Wesley
Design Patterns CD Gamma • Helm Johnson • Vlissides Addison Wesley

PATTERN
SHOP
Caution
Uneven Floor
Patterns



The Timeless Way of Building

Christopher Alexander

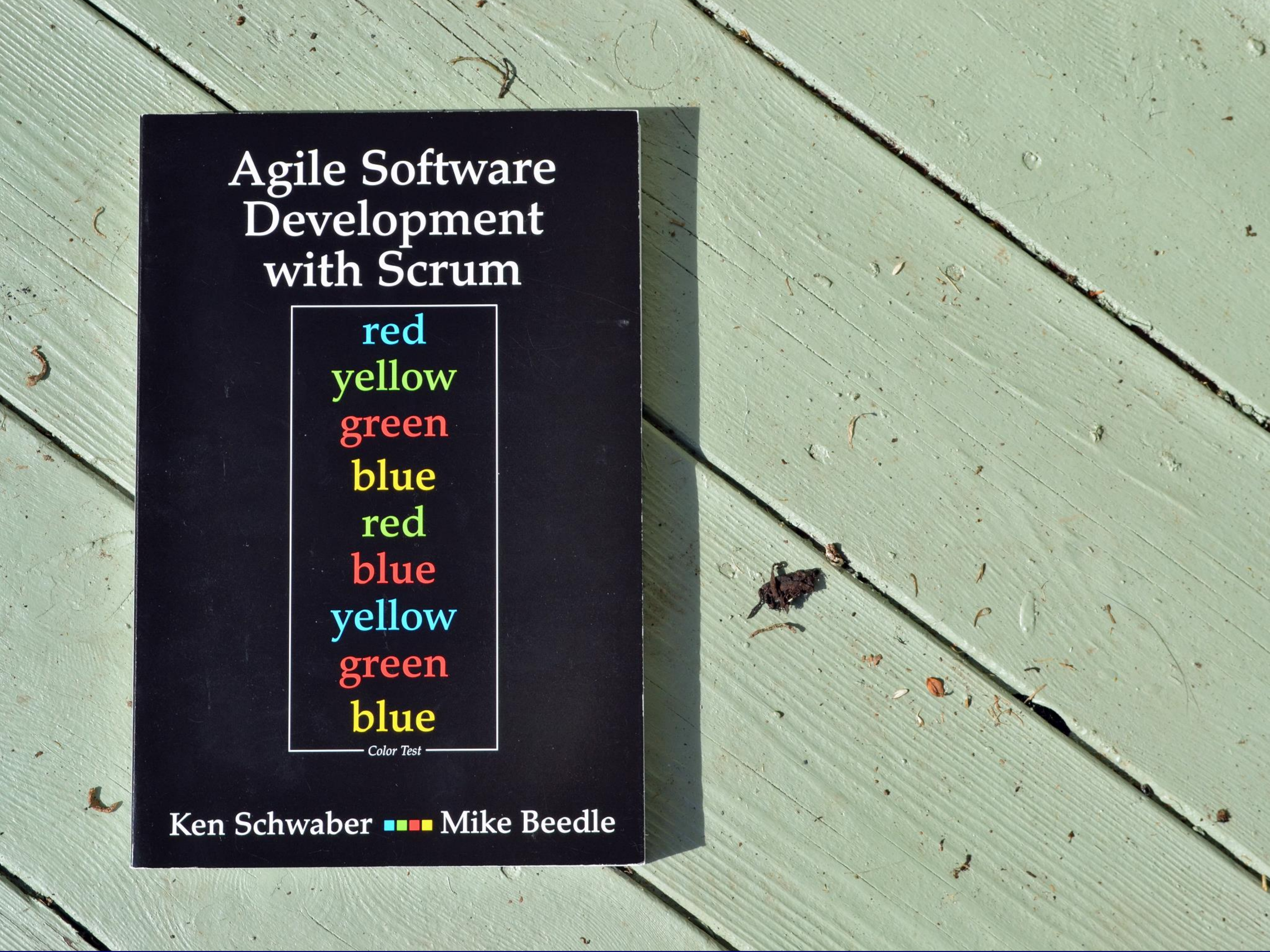


We know that every pattern is an instruction of the general form:

context → conflicting forces → configuration

So we say that a pattern is good, whenever we can show that it meets the following two empirical conditions:

1. *The problem is real.* This means that we can express the problem as a conflict among forces which really do occur within the stated context, and cannot normally be resolved within that context. This is an empirical question.
2. *The configuration solves the problem.* This means that when the stated arrangement of parts is present in the stated context, the conflict can be resolved, without any side effects. This is an empirical question.



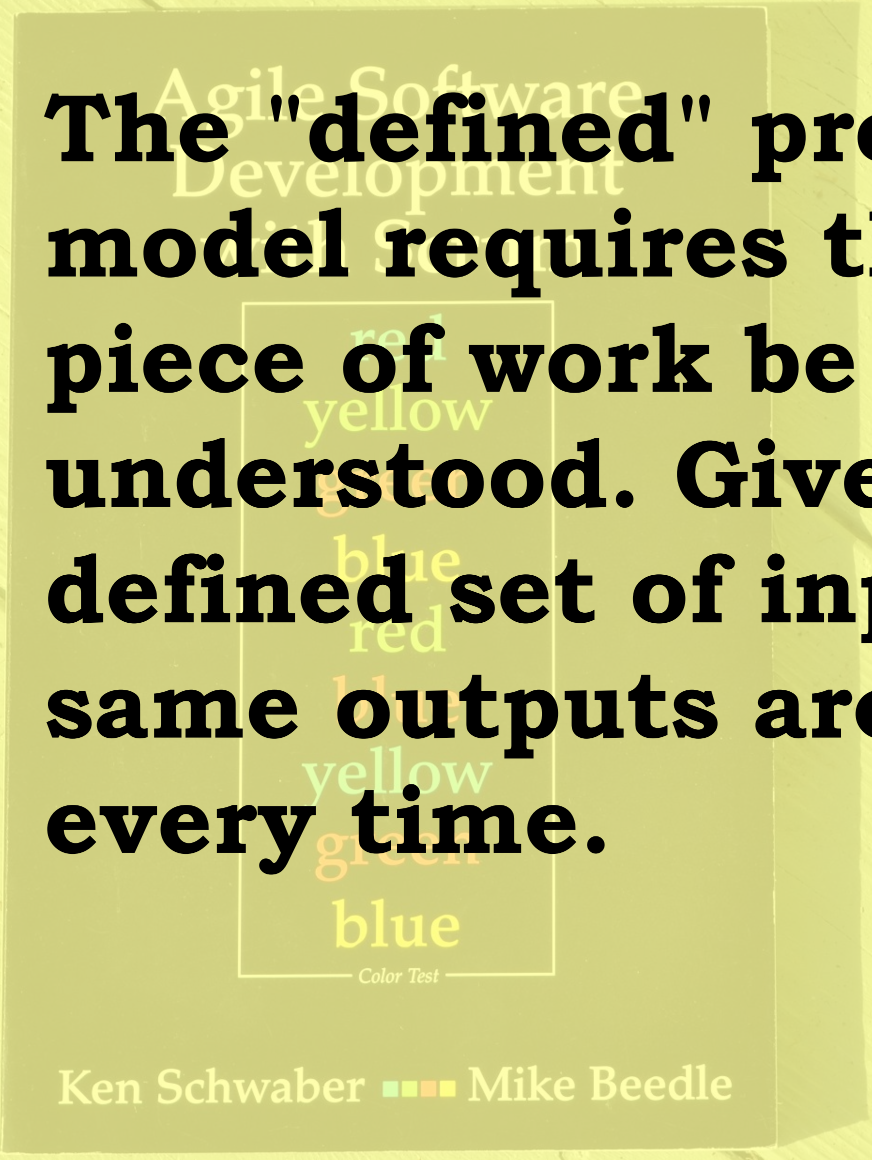
Agile Software Development with Scrum

red
yellow
green
blue
red
blue
yellow
green
blue

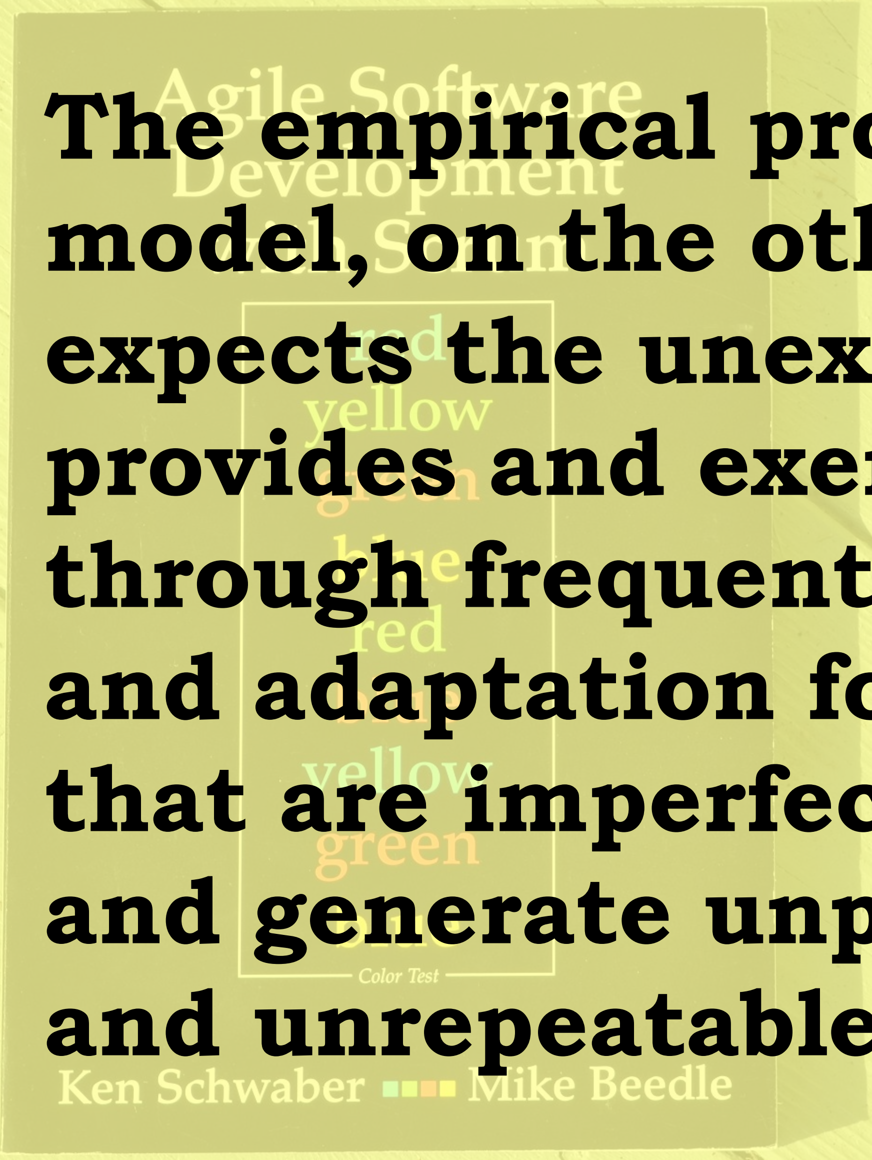
Color Test

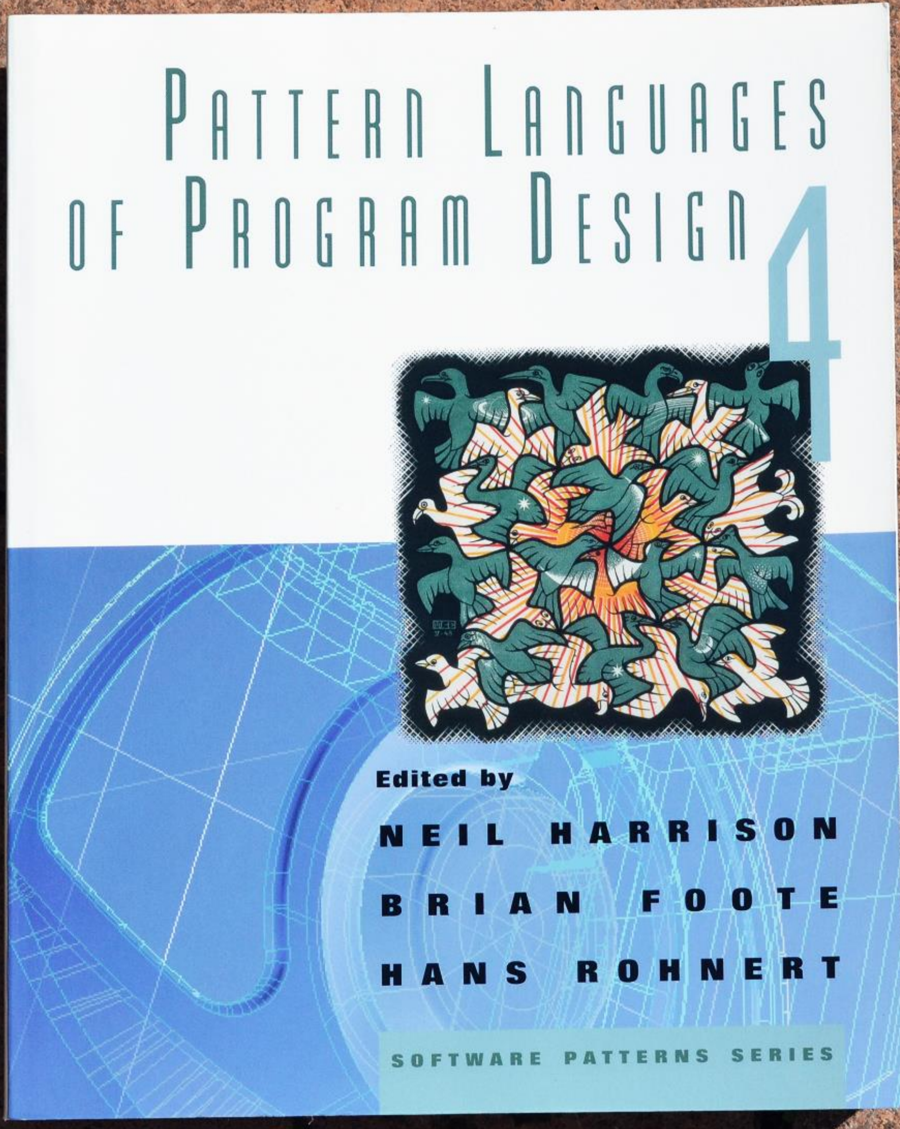
Ken Schwaber Mike Beedle

The "defined" process control model requires that every piece of work be completely understood. Given a well-defined set of inputs, the same outputs are generated every time.



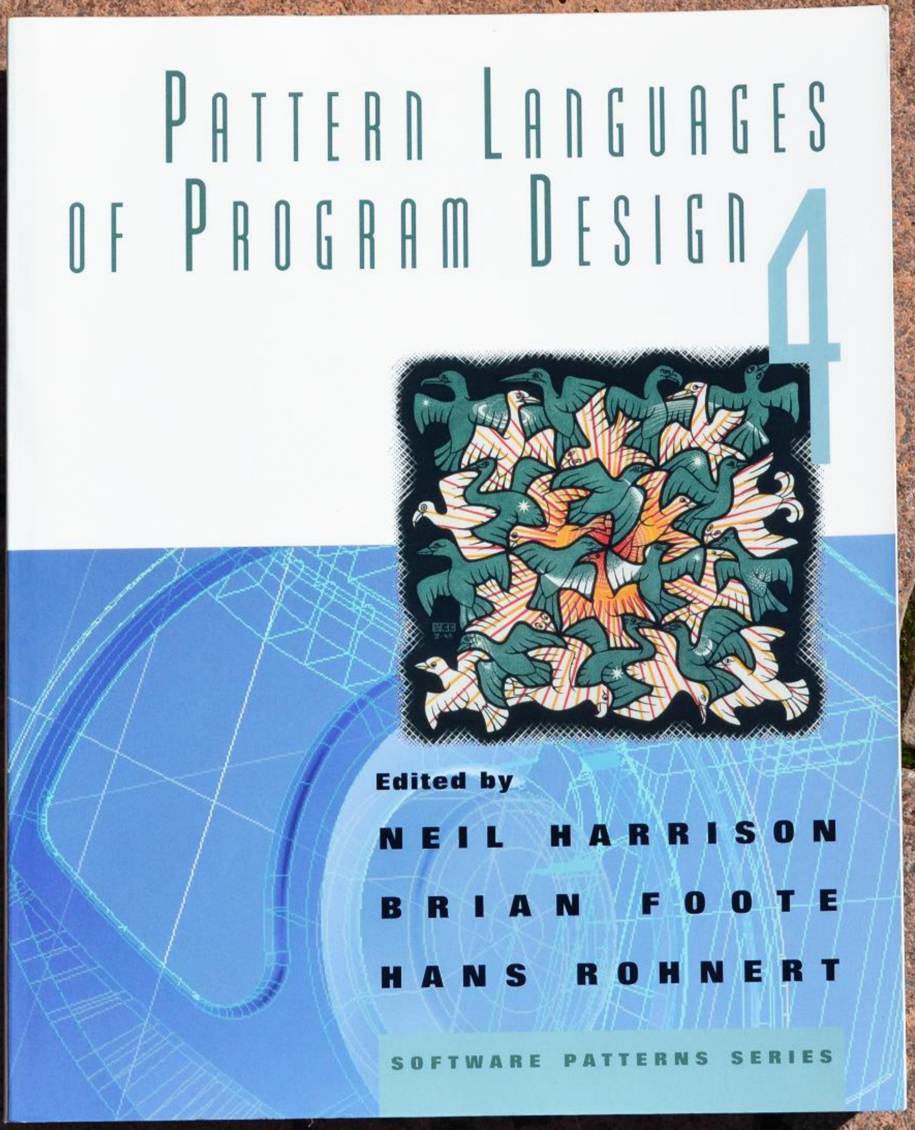
The empirical process control model, on the other hand, expects the unexpected. It provides and exercises control through frequent inspection and adaptation for processes that are imperfectly defined and generate unpredictable and unrepeatable results.





SCRUM: A Pattern Language for Hyperproductive Software Development Teams

Mike Beedle, Martine Devos,
Yonat Sharon,Ken Schwaber,
and Jeff Sutherland



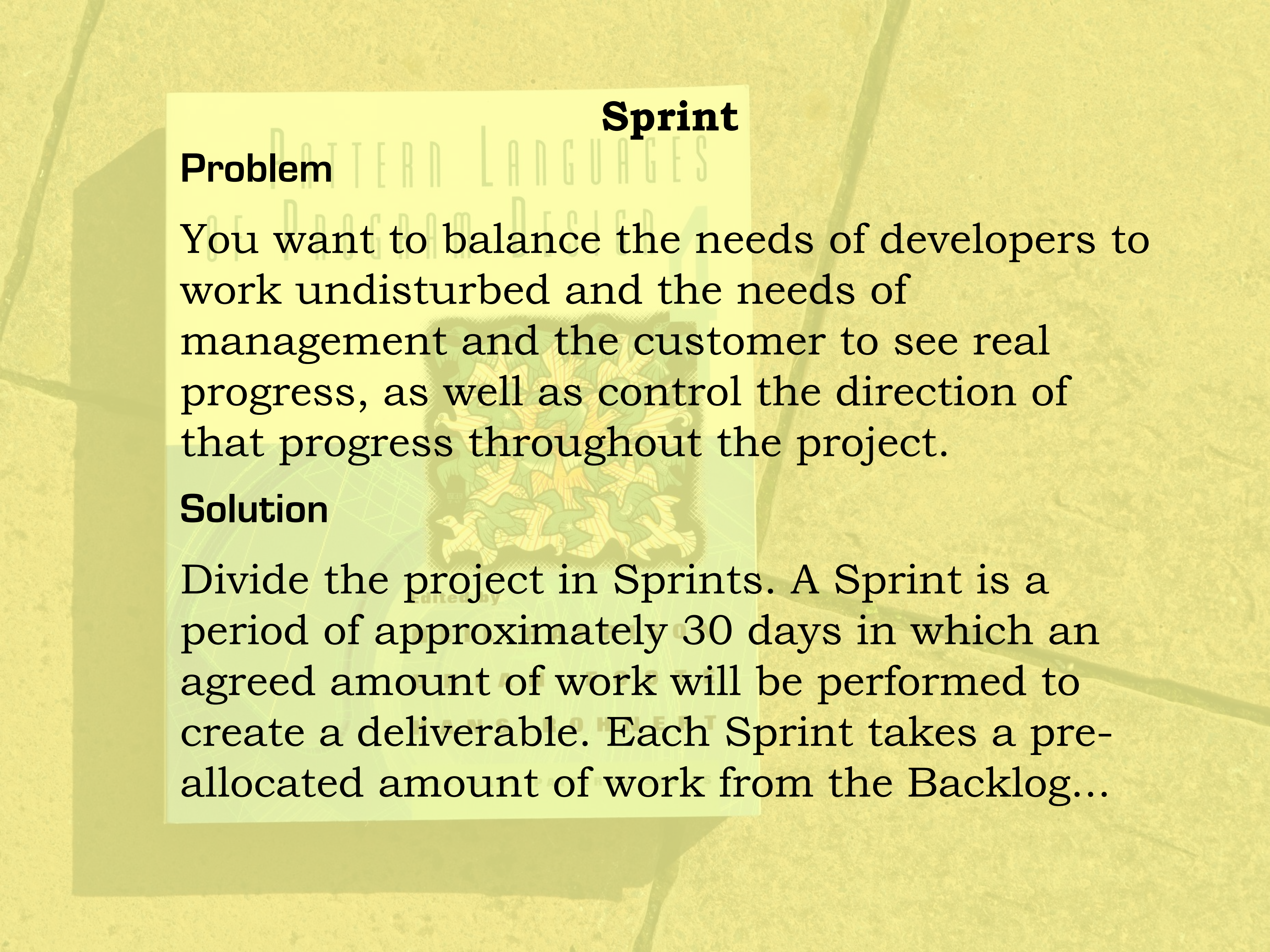
SCRUM Master

Sprint

Backlog

SCRUM Meetings

Demo After Sprint



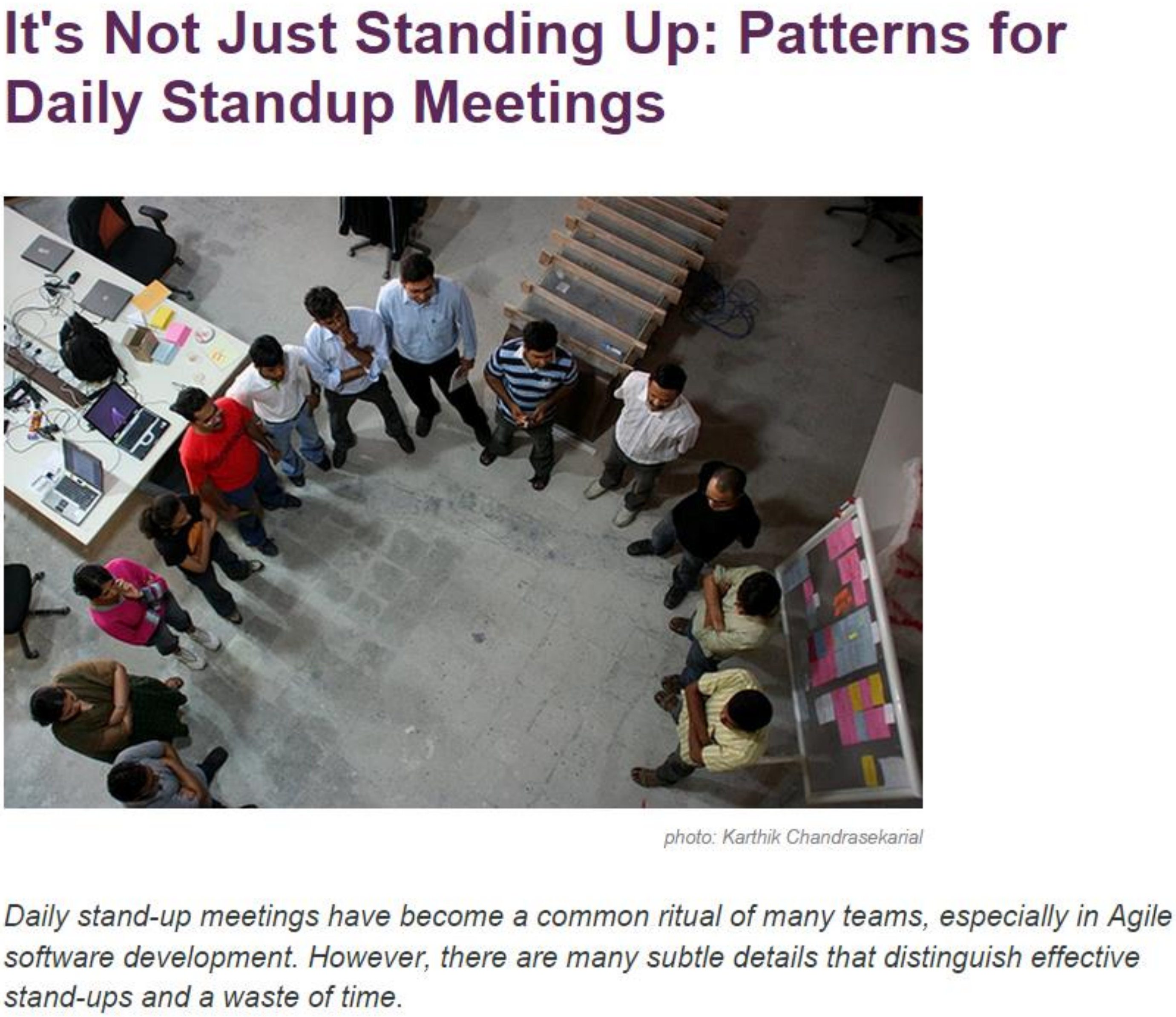
Sprint

Problem

You want to balance the needs of developers to work undisturbed and the needs of management and the customer to see real progress, as well as control the direction of that progress throughout the project.

Solution

Divide the project in Sprints. A Sprint is a period of approximately 30 days in which an agreed amount of work will be performed to create a deliverable. Each Sprint takes a pre-allocated amount of work from the Backlog...

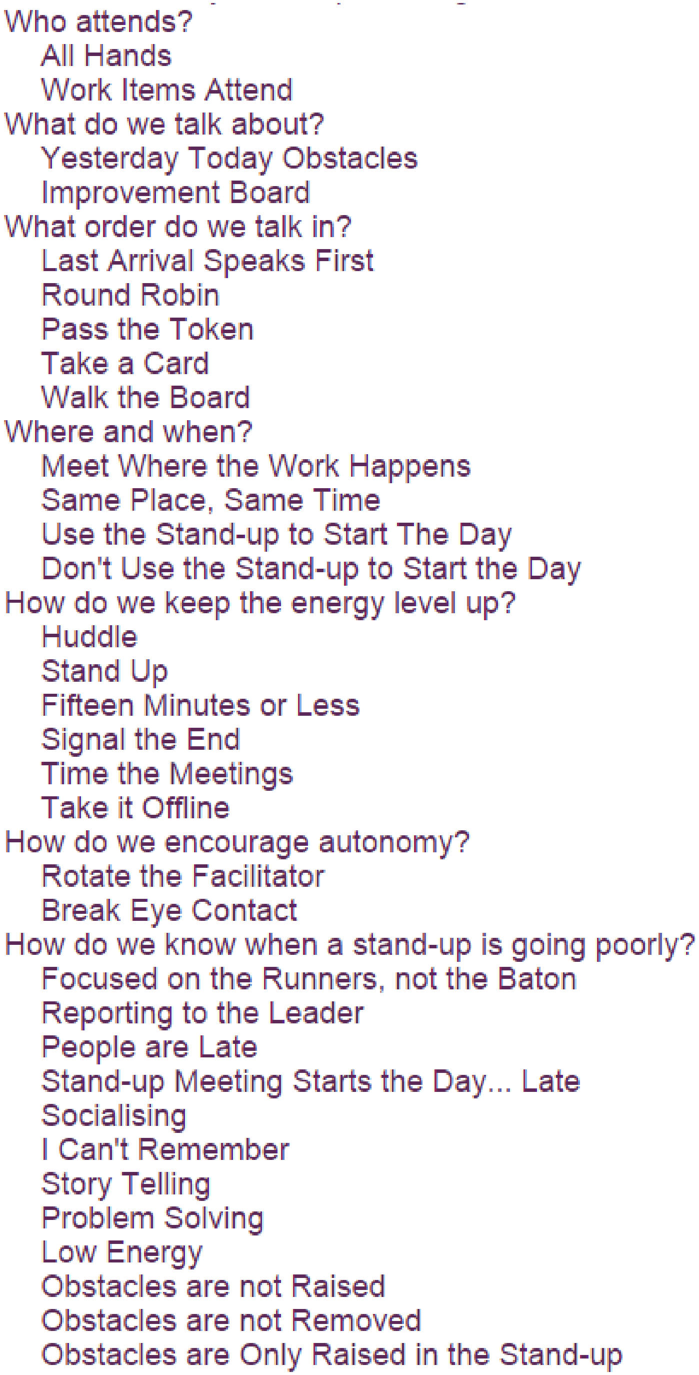


It's Not Just Standing Up: Patterns for Daily Standup Meetings

photo: Karthik Chandrasekarial

Daily stand-up meetings have become a common ritual of many teams, especially in Agile software development. However, there are many subtle details that distinguish effective stand-ups and a waste of time.

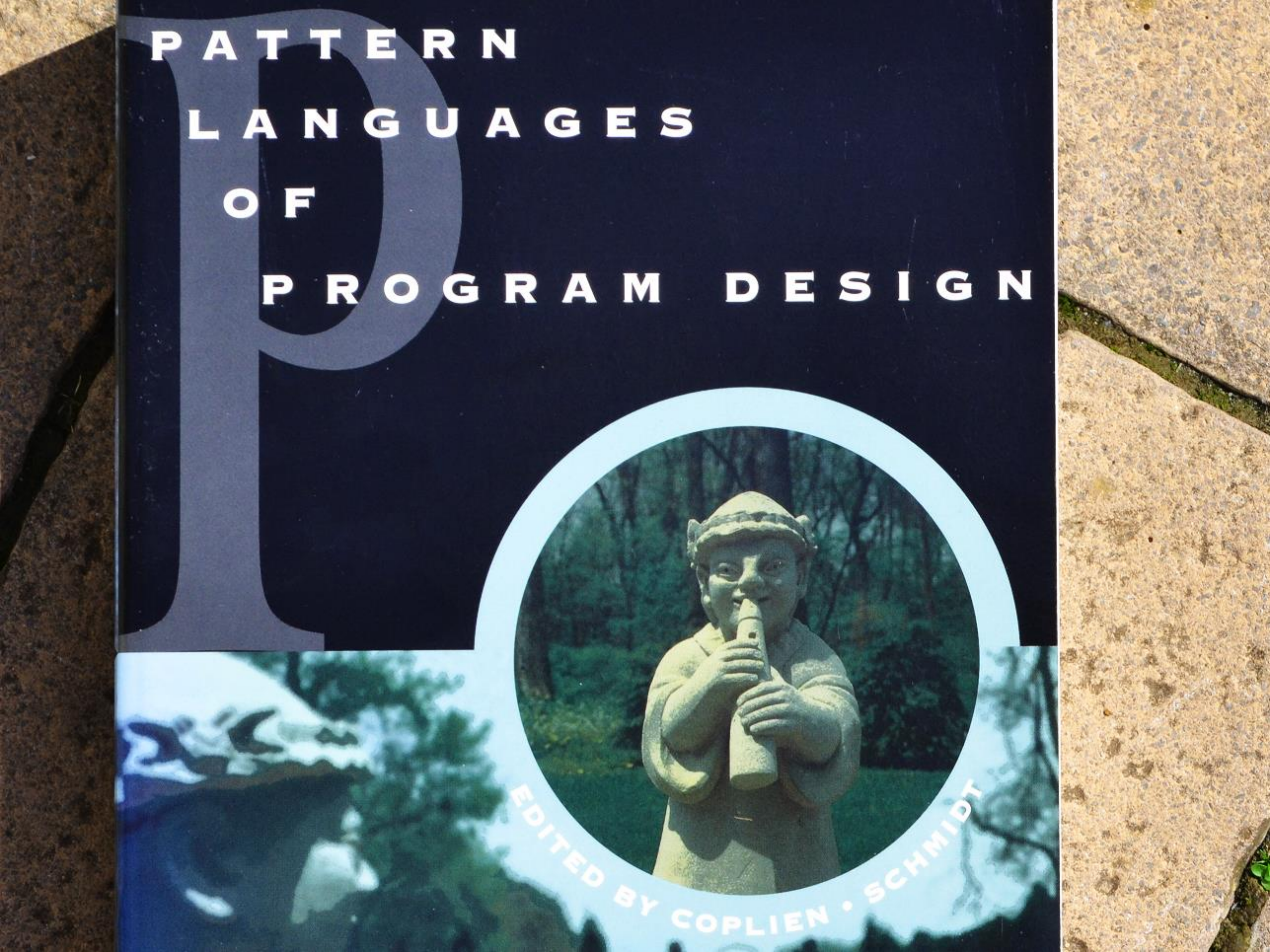
Jason Yip
http://martinfowler.com/articles/itsNotJustStandingUp.html



- Who attends?
    - All Hands
    - Work Items Attend
- What do we talk about?
    - Yesterday Today Obstacles
    - Improvement Board
- What order do we talk in?
    - Last Arrival Speaks First
    - Round Robin
    - Pass the Token
    - Take a Card
    - Walk the Board
- Where and when?
    - Meet Where the Work Happens
    - Same Place, Same Time
    - Use the Stand-up to Start The Day
    - Don't Use the Stand-up to Start the Day
- How do we keep the energy level up?
    - Huddle
    - Stand Up
    - Fifteen Minutes or Less
    - Signal the End
    - Time the Meetings
    - Take it Offline
- How do we encourage autonomy?
    - Rotate the Facilitator
    - Break Eye Contact
- How do we know when a stand-up is going poorly?
    - Focused on the Runners, not the Baton
    - Reporting to the Leader
    - People are Late
    - Stand-up Meeting Starts the Day... Late
    - Socialising
    - I Can't Remember
    - Story Telling
    - Problem Solving
    - Low Energy
    - Obstacles are not Raised
    - Obstacles are not Removed
    - Obstacles are Only Raised in the Stand-up

Jason Yip

http://martinfowler.com/articles/itsNotJustStandingUp.html



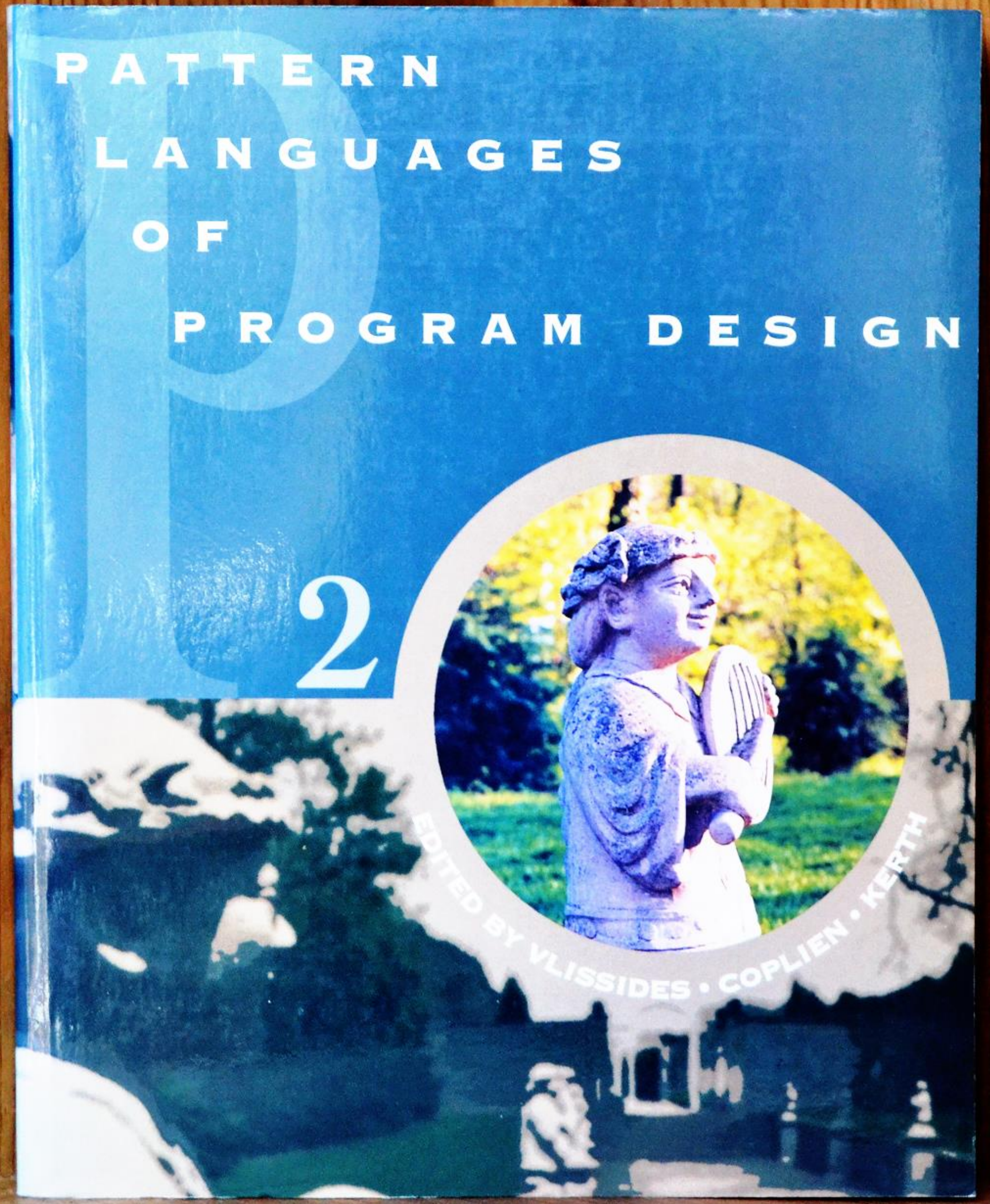
PATTERN LANGUAGES OF PROGRAM DESIGN

EDITED BY COPLIEN • SCHMIDT

Developer Controls Process

Place the Developer role at a hub of the process for a given feature. A feature is a unit of system functionality, implemented largely in software, that can be separately marketed and for which customers are willing to pay. The Developer is the process information clearinghouse. Responsibilities of Developers include understanding requirements, reviewing the solution structure and algorithm with peers, building the implementation, and unit testing.

A Generative Development-Process Pattern Language
James O Coplien


PATTERN LANGUAGES OF PROGRAM DESIGN
2
EDITED BY VLISSIDES • COPLIEN • KERTH

EPISODES: A Pattern Language of Competitive Development

Ward Cunningham

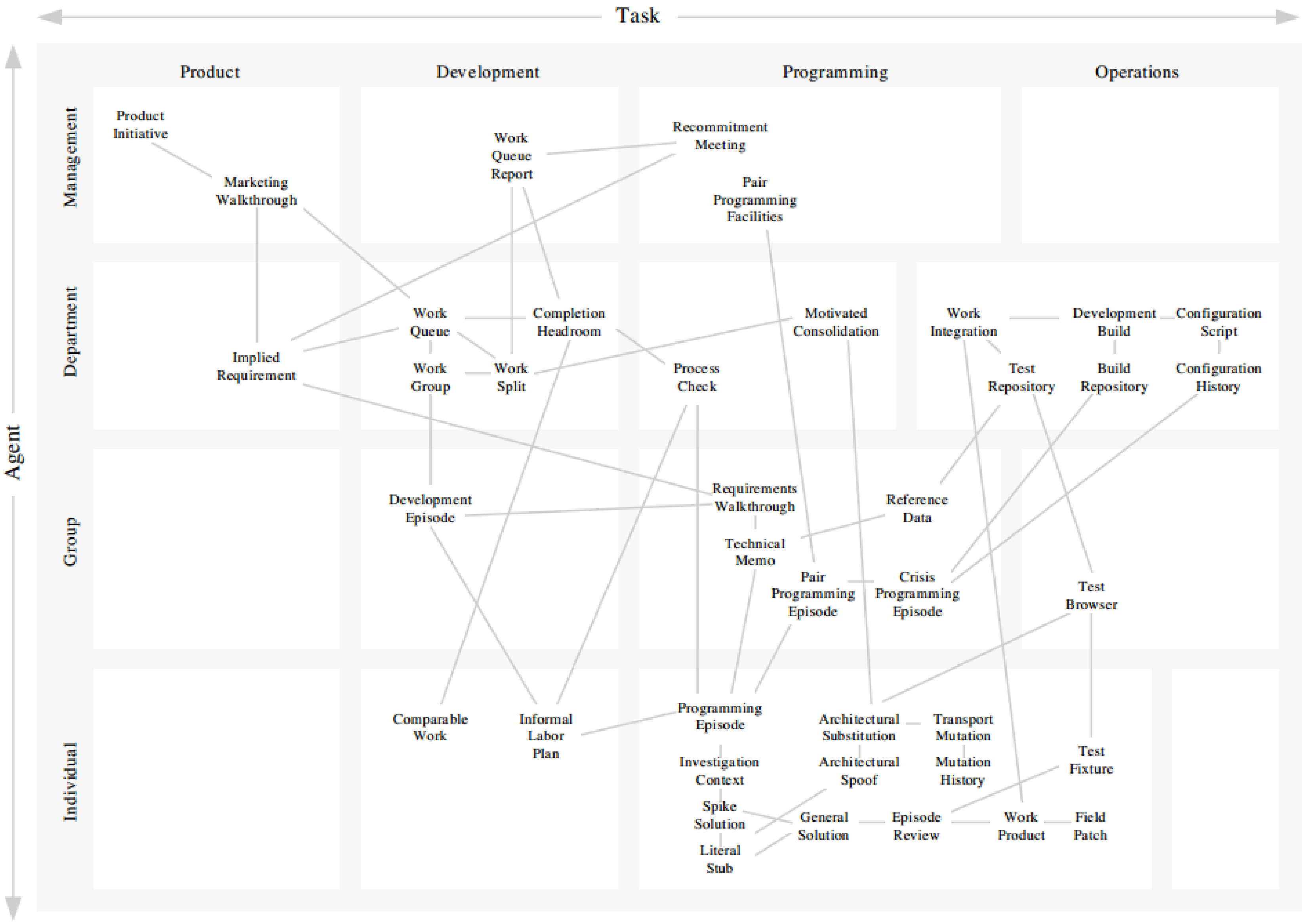
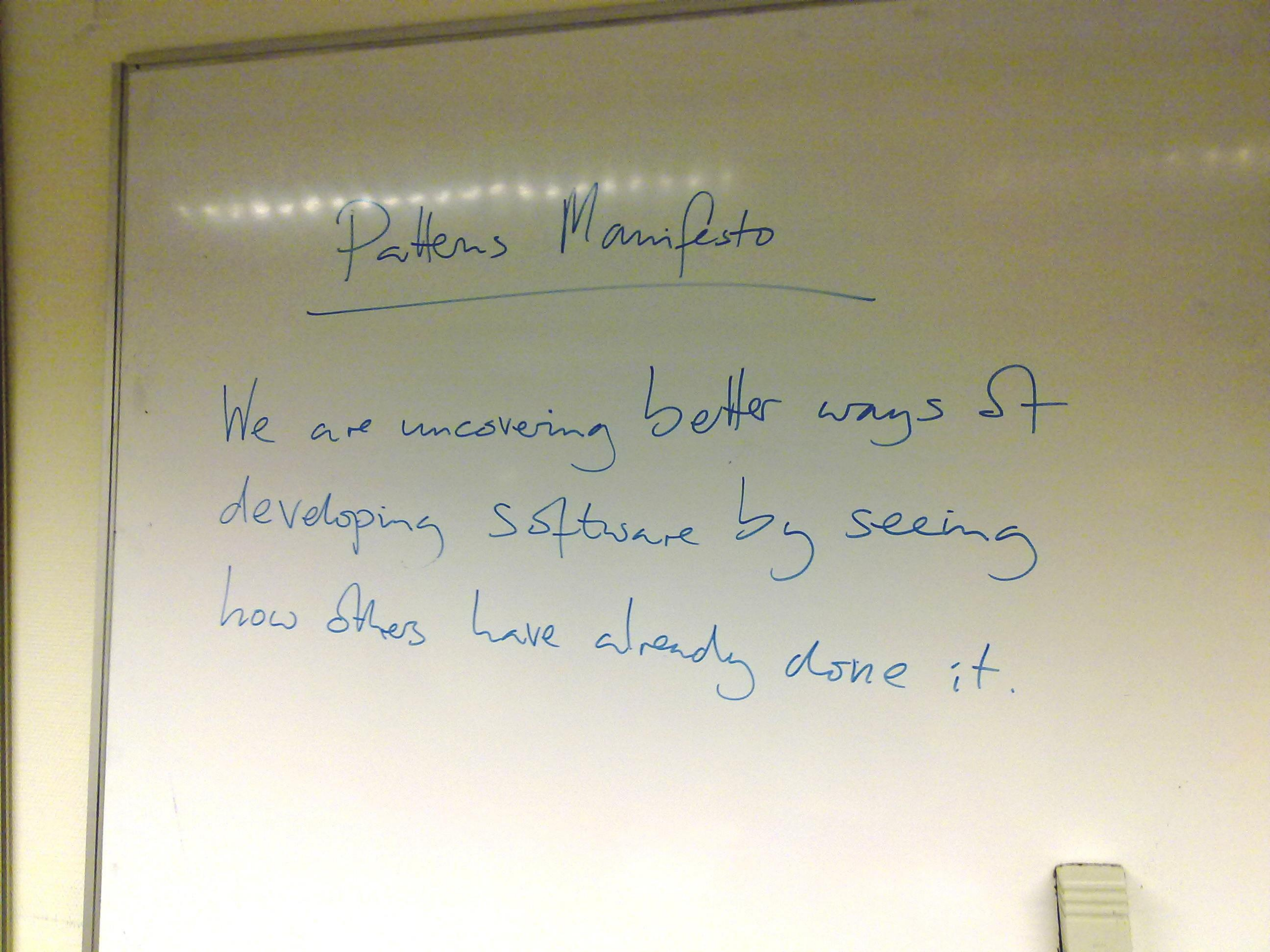


Figure 1. Map of EPISODE patterns and their relations.

Command Pattern
ctor executes commands from
ue in a separate thread
ommand is sent to the actor
need to know when the command
been executed
d a "Blocking Command" to the
or. Wait for the blocking command
be executed. When it has been
cuted you can be sure that the
vious command has also been
ecuted.
Example
Test System Actor
Loop continuations
Peppes Pizza
1970
American way
New Orleans
Connecticut
Margherita
2222 5555
www.peppes.no



Patterns Manifesto

We are uncovering better ways of developing software by seeing how others have already done it.