Extracted from:

Lean from the Trenches

Managing Large-Scale Projects with Kanban

This PDF file contains pages extracted from *Lean from the Trenches*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

Copyright © 2010 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

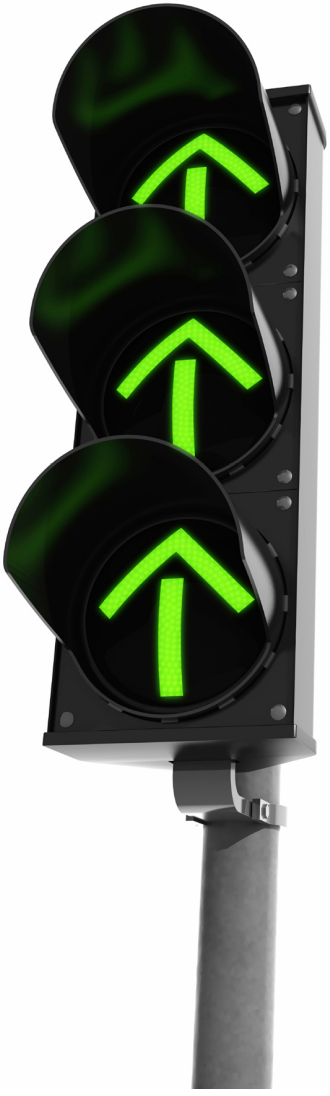
The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The Pragmatic Programmers

Lean from the Trenches

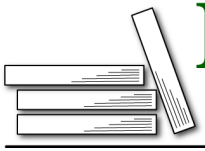
Managing Large-Scale Projects with Kanban



Henrik Kniberg

Foreword by Kent Beck

Edited by Kay Keppler



Pragmatic Bookshelf

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at *http://pragprog.com*.

The team that produced this book includes:

Kay Keppler (editor)
Potomac Indexing, LLC (indexer)
Kim Wimpsett (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Copyright © 2011 The Pragmatic Programmers, LLC.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.
ISBN-13: 978-1-934356-85-2
Printed on acid-free paper.
Book version: P1.0—December, 2011

Contents

Foreword . . . . . ?

Preface . . . . . ?

Part I — How We Work

1. About the Project . . . . . ?
   - 1.1 Timeline ?
   - 1.2 How We Sliced the Elephant ?
   - 1.3 How We Involved the Customer ?

2. Structuring the Teams . . . . . ?

3. Attending the Daily Cocktail Party . . . . . ?
   - 3.1 First Tier: Feature Team Daily Stand-up ?
   - 3.2 Second Tier: Sync Meetings per Specialty ?
   - 3.3 Third Tier: Project Sync Meeting ?

4. The Project Board . . . . . ?
   - 4.1 Our Cadences ?
   - 4.2 How We Handle Urgent Issues and Impediments ?

5. Scaling the Kanban Boards . . . . . ?

6. Tracking the High-Level Goal . . . . . ?

7. Defining Ready and Done . . . . . ?
   - 7.1 Ready for Development ?
   - 7.2 Ready for System Test ?
   - 7.3 How This Improved Collaboration ?

8. Handling Tech Stories . . . . . ?
8.1 Example 1: System Test Bottleneck ?
8.2 Example 2: Day Before the Release ?
8.3 Example 3: The 7-Meter Class ?

9. Handling Bugs . . . . . ?
9.1 Continuous System Test ?
9.2 Fix the Bugs Immediately! ?
9.3 Why We Limit the Number of Bugs in the Bug Tracker ?
9.4 Visualizing Bugs ?
9.5 Preventing Recurring Bugs ?

10. Continuously Improving the Process . . . . . ?
10.1 Team Retrospectives ?
10.2 Process Improvement Workshops ?
10.3 Managing the Rate of Change ?

11. Managing Work in Progress . . . . . ?
11.1 Using WIP Limits ?
11.2 Why WIP Limits Apply Only to Features ?

12. Capturing and Using Process Metrics . . . . . ?
12.1 Velocity (Features per Week) ?
12.2 Why We Don't Use Story Points ?
12.3 Cycle Time (Weeks per Feature) ?
12.4 Cumulative Flow ?
12.5 Process Cycle Efficiency ?

13. Planning the Sprint and Release . . . . . ?
13.1 Backlog Grooming ?
13.2 Selecting the Top Ten Features ?
13.3 Why We Moved Backlog Grooming Out of the Sprint Planning Meeting ?
13.4 Planning the Release ?

14. How We Do Version Control . . . . . ?
14.1 No Junk on the Trunk ?
14.2 Team Branches ?
14.3 System Test Branch ?

15. Why We Use Only Physical Kanban Boards . . . . . ?

16. **What We Learned** . . . . . ?
16.1 Know Your Goal ?
16.2 Experiment ?
16.3 Embrace Failure ?
16.4 Solve Real Problems ?
16.5 Have Dedicated Change Agents ?
16.6 Involve People ?

Part II — A Closer Look at the Techniques

17. **Agile and Lean in a Nutshell** . . . . . ?
17.1 Agile in a Nutshell ?
17.2 Lean in a Nutshell ?
17.3 Scrum in a Nutshell ?
17.4 XP in a Nutshell ?
17.5 Kanban in a Nutshell ?

18. **Reducing the Test Automation Backlog** . . . . . ?
18.1 What to Do About It ?
18.2 How to Improve Test Coverage a Little Bit Each Iteration ?
18.3 Step 1: List Your Test Cases ?
18.4 Step 2: Classify Each Test ?
18.5 Step 3: Sort the List in Priority Order ?
18.6 Step 4: Automate a Few Tests Each Iteration ?
18.7 Does This Solve the Problem? ?

19. **Sizing the Backlog with Planning Poker** . . . . . ?
19.1 Estimating Without Planning Poker ?
19.2 Estimating with Planning Poker ?
19.3 Special Cards ?

20. **Cause-Effect Diagrams** . . . . . ?
20.1 Solve Problems, Not Symptoms ?
20.2 The Lean Problem-Solving Approach: A3 Thinking ?
20.3 How to Use Cause-Effect Diagrams ?
20.4 Example 1: Long Release Cycle ?
20.5 Example 2: Defects Released to Production ?
20.6 Example 3: Lack of Pair Programming ?
20.7 Example 4: Lots of Problems ?

20.8 Practical Issues: How to Create and Maintain the Diagrams ?
20.9 Pitfalls ?
20.10 Why Use Cause-Effect Diagrams? ?

21. Final Words . . . . . ?

A1. Glossary: How We Avoid Buzzword Bingo . . . . . ?

Index . . . . . ?