

Extracted from:

# Ship It!

---

## A Practical Guide to Successful Software Projects

This PDF file contains pages extracted from Ship It!, one of the Pragmatic Starter Kit series of books for project teams. For more information, visit [http://www.pragmaticprogrammer.com/starter\\_kit](http://www.pragmaticprogrammer.com/starter_kit).

**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 © 2005 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.

# Contents

---

<b>Foreword</b>	<b>ix</b>
<b>Preface</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Habitual Excellence . . . . .	2
1.2 A Pragmatic Point of View . . . . .	3
1.3 Road Map . . . . .	5
1.4 Moving On . . . . .	7
1.5 How Should I Read This Book? . . . . .	7
<b>2 Tools and Infrastructure</b>	<b>12</b>
1. Develop in a Sandbox . . . . .	16
2. Manage Assets . . . . .	19
3. Script Your Build . . . . .	25
4. Build Automatically . . . . .	30
5. Track Issues . . . . .	35
6. Track Features . . . . .	39
7. Use a Testing Harness . . . . .	42
8. On Choosing Tools . . . . .	50
9. When Not to Experiment . . . . .	52
<b>3 Pragmatic Project Techniques</b>	<b>54</b>
10. Work from The List . . . . .	56
11. A Tech Leads . . . . .	68
12. Coordinate and Communicate Every Day . . . . .	77
13. Review All Code . . . . .	87
14. Send Code Change Notifications . . . . .	97
15. Putting It All Together . . . . .	102

<b>4</b>	<b>Tracer Bullet Development</b>	<b>103</b>
<b>5</b>	<b>Common Problems and How to Fix Them</b>	<b>127</b>
	16. Help! I've Inherited Legacy Code . . . . .	128
	17. Testing Untestable Code . . . . .	130
	18. Features Keep Breaking . . . . .	131
	19. Tests? We Stopped Using Them . . . . .	132
	20. But It Works for Me! . . . . .	134
	21. It Hurts When I Integrate Code . . . . .	135
	22. Can't Build the Product Reliably . . . . .	137
	23. Customers Are Unhappy . . . . .	138
	24. You've Got a Rogue Developer . . . . .	140
	25. Your Manager Is Unhappy . . . . .	143
	26. Team Doesn't Work Well Together . . . . .	145
	27. Can't Get "Buy-in" on Essential Points . . . . .	146
	28. The New Practice Didn't Help . . . . .	149
	29. There's No Automated Testing . . . . .	152
	30. We're Junior Developers, With No Mentor . . . . .	154
	31. We're on a "Death March" Project . . . . .	155
	32. Features Keep Creeping In . . . . .	157
	33. We're Never Done . . . . .	158
<b>A</b>	<b>Tip Summary</b>	<b>161</b>
<b>B</b>	<b>Source Code Management</b>	<b>163</b>
<b>C</b>	<b>Build Scripting Tools</b>	<b>166</b>
<b>D</b>	<b>Continuous Integration Systems</b>	<b>170</b>
<b>E</b>	<b>Issue Tracking Software</b>	<b>173</b>
<b>F</b>	<b>Development Methodologies</b>	<b>176</b>
<b>G</b>	<b>Testing Frameworks</b>	<b>179</b>
<b>H</b>	<b>Suggested Reading List</b>	<b>182</b>
	H.1 Bibliography . . . . .	185