

Extracted from:

The ThoughtWorks Anthology 2

More Essays on Software Technology and Innovation

This PDF file contains pages extracted from *The ThoughtWorks Anthology 2*, 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 © 2012 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

The ThoughtWorks® Anthology 2

More Essays on
Software Technology
and Innovation



Edited by Michael Swaine

The ThoughtWorks Anthology 2

More Essays on Software Technology and Innovation

Farooq Ali	Ola Bini
Brian Blignaut	James Bull
Neal Ford	Martin Fowler
Luca Grulla	Alistair Jones
Aman King	Patrick Kua
Marc McNeill	Julio Maia
Mark Needham	Sam Newman
Rebecca Parsons	Cosmin Stejerean

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

Contents

	Preface	?
	About the Authors	?
1.	Introduction	?
Part I — Languages		
2.	The Most Interesting Languages	?
	2.1 Why Languages Matter	?
	2.2 A Few Languages	?
	2.3 Wrapping Up	?
3.	Object-Oriented Programming: Objects over Classes	?
	3.1 Objects over Classes?	?
	3.2 Class Focus vs. Object Focus	?
	3.3 Object-Focused Languages	?
	3.4 Recap of Ideas	?
	3.5 Wrapping Up	?
4.	Functional Programming Techniques in Object-Oriented Languages	?
	4.1 Collections	?
	4.2 First-Class and Higher-Order Functions	?
	4.3 Minimizing State	?
	4.4 Other Ideas	?
	4.5 Wrapping Up	?

Part II — Testing

5.	Extreme Performance Testing	?
5.1	Stating the Problem	?
5.2	A Different Approach	?
5.3	Extreme Performance Testing Practices	?
5.4	How This Helps You	?
5.5	Wrapping Up	?
6.	Take Your JavaScript for a Test-Drive	?
6.1	The JavaScript Renaissance	?
6.2	Current JavaScript Approach and Problems	?
6.3	Separation of Concerns	?
6.4	Our Testing Approach	?
6.5	Continuous Integration	?
6.6	Tools	?
6.7	Wrapping Up	?
7.	Building Better Acceptance Tests	?
7.1	Fast Tests	?
7.2	Resilient Tests	?
7.3	Maintainable Tests	?
7.4	Making It Work	?
7.5	Wrapping Up	?

Part III — Issues in Software Development

8.	Modern Java Web Applications	?
8.1	The Past	?
8.2	Stateless Server	?
8.3	Container Considered Optional	?
8.4	Segmentation by Freshness	?
8.5	Post Redirect GET	?
8.6	Wrapping Up	?
9.	Taming the Integration Problem	?
9.1	The Continuous Integration Approach	?
9.2	Defining Integration Contracts	?
9.3	Metrics and Visibility	?
9.4	Wrapping Up	?

10.	<u>Feature Toggles in Practice</u>	?
10.1	<u>Simple Feature Toggles</u>	?
10.2	<u>Maintainable Feature Toggles</u>	?
10.3	<u>Separating Static Assets</u>	?
10.4	<u>Preventing Accidental Disclosure</u>	?
10.5	<u>Runtime Toggles</u>	?
10.6	<u>Incompatible Dependencies</u>	?
10.7	<u>Testing of Feature Toggles</u>	?
10.8	<u>Removing Toggles for Completed Features</u>	?
10.9	<u>Wrapping Up</u>	?
11.	<u>Driving Innovation into Delivery</u>	?
11.1	<u>Value Stream or Value Trickle</u>	?
11.2	<u>A New Approach</u>	?
11.3	<u>Wrapping Up</u>	?

Part IV — Data Visualization

12.	<u>A Thousand Words</u>	?
12.1	<u>Smelling the Coffee</u>	?
12.2	<u>Visualization Design Principles</u>	?
12.3	<u>The Visualization Design Process</u>	?
12.4	<u>Visualization Design Patterns</u>	?
12.5	<u>Tools and Frameworks</u>	?
12.6	<u>Wrapping Up</u>	?
	<u>Bibliography</u>	?
	<u>Index</u>	?