

Extracted from:

The Nature of Software Development

Keep It Simple,
Make It Valuable,
Build It Piece by Piece

This PDF file contains pages extracted from *The Nature of Software Development*, 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 printed versions; the content is otherwise identical.

Copyright © 2015 The Pragmatic Programmers, LLC.

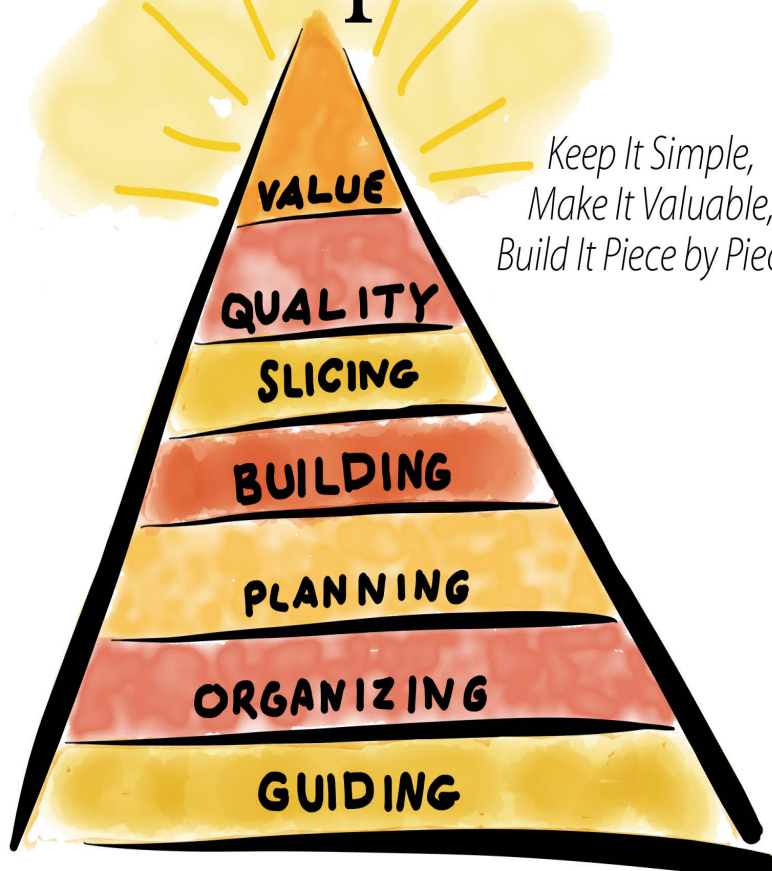
All rights reserved.

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 Nature of Software Development



Ron Jeffries

edited by Michael Swaine

The Nature of Software Development

Keep It Simple,
Make It Valuable,
Build It Piece by Piece

Ron Jeffries

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina



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 <https://pragprog.com>.

The team that produced this book includes:

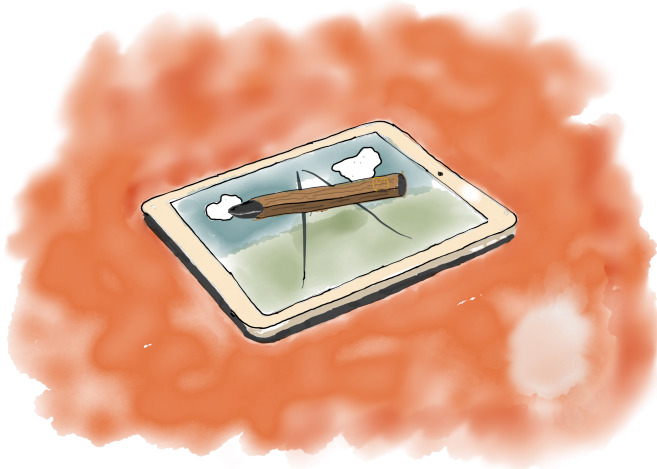
Michael Swaine (editor)
Potomac Indexing (indexer)
Liz Welch (copyeditor)
Dave Thomas (typesetter)
Janet Furlow (producer)
Ellie Callahan (support)

For international rights, please contact rights@pragprog.com.

Copyright © 2015 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: 978-1-941222-37-9
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—February, 2015



Preface

I've been doing software for over a half century. I've had some great successes and some truly colossal failures.

For all that time, I've been talking with people, coaching, and teaching about software development. And mostly, I've been thinking. I've been trying to figure out how this can all seem so simple and yet be so complex. If you've been involved in software development, you too have probably often felt that all this should be simple, but somehow it gets all complicated.

Thanks to being in the right place at the right time, I've been part of the Agile movement since the very beginning. That has drawn me back toward simplicity.

Like many of the best ideas in software development, modern "Agile" software development offers to make software development more productive and better controlled by making it simpler. Agile is simple. Four values, a dozen principles. How complex could it be? Well, it still seems to get pretty darn complex.

Agile methods like Scrum and XP are also simple. Again a few values, a couple of meetings, a handful of artifacts, how complex could they be? And still it gets so complicated so quickly.

What's up with that?

I have begun to see a way of looking at the whole process of software development. I'm starting to see a general overview that might help us keep things simple. Inside, there will still be plenty of complexity, but I hope this high-level map will help us pull back and find the simplicity when we find ourselves in the weeds.

Software development has many facets: determining value, managing value flow, organizing around the work, planning, building, and so on. Each of these facets needs to focus on producing value. Value needs to be visible so that it can be guided and managed. For this, we need to step back from the details and find the essential simplicity in this very complex activity.

When I think about things, I draw pictures that focus on some aspect of the topic. I try to think of a few words that will quickly focus my thinking when next I think about the topic. I use pictures to give me a different perspective. Since my drawings are perforce simple—I'm not very skilled—I use them to cut away complexity and look at what's left. I'm giving you a look at that thinking.

This book is an attempt at finding some essential simplicity inside the complex activity of building software products. I believe I have a handle on some good ideas. At best, this is a bit of a clearing along a tangled trail. Please take these thoughts and use them to find your own sense of simplicity amid all the chaos. Good luck!