

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.

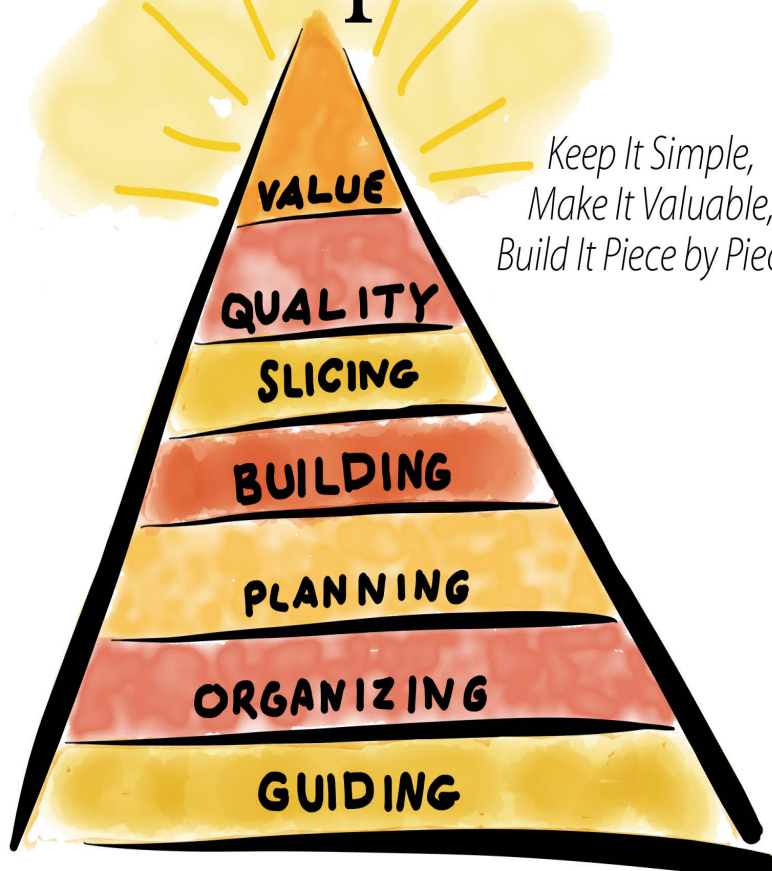
All rights reserved.

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](mailto: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

# Value—How Can We Measure It?

---

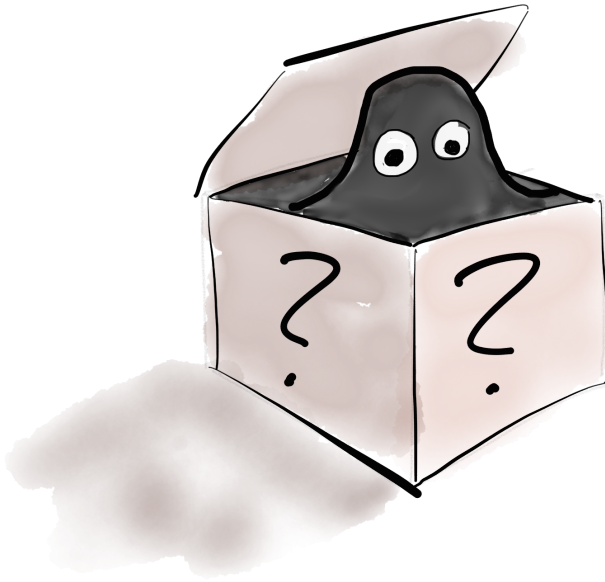
We began by suggesting that “Value is what you want” and advising you to produce small bits of real value every couple of weeks. We asked you to say “Show us the software” to see what’s really going on. The main idea is to concentrate on value, not cost, and to see that value in terms of real, running software with features we can understand.

*But Ron, you talk about value as if it’s purely subjective. Shouldn’t we be using real, solid numerical information to make our decisions? What do you have against measurement?*

Well, it’s a fair cop: I am pushing back against using numerical measures, in value and even in cost estimates. There are some reasons for this.

*We don’t really know the numbers.*

For almost any product of interest, we don’t know the numbers. We don’t know how many users will use our feature. We don’t know how many lives will be saved. We don’t know if people will rate our latest idea with three stars or five. We don’t know if people will buy our product based on this idea, or refuse to buy it.

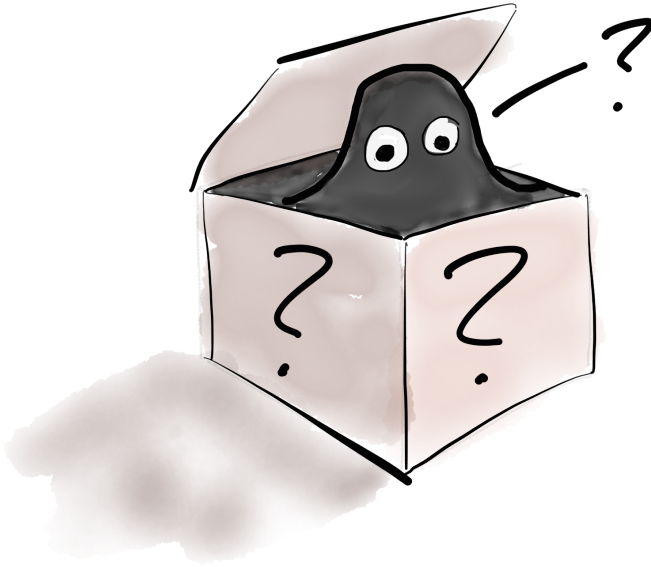


*Big differences are important; small ones aren't.*

When we look at all the feature options before us, some of them are incredibly important, and some are really dull and boring. That's the distinction that matters: what's incredibly important versus what's boring.

*Different kinds of value aren't comparable.*

A Product Champion values many different things. The values go up and down as time goes on. Sometimes we need customer information: will people find this idea useful? Sometimes we need development information: will this idea take a day to do, a week, or ten years? Sometimes we value pleasing some customer or prospect: if we can give them this feature quickly, they'll give us their money. *Then what should we do?*



*But Ron, then what is “value”? How should we measure it? How will we know we have it?*

Don’t panic! You already have a very good idea what value is. Take any two things you might do. Ask yourself which one to do next—which one is more valuable right now. It turns out you almost always know. If you don’t know, ask yourself whether either one of them is worth doing at all. Often, you’ll find they’re not.

Then ask yourself *why* you prefer what you prefer. Write down a few notes. Ask again: *why*. This will give you a sense of some of the dimensions of value to you. Ask your stakeholders. Whether they agree with you or not, ask why. Build up a grasp of what’s important, and why.

It’s always tempting to try to express value in numbers, and if you have a way to do that, go ahead and use it. Then look at two things your numbers suggest and ask yourself whether you agree. If you do, go ahead. If not, that’s interesting! Dig into that until you and your numbers agree. If they just won’t, I’d advise throwing out the numbers.

Looking for numbers puts us on a bit of a slippery slope. If the company's purpose with the product is to make money, then we could measure the product by how much money is made. But there's no useful way to know whether that number is a good measure of anything, since it conflates sales concerns, product concerns, and of course, customer concerns.

Even worse, most monetary measures are trailing indicators: we don't get information until it's too late. We have no real way of knowing whether the revenue number is really good or really bad. Money is a terrible indicator: too slow, unclear when we get it.

I wish I had an easy answer, like counting up function points or user preference clicks or the like. Use those if you find them valuable, but know that the true value of all the measures you can think of is to build up an understanding, in your Product Champion, your stakeholders, and your team, about what's really valuable.

Instead, sit with your developers and stakeholders. Consider the things you might do. Select a combination of next things to do that the group agrees is most valuable. The true value in doing this is the learning that will come from reaching a consensus.

Then build it, ship it promptly, and listen to your users. Repeat.