

The Rock Crusher

A Model for Flow-Based Backlog Management



By Steve Adolph, Shane Hastie, and Ryland Leyton



Published by International Institute of Business Analysis, Toronto, Ontario, Canada.

© 2023 Steve Adolph, Shane Hastie, and Ryland Leyton. All rights reserved.

Print Edition ISBN: 978-1-927584-34-7

eBook Edition ISBN: 978-1-927584-35-4

This document is provided for educational purposes. The authors and IIBA® do not warrant that it is suitable for any other purpose and make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information contained herein.

IIBA®, *BABOK® Guide*, BACCMTM, the IIBA logo, and the IIBA Publishing logo are registered trademarks owned by International Institute of Business Analysis. CBAP® is a registered certification mark owned by International Institute of Business Analysis.

SAFe® is a registered trademark owned by Scale Agile Inc. and is used with permission.

Scrum@Scale is a registered trademark owned by Scrum Inc. and is used with permission.

Nexus™ is a registered trademark owned by Scrum,org.

No challenge to the status or ownership of these or any other trademarked terms contained herein is intended by the authors or International Institute of Business Analysis.

Any inquiries regarding this publication, requests for usage rights for the material herein, or corrections should be emailed to info@iiba.org.

Dedication

Steve: To my greatest teacher, my daughter, Sophia.

Shane: To my wife, Nancy-your support and encouragement enable me to thrive.

Ryland: To my husband, Mike, for his support and understanding in this project and in my life.

In Memoriam

The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is so difficult as establishing the detailed technical requirements. ... No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.

~ Frederick P. Brooks Jr., *No Silver Bullet*

Dr. Brooks passed away while we were writing this book.¹ He was truly a pioneer of the computing profession and an inspiration for generations of software engineers. Among many other achievements, he was the first person to coin the term "computer architecture" and was the father of the 8-bit byte. The world and our industry are worse off for his passing.

1. Hastie, S. *Mythical Man Month Author and Father of the 8-Bit Byte, Fred Brooks, Dies at 91*. InfoQ, December 6, 2022. <https://www.infoq.com/news/2022/12/fred-brooks-obituary/>.

Preface

For many of us, the Agile Manifesto was a breath of fresh air, an affirmation of what we thought was the right way to create and deliver software.

Why? Because it described a collaborative approach to coping with the hardest part of building a software system: choosing what to build.

The late Fred Brooks, one of the grand elders of software engineering, said in the early 1980s that the hardest part of building a software system is “deciding precisely what to build.” That sentiment echoed in the third value of the Agile Manifesto, which calls for “customer collaboration over contract negotiation.” The agile principles go on to emphasize that businesspeople and developers must work together daily to harness change for the customers' competitive advantage.

The heart of most Agile methodologies is the backlog, a beautiful and elegant approach to coping with changing requirements. Owned by a product owner, the backlog was supposed to be the single source of work for a team. There would be no ambiguity about what to do next, because all work came from the backlog. Determining precisely what to build would be a conversation between individuals rather than throwing documents over a wall. The coordination delays, the ambiguities, the scrambling to meet arbitrary process-oriented milestones would disappear, replaced by continuous objective feedback and learning that created superior economic value.

As agile coaches and consultants, we get called into organizations that are seeking improvement, which means we don't hear from the teams that were able to realize the agile ideal. This may skew our data, but in our experience many teams, maybe even most teams, have not been able to realize the benefits of agile software development and business agility. One root cause, in our assessment, is a model of backlog management that does not address the needs of teams and enterprises that operate outside that agile ideal.

To cope with the organizational reality facing most teams, we need to rethink the backlog. The core of our argument is simple: the backlog is a key tool for enabling the agile enterprise, yet guidance on how to manage it in any real enterprise context is woefully inadequate. Many agile methodologists do not want to be seen as prescriptive, but people

need more than two or three paragraphs on how to manage such a critical component of driving great economic outcomes.

We believe this shortage of guidance has resulted in the challenges we have seen at most of our clients: a backlog that is merely a reservoir of precommitted work, and teams that are at best doing incremental development-some not even that, given the way that they carry over backlog items from iteration to iteration.

However, many clients do get it. We have seen clients who benefit from real customer collaboration, who can quickly test value hypotheses and learn what is really valuable, who have developed good practices for managing their backlog in the real world.

This book captures the practices we have seen in these successful agile teams-the teams that get it and enable their whole organization to benefit from the promises underlying agility and agile approaches.

Going Beyond Software Engineering

Wall three of us have extensive software engineering experience and so may have a software engineering bias, but the artifacts, roles, and practices we present here go well beyond software engineering and Agile software development methodologies. Agility is not just for software engineering: It is a well-known strategy for economic success. It could even be argued that the software community appropriated the term agility.

The concept of value streams becomes important to support our agility at this stage. As Rather and Shook observed, "wherever there is a product there is a value stream."¹ This is true whether we are writing software, maintaining infrastructure, designing cyber-physical systems, or just organizing volunteers for a community event. We can classify activities in the value stream as either determining precisely what to do or doing it. Agility in any context explicitly incorporates a fast learning cycle: Hypothesize what we should do, do it, learn, repeat. From what we have observed, most organizations have broken value streams, with the "determining precisely what to do" steps hidden or disconnected from the "doing it" part. This impedes or altogether blocks the most valuable aspect of agility: learning what we should really be building.

1. Rother, M., and Shook, J. *Learning to See: Value-Stream Mapping to Create Value and Eliminate Muda*, 20th anniversary edition. Version 1.5. Lean Enterprise Institute. 2018.

How Did The Rock Crusher Come to Be?

The Rock Crusher has been nearly a decade in the making since its humble beginnings in a New Jersey bar. Here is each author's version of the story.

Steve Adoph

Like how much of software engineering and agile development has emerged, the initial concept for The Rock Crusher emerged on the back of a beer mat. In the winter of 2012, I was working with a very cool marquis instrumentation client in New Jersey. The client had taken us to their favorite bar, and we began discussing agile concepts, and particularly how the backlog worked. I found myself sketching the backlog - yes on the back of a beer mat. At this point our client said “oh, I get it, I use to work in materials handling and it's kind of like a Rock Crusher, you guys should change the way you draw this, it would be so much easier to understand.” He then proceeded to sketch a caricature of what we today are calling the Rock Crusher. I wish I had kept that beer mat, it would have been cool to have had a shot of it in this book.

But after that, so much of the metaphor made sense to me and it really threw into sharp relief the problems I had observed with the way many organizations have implemented their backlogs. The traditional model of backlog management or what we are calling the stacked plates model simply begs to be a reservoir and disconnects the team from the organization. A reservoir breaks the value stream. There is no concept of flow, and no explicit means for removing less valuable contents. There is no concept of turbulence and managing that turbulence for economic gain. The process of how things actually got into that backlog is hidden behind an overworked or disinterested product owner.

After this I started sketching backlogs using The Rock Crusher model to explain the backlog to my clients. The key learning was turning the backlog upside down to highlight how work items flowed through the backlog. The funnel shape of the backlog immediately implied far more work enters the backlog than could drain out through the bottom which implied the existence of a waste gate.

What really kept me thinking about The Rock Crusher was a number of clients that I worked with who referred to their backlog management process as “rock crushing” or described patterns of backlog management that aligned with The Rock Crusher metaphor. The Rock Crusher was not a new idea, rather it is a model for capturing common success stories.

Long before The Rock Crusher, Shane Hastie had been a professional colleague who was introduced to me by mentor Philippe Kruchten. Shane is not only my colleague, but I regard him as my personal friend. He invited me to join a group of thought leaders collaborating

with IIBA to create a set of guidelines for applying the agile mindset to agile business analysis. This was the *Agile Extension to the BABOK Guide*® version 2.

This is where I met Ryland. My personal perspective on Ryland is that between the 3 of us, he is the most grounded. While Shane and I advocate and work with clients to improve their ways of working, Ryland makes his bread and butter doing real work as a practicing business analyst. One running joke we have is that I generally write like an academic (I do have a PhD and after that indoctrination it is really hard to write human). Ryland is wonderful in saying “Steve, ...no.” and translating my academic arrogance into something that is approachable by people who are just trying to get a job done.

After completing the *Agile Extension to the BABOK Guide*® version 2., the three of us began informally collaborating on The Rock Crusher concept, often incorporating the ideas into our teaching and consulting. For example Shane published *#noprojects* with Eric Leybourne which included an early representation of The Rock Crusher.

Things really took off when IIBA invited me to contribute to *IIBA’s Knowledge Hub*. This sort of strapped on the boosters and inspired us to keep on going and write the book. The pandemic hit at the same time and there is only so much Netflix binging you can do while social isolating. So two years later, after a lot of work, revisions, more revisions, and even more revisions I hope we have captured all those ideas patterns in a manner you find helpful.

Shane Hastie

As Steve mentions, we were introduced by Philippe Kruchten sometime around 2008. Philippe was a fairly frequent visitor to SoftEd in New Zealand, speaking at our conferences and presenting his master classes. One of his ideas that strongly resonated with me was explicitly calling out different types of work in the product backlog.¹ He also mentioned this Canadian chap Steve who was doing some interesting things with backlogs and product agility. We invited Steve to the SDC conference in Wellington and Sydney in 2010 and this became the foundation for a firm friendship and lots of collaboration over the years.

At that time, there was a bit of a schism between the agile development community and the business analysis community. In 2008 I listened to a talk where the role of the analyst in agile development was described as being "the copy boy" for the development team. This inspired me to write an article sharing my perspective on the value of having an analyst on

1. Hastie, S. *What Color Is Your Backlog?* InfoQ. May 2, 2010. www.infoq.com/news/2010/05/what-color-backlog/.

the team.¹ This was my first experience with publishing something, and led to me taking on a part-time role as news editor for InfoQ.com.

At that stage I was heavily involved as a volunteer with IIBA in New Zealand and Australia. I got involved with the initial work writing the first version of the *Agile Extension to the BABOK® Guide* and advocated for it to become a joint effort of the Agile Alliance and IIBA. Version 1 was a reasonable starting point, but we knew that the ideas would evolve as they were applied in the wild, and lots of new learning would emerge.

When it was time for version 2, Steve was one of the thought leaders who stepped up to share his ideas. Ryland joined the team, along with James King, Kent McDonald, Stephanie Vineyard, Jas Phul, and Paul Stapleton. The discussions were enlightening, and it was a wonderful collaboration activity. Friendships were forged and deepened, and we were keen to find ways to continue the collaboration.

Steve and I had been talking about writing a book for years and had thrown lots of ideas into a virtual pot to see what came out as the thinking simmered. Steve was definitely the driver, taking the often unformed thoughts and giving them structure and academic rigor. As we continued putting them together, we realized that we needed the perspective of an active practitioner to turn our somewhat abstract ideas into practical advice. Both of us had been impressed by and enjoyed working with Ryland on the Agile Extension, so it was logical to invite him along on the ride. Fortunately, he was keen to be involved and joined us on the rollercoaster.

A note on crushing rocks: Early in my career as a systems analyst working on data processing products (sometime in the late 1980s) I worked on automating processes for mining companies. One of my strongest memories is of visiting a diamond mine in the Kimberlite fields in South Africa. There they blast large chunks of rock out of ancient volcanic pipes and put the rocks through a series of crushers, discarding the waste and filtering out the gems—in this case, real diamonds. It was fascinating to see how much work goes into extracting the gems and how much waste it creates. Even then I could see the analogy to identifying the real requirements for the products we were building—there were so many good ideas that weren't worth implementing when we examined them. When Steve first mentioned the rock crusher metaphor, it immediately reminded me of visiting the mine and of how this lines up with my experience over multiple decades.

If you take nothing else away from this book, do please implement a waste gate on your backlog and use it to filter out the real gems that constitute customer and business value.

1. Hastie, S. *The Role of the Analyst in Agile Projects*. InfoQ. December 5, 2008. www.infoq.com/articles/agile-business-analyst-role/.

Ryland Layton

I had the pleasure of meeting Steve and Shane during work on IIBA's *Agile Extension to the BABOK® Guide* version 2. At that point in my career, I was surprised to be in the room with them and the rest of the team!

To summarize working with Steve and Shane in just a sentence or two, Steve was a towering academic who usually could cite several sources relevant to the point he was making. Shane, meanwhile, was a cross between Santa, because of his beard, and Yoda, because of his deep knowledge and insights which are often framed as a question.

As for myself, I agree I tend to take the practitioner view of things: How will the average "analyst in the street" use this? Is this information accessible? Can it be applied easily in their work?

Between that view, and the heavy thinking often expressed during *Agile Extension* working sessions, I had a funny moment relating to a particular contribution to our work together. People could take a long time, sometimes three to five minutes, explaining that they loved a particular thought and idea. These ideas were in fact just great-truly valuable!-but perhaps a little far afield from the topics for the *Agile Extension*. After this happened several times, I proposed that we replace those conversations with the phrase "that sounds like a great topic for your next book." This would stand in for the lengthy (and respectful) conversation, and we could still talk about the subject if we needed to.

We all found ourselves on the listening end of that phrase.

In short, it became a good working relationship, and when we were done with the *Agile Extension* I certainly missed their company.

In 2021, Steve and Shane included me in their early review group to look at the rough manuscript of the book you are now reading. I liked it. I really liked it, and I feel the ideas presented are very valuable to the analyst toolkit and to those who are seeking better practice and want to create positive organizational change.

A few days after I read it, I walked around my backyard on my phone while Steve and I threw ideas back and forth about the book, the directions it could go, and what it needed to get there. I mentioned I thought the writing was excellent but, as he notes, needed some grounding in the core of the matter for the reader, and to stay away from distracting topics. I offered to help with this if they were considering another author. Steve immediately told me he and Shane had been trying to figure out how to ask me so, yes, they were up for it.

I'm happy to have made contributions to the book and added some ideas, and I think I did add clarity and brevity in parts. However, I must emphasize that the work came from Steve and Shane. I am very happy to have been included in what they started and shaped.

That said, if you read this whole book and still feel something is missing, such as another 200 pages on this topic, contact Steve or check out the website at RockCrusher.org.

Finally, a word on what I find so appealing about this book. I love how the Rock Crusher helps show the actual work that is done in product design and ideation. So much of this is hidden in actual practice that it becomes frustrating for analysts, product professionals, and developers. We hear things like "You just have to get the requirements! How hard is it !?!?" from people who are new to the field, or who don't understand that-as Steve points out-deciding what to build is really, really, really important. It is also no small effort to make those choices.

I also was a fan of the Rock Crusher manuscript for showing how to set this system up, explaining how to make the work and the flow visible in an organization, and describing the value this would bring. So many people try to create real change and have a hard time showing the way, creating a vision, and walking the path. I think this book can help by giving you a good, solid approach to such change.

I hope you will find this as interesting and valuable as I do. My best wishes to you as you apply the principles and techniques in your workplace.

Acknowledgments and Appreciations

We are not the first to say that “It Takes A Village”, and we’d like to appreciate the following people and organizations for their help.

First, the International Institute of Business Analysis. We are very proud to be the first formal book from the IIBA Publications. We did not choose this lightly - there were other options should we have preferred them. We chose IIBA because we felt that the mission, vision, and values of the organization were strongly aligned with our own, and that this would result in greatest service to the analyst and product communities.

Second, the work of IIBA team members Paul Stapleton, Tiffani Iacolino, Yana Kelly, and Scott Lidstone, in the creation of the book itself. The success of the end product would simply not have been possible without your strong, mindful, and collaborative work.

We wish to thank our professional editor, DeAnna Burghart, not only for being a pleasure to work with but also for her professional skill, her knowledge of the domains of business analysis and agility, and her patience in working with the three of us. DeAnna truly is a master of the art of cat herding.

Being a professional reviewer of a book is a commitment of time and thinking. We'd like to appreciate our reviewers for their contribution: Kevin Brennan, Barb Carkenord, Rita Emmons, James King, John Kosco, Evan Leybourn, Kent McDonald, Johanna Rothman, and Craig Smith. You are all strong members of the field and your opinions and feedback have been valuable to each of us as authors and professionals. You have our sincerest thanks for your help

Table of Contents

Preface.....	i
Going Beyond Software Engineering	ii
How Did The Rock Crusher Come to Be?	iii
Acknowledgments and Appreciations	viii
Chapter 1: Introducing The Rock Crusher	1
Stacked Plates and Broken Value Streams.....	2
The Product Owner: Only Superhumans Need Apply	9
The Rock Crusher: A Grown-Up Model of Backlog Management	10
Try This.....	15
Chapter 2: Exploring The Rock Crusher.....	17
The Rock Crusher’s Components	18
The Waste Gate.....	22
Solutions and Solution Increments	25
The Village—Rock Crusher Roles.....	25
Implementing the Rock Crusher with Kanban	30
Handling Rocks on and off the Roadmap	30
Readiness Horizons.....	32
Rock Crusher Metrics	33
Try This.....	34
Chapter 3: Rock Crusher Roles and Responsibilities	35
Accountable Roles.....	37
Responsible Roles	40
Try This.....	44

Chapter 4: The Challenges of the Ownership Roles	45
The Challenge of Misaligned Ownership.....	46
Ownership Collaboration Models	47
Good Practice Ideas	52
SME and Backlog Owner Collaborations.....	53
Collaboration Is Key.....	54
Try This.....	55
Chapter 5: Correctly Defining and Managing Rocks.....	57
Well-Formed Rocks.....	59
Identifying Well-Formed Rocks	60
Right-Sizing Rocks	61
Progressive Refinement.....	62
Rock Quality	64
Discarding Rocks	65
Verifiable Models, Solution Increments, and Solutions.....	65
Try This.....	67
Chapter 6: Use Crushers to Make Analysis Visible.....	69
Crushers—Rocks for Backlog Refinement	71
Highlighting the Value of Modeling	74
The Economics of Accuracy and Precision	75
Managing the Flow	79
Avoiding Stealth BUFD.....	80
Try This.....	81
Chapter 7: Crushing Rocks the Right Way.....	83
Value Breakdown versus Work Breakdown	84
Rock Refinement Techniques	85
Try This.....	93
Chapter 8: Backlog Refinement with the Rock Crusher	95
Avoiding Planning Panic.....	97
The Backlog Refinement Meeting	97
Answering the Four Questions.....	102
Strategic and Tactical Refinement Meetings.....	105
Try This.....	107

Chapter 9: Handling Rocks on and off the Roadmap	109
Handling Non-Roadmapped Rocks	111
Managing Incoming Rocks	111
Try This.....	117
Chapter 10: Rock Entry through Front and Back Doors.....	119
Intake	120
Rolling Wave Planning with Rock Crusher Horizons	124
Try This.....	126
Chapter 11: Using The Rock Crusher to Facilitate Agile Business Analysis	127
Agile Business Analysis	129
Principles of Agile Business Analysis	133
Try This.....	136
Chapter 12: Doing Business Analysis with the Rock Crusher	137
Analysis Planning	139
Creating Crushers	141
Just Enough Analysis Planning	145
Try This.....	148
Chapter 13: Is It Working? Rock Crusher Metrics	149
Metrics Categories	150
Connecting OKRs and Metrics	152
Performance Metrics.....	153
Hygiene Metrics	157
Try This.....	159
Chapter 14: The Rock Crusher at Scale.....	161
Scaling Practice 0: First, Do You REALLY Need to Scale?	162
Scaling Practice 1: Enlist the Village	163
Scaling Practice 2: Delegated Backlog Ownership Accountability	165
Scaling Practice 3: Using the Strategic Refinement Meeting to Coordinate and Maintain Alignment	167
Scaling Practice 4: Multi-Horizon Planning.....	169
Work Intake at Scale	172
Try This.....	176

Chapter 15: Implementing Your Rock Crusher.....	177
Form a Rock Crusher Hypothesis	178
Choose a Value Stream.....	179
Identify Your Village.....	182
Visualize Your Rock Crusher.....	183
Establish Your Intake Policies.....	188
Establish Your Waste Gate Policy.....	190
Schedule the Ceremonies	190
Continuously Improve Your Rock Crusher	191
Final Advice.....	191
Try This.....	194
Glossary	195
Index.....	201
About the Authors	207

1. Introducing The Rock Crusher

Learning Objectives

After reading this chapter, you should be able to explain:

- the value and benefit of the backlog,
- the economic consequences of a broken value stream,
- how our current model of backlog management can break the value stream by impeding flow and disconnecting the team, and
- at a high level, the benefits of the Rock Crusher model for backlog management.

The backlog is a beautiful, powerful, and delightfully simple tool for managing work in modern agile organizations. It is the single repository from which a team pulls its next most valuable work item. The backlog fosters agility by enabling the additions, removal, reprioritization, and, most importantly, visualization of all the potential work for a product.

The product owner facilitates this simplicity by representing a single clear line of content authority. According to the Scrum Guide, the product owner is “accountable for effective Product Backlog management.”¹ Simply put, the product owner is accountable for ensuring that the development team works on the right things. This arrangement helps businesses avoid the horrendous waste that's created when developers are forced to spend time and effort coordinating competing work requests rather than doing the work.

This change in how we conceptualized and authorized work-combined with values and principles expressed in the Agile Manifesto-was a true paradigm shift in how teams got the job done.

1. Schwaber, K., and Sutherland, J. *The Scrum Guide*. 2020.