



The Business Value of Developer Relations

How and Why Technical Communities Are Key To Your Success

—

With a Foreword by Jono Bacon

—

Mary Thengvall

Apress®

The Business Value of Developer Relations

**How and Why Technical
Communities Are Key To
Your Success**

Mary Thengvall

With a Foreword by Jono Bacon

Apress®

The Business Value of Developer Relations

Mary Thengvall
San Francisco, California, USA

ISBN-13 (pbk): 978-1-4842-3747-2
<https://doi.org/10.1007/978-1-4842-3748-9>

ISBN-13 (electronic): 978-1-4842-3748-9

Library of Congress Control Number: 2018958879

Copyright © 2018 by Mary Thengvall

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Louise Corrigan
Development Editor: James Markham
Coordinating Editor: Nancy Chen

Cover image designed by Erick Zelaya

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at www.apress.com/bulk-sales.

Any other supplementary material referenced by the author in this book is available to readers at <https://www.marythengvall.com/devrelbook>.

Printed on acid-free paper

*To all the community builders and Developer Relations professionals
who have been by my side for the last decade as we've built this
industry from the ground up . . . this is for you*

Table of Contents

About the Author	xi
About the Technical Reviewer	xiii
Acknowledgments	xv
Foreword	xvii
Introduction	xxi
Part I: What Is the Value of a Technical Community?	1
Chapter 1: An Introduction to Community	3
Establish the (Flexible) Boundaries	4
Why Do You Want a Community?	6
What Do You Hope to Accomplish with a Community?	7
Who Makes Up Your Community?	9
Which Segment of The Community Do You Want to Focus on First?	11
So, Do You Need a Developer Relations Team?	12
Be Willing to Make Changes	14
Chapter 2: Selling Community to Your Company	17
Gathering of the Stakeholders	18
What Are You Trying to Accomplish?	21
Your Version May Vary	22
Setting Expectations	24
Trouble in Paradise	26
All Aboard	27

TABLE OF CONTENTS

- Chapter 3: Keeping a Pulse on the Community 29**
 - The Role of Social Media in Community Building 32
 - Establishing Voice 33
 - Choosing Platforms 34
 - Working Together 35
 - Technically Speaking 36
 - Focus on Developers..... 37
 - Elevate and Amplify 40
 - Personal Brand vs. Company Brand..... 41

- Chapter 4: Measuring Your Success..... 43**
 - Storytelling Is Part of the Job Description 46
 - Checks and Balances..... 47
 - Trust Your Gut, But 48
 - It's Not All About the Data..... 48
 - ROI Is a Difficult Metric for DevRel 49
 - Qualitative + Quantitative = Sweet Success 52
 - Online Engagement 52
 - Multichannel Endeavors 53
 - In-Person Connections 54
 - Anecdotes vs. Facts 56
 - Time-to-value..... 59
 - Quick Wins and Weekly Reports..... 60
 - Finding Your Specific Metrics: Libby Boxes..... 61
 - Pirate Metrics 62
 - Setting Objectives..... 63
 - Building a World-Class Developer Relations Team 68

Chapter 5: Building a Developer Relations Team.....	73
Draw Your Mall Map.....	73
What's in a Name?.....	74
Team Title: Developer Relations.....	76
Strategic Lead: Developer Relations Manager.....	76
Technical Contributor: Developer Advocate.....	77
Cruise Director: Technical Community Builder.....	80
Developer Experience Manager.....	80
Technical Ambassador.....	81
Technical Engagement Manager.....	82
DevRel Project Manager.....	83
Full-Time Engineer.....	83
Who's First?.....	84
Know Your Strengths and Weaknesses.....	84
Make the Investment.....	86
The All-Important Question of Where.....	89
Marketing.....	90
Engineering.....	90
Customer Success.....	92
Product.....	93
Training or Content Division.....	94
Internal Consulting Firm.....	94
Community Department.....	95
Geographic Location.....	95
Where Does That Leave Us?.....	96
Set Them Up for Success.....	99

TABLE OF CONTENTS

- Part II: Building and Engaging with the Community 101**
- Chapter 6: Finding Your Community 103**
 - Relationships 101 104
 - Don't Start from Scratch 107
 - Open Source Communities..... 110
 - Read vs. Write Communities..... 110
 - The “Stickiness” of Open Source Communities 111
 - The Person in Charge of the Community 113
 - The People Around Them 114
 - Feeling Like They Belong 115
 - Open Source Is an Investment 115
 - Giving Back to the Community 117
- Chapter 7: Building a Healthy Community 121**
 - Finding Your North Star: Community Mission and Vision 122
 - Setting Expectations: Code of Conduct 125
 - Keeping Topics on Track: Community Guidelines 126
 - Providing Stability: Community Rituals 127
 - A Gathering Place..... 127
 - Go with the Flow..... 130
 - Community Advocates 130
 - Diversify..... 135
 - Build a Partnership 136
 - Communicate 138
 - Prepare 139
 - Scale..... 140
 - The Question of Compensation..... 143
 - A Stitch in Time Saves Nine 146

Chapter 8: In-Person Events: The How, Why, and Where	147
What Are You Trying to Accomplish?.....	148
Brand Awareness.....	148
Feedback.....	149
Engage with the Community.....	152
Test the Waters	153
Maximize time, Minimize travel	153
Conferences, Conferences Everywhere	156
Navigating the Maze of Meetups	159
The Exception to the Rule.....	161
Speaking Engagements	165
Chapter 9: Dealing with Common Community Issues.....	169
Burnout	169
How to Prevent Burnout in Your Team	171
How to Recover from Burnout.....	178
Imposter Syndrome.....	182
You Don't Have to Be Good at Everything	183
The Generation Gap.....	184
How to Fight Off Imposter Syndrome	185
Battling Distrust	188
Chapter 10: Building Your Personal Brand	193
Views Are (Not Only) My Own.....	193
Amplify the Good.....	194
Foster Your Own Community.....	196
Be Authentic.....	198
Let Your Personality Shine.....	198
Don't Overdo It.....	200
Practice Empathy.....	201
Use Your Platform.....	202

TABLE OF CONTENTS

- Appendix A: Trip Reports 203**
 - Sample Trip Report..... 204
- Appendix B: Event Scorecard 207**
- Appendix C: Hackathon One-Page Handout 209**
- Appendix D: Developer Resource Card 211**
 - Chef Developer Resource card..... 212
 - SparkPost Developer Resource card..... 213
- Appendix E: Sample Event Process and Playbook..... 215**
 - Event Process Template 215
 - Event Goals..... 216
 - Event Selection..... 217
 - Event Preparation 221
 - Event Execution 224
 - Large Event Playbook (Hackathons and Conferences) 225
 - Meetup Playbook..... 230
 - Event Follow-up 231
- Index..... 233**

About the Author



Mary Thengvall is a connector of people at heart, both personally and professionally. She loves digging into the strategy of how to build and foster developer communities and has been doing so for more than ten years. After building community programs at O’Reilly Media, Chef Software, and SparkPost, she’s now consulting for companies looking to build out a Developer Relations strategy. In addition to her work, she’s known for being “the one with the dog,” thanks to her ever-present medical alert service dog Ember.

Mary is founder and cohost of *Community Pulse*, a podcast for community managers and developer evangelists who are looking for information on community building. She curates *DevRel Weekly*, a weekly newsletter that provides a curated list of articles, job postings, and events every Thursday. She’s also a founding member and “Benevolent Queen” of the DevRel Collective Slack team.

She is also a member of Prompt, a nonprofit that encourages people to openly talk about mental illness in tech. She speaks at various conferences and events about building and fostering technical communities, as well as how technology professionals and teams can prevent burnout.

About the Technical Reviewer



Jono Bacon is a leading community manager, speaker, author, and podcaster. He is the founder of Jono Bacon Consulting, which provides community strategy/execution, developer workflow, and other services. He also previously served as director of community at GitHub, Canonical, XPRIZE, and OpenAdvantage and has consulted and advised a range of organizations.

Bacon is a prominent author and speaker on community management and best practice, and wrote the best-selling *The Art of Community* (O'Reilly Media, 2012). He is the founder of the Community Leadership Summit, the primary annual conference for community managers and leaders, as well as the Open Collaboration Conference. Bacon is a regular keynote speaker at events on community management, organizational leadership, and best practice.

Bacon has provided community management consultancy, for both internal and external communities, for a range of organizations. These include Deutsche Bank, Huawei, GitLab, Intel, SAP, HackerOne, data.world, Sony Mobile, Samsung, Open Compute Project, IBM, Dyson, Mozilla, FINOS Foundation, Executive Centre, AlienVault, and others. He holds advisory positions at AlienVault, Moltin, data, world, Open Networking Foundation, and Open Cloud Consortium.

In addition to writing *The Art of Community*, Bacon is a columnist for *Forbes* and *opensource.com*, author of *Dealing with Disrespect*, and coauthored *Linux Desktop Hacks* (O'Reilly Media, 2005), and *The Official Ubuntu Book*, 2nd Edition (Prentice Hall, 2007). Bacon has published more than 500 articles in 12 different publications. He writes regularly for a range of magazines.

Bacon was cofounder of the popular *LugRadio* podcast, which ran for four years with more than 2 million downloads and 15,000 listeners, and which spawned five live events in both the UK and the United States. He also cofounded the *Shot Of Jaq* podcast and the *Bad Voltage* podcast, a popular show about technology, open source, politics, and more.

He lives in the San Francisco Bay Area with his wife, Erica, and their son, Jack.

Acknowledgments

This book was not a solitary activity. Like most things in my life, it took a community. I can't possibly thank them all, but I'd like to shine a spotlight on a few particular people.

First and foremost, my partner Jeremy Price—thank you for giving me your blessing to write this book (perhaps against your better judgement ;-)) and for standing by me through it all, even when the amount of work increased dramatically. Your support and encouragement have meant the world to me, and there's no one I'd rather share this journey with.

To my parents, thanks for your belief in my writing ability. Your encouragement over the years, from my elementary school days when I was publishing a neighborhood newspaper to your willingness to take a first pass at this book even though it's out of your element, has pushed me to where I am today.

To my sister Sarah, thank you for your pride in my work despite not understanding what it is exactly that I do ;-), and to my brother-in-law Dan, for being my virtual writing partner as you worked on your dissertation. Thanks to both of you for creating the three most amazing nieces and nephew in the whole world. Emily, Ethan, and Hannah: I love you to the very edge of the universe and back.

To my “brother” Tim—thank you for being a constant in my life, no matter how busy I am or how many times I miss your phone calls. I know you'll always be there for me, which means more than I can say.

To Aydrian, Avi, and Danielle: the original (and dare I say the best) Developer Avocados, and to Daeyon Griffin, for the slip of the tongue that has led to my professional brand. We learned a lot of lessons along the way (many of which wound up in this book!) and, more importantly, have formed friendships that I wouldn't trade for the world. You'll always be my favorite Developer Avocado team. Rest assured that I'm wearing my avocado socks for good luck as I'm writing these words.

To Louise, my amazing editor, who put forth the idea of this book and has supported and encouraged me along the way, as well as kicking my butt into gear when I needed it. You have talked me down from more than one cliff through this process, and I'll always be grateful for the friendship that we've built along the way!

ACKNOWLEDGMENTS

To Julie Gunderson, Jason Yee, and Nathen Harvey, who volunteered me to be the fourth and final author. You may not be with me at the finish line, but you successfully passed the baton to me to finish the marathon, and that's no easy task. I wouldn't be here without you, and this book wouldn't be the same without your insights and experiences.

To Laura Baldwin, president of O'Reilly Media, who gave a fresh-faced 20-something the chance of a lifetime. You jump-started my journey and opened me up to opportunities I couldn't have even dreamed of when I first asked to have conversations *with* our audience instead of just talking to them. To you and all the rest of my O'Reilly family: Sara and Sara, Courtney, Brady, Marsee, Jon, Betsy, Keith, Simon, Tony, and so many others . . . thank you for the eight years we shared!

To Carol Adams—the teacher of a lifetime. You nurtured my love for learning in ways that few others have been able to do. I still have your note from my third-grade yearbook, and I'm finally making good on sending you an autographed copy of my first book. Keep an eye on your mailbox!

Jodi, Pete, Matt, and Caryn—the four of you are some of the best friends I could ever ask for. Your encouragement, laughter, prayers, and reminders that this too shall pass got me through these last few months with (most of) my sanity in check.

And Paul—for not only listening to my explanation of why Developer Relations is like avocados with bemusement, but coming up with a brilliant name for Persea Consulting as a result. To say that your patience as I've finished this book while simultaneously planning REdeploy has been greatly appreciated is a huge understatement.

Last but certainly not least, thanks be to God, who gave me the strength to finish this book and the talent and opportunity to do it in the first place. Thank you for the continual reminder that no matter how others see me or what religious edicts are in place, I am, and always will be, yours.

Foreword

I am deeply passionate about communities. I have been ever since my earlier involvement in open source and when I started experimenting with building my first few communities back when I lived in the UK.

This early interest really culminated when I strapped on my goggles and nose-dived into a new project called Ubuntu, tasked with building a global movement around an open source operating system. Eight years later I came up for air, and we had built a community comprised of hundreds of developers around the world, millions of users, local Ubuntu groups littered across the globe, and people actively shaping Ubuntu every day by producing documentation, translations, marketing initiatives, testing, and more.

This early experience in my career showed me the true potential for harnessing groups of engineering talent and availability, particularly when wrapped around a clearly articulated mission.

I didn't fully appreciate this, though, until I worked at XPRIZE, an organization that coordinates and leads major incentive competitions. My first XPRIZE, the Global Learning XPRIZE, challenged teams to build an Android app that taught kids how to read, write, and perform arithmetic within 18 months without the aid of a teacher. It was designed to bring education to the more than 250 million kids who don't currently get it around the world. The prize purse was a whopping \$15 million, underwritten in part by everyone's favorite science-reality entrepreneur, Elon Musk.

Working at XPRIZE was at times bizarre. It was an environment where over lunch we would talk about how we could solve water sanitization issues, produce vehicles to improve goods and services in rural third-world areas, use 3D printing to print houses for low-cost widespread housing initiatives, and more. This is what I loved about XPRIZE: genuine "moonshot" thinking underlined by the question "How do we incentivize people enough to make this reality?"

Unsurprisingly, the Global Learning XPRIZE attracted a wide range of people, from educators to startup founders, artists, translators, and more.

Many developers joined, but due to the competitive nature of the prize, their ideas and talent were . . . well, rather private. Competitors formed into their own teams and

FOREWORD

attacked the challenge with steel smiles and open minds, but this innovation occurred largely in a private setting, with the exception of some shared foundational projects.

As such, one element that was so compelling and familiar to me about the open source developer experience was curiously missing: problem-solving and collaboration around tractable solutions, out in the open.

This was entirely to be expected and no fault of XPRIZE. The organization was not an open source project, and the whole point of the incentive prize model is that people actually compete with each other. However, as Eleanor Roosevelt once mused, “Absence makes the heart grow stronger,” and I started to miss the culture of collaborative engineering communities, with the broader open source community being one such prime example.

This realization was not merely a cultural one. It got me thinking about the psychology and driving forces behind developers. What really makes great engineers tick? Throughout my career at that point, there were clear differences in engineering cultures—such as the difference between a “GitHub generation” startup and an enterprise Windows shop—but there were a remarkable amount of consistency and common themes too.

This is something that would illustrate itself in droves over the following years. As a consultant, I build community strategy and execution for a wide range of industries, including technology, financial services, consumer products, entertainment, security, professional services, and others. A significant chunk of my work is focused on building developer communities and ecosystems where developers either build on top of a platform or contribute to a core platform itself.

Across these widely varying clients, it has been fascinating to see these consistent themes emerge among these different types of developers, even when the scope, focus, culture, and norms of these different organizations vary so significantly. Of these commonalities, though, I see one of the most significant as being a desire for *mastery*.

With any discipline—carpentry, mechanical engineering, chemical engineering, biomedicine, robotics, and others—there is a journey from *learning* to *optimization* and then *mastery*.

A fundamental principle in converting learning to mastery is not just that we can furnish ourselves with the education to deliver good work, but—importantly—that we can construct hypotheses for how to execute this work well, and then something or someone validates or rejects our hypothesis. In other words, we need to be able to experiment and see whether our experiments work out.

For many of these other disciplines, this experimentation is limited. If you are learning bioengineering, you probably lack the equipment, certifications, and budget to do it in your home office. If you are learning chemical engineering, I doubt you have a refinery in your backyard. For many software developers, though, these limitations largely don't apply: a computer, an Internet connection, a curious mind, and a giant bucket of coffee will get you most of the way there.

There is another critical component in this developer soup du jour, though. Unlike many other institutions, code can be shared, improved, refined, and iterated at little to no cost. A global patchwork of GitHub and GitLab repositories don't just contain software—they contain our shared understanding and collaboration around common interests and problem solving.

With this common tooling, readily available (and normative) peer review, and a self-starter culture of learning and collaboration, this has jump-started millions of initial curiosities about programming into well-compensated, rewarding, in-demand careers.

The impact of all this has been profound. We have seen entire industries “disrupted” and reconstructed by talented developers with the aforementioned, not-entirely-insignificant bucket of coffee and a bundle of self-motivation. We have seen developers able to apply their talents to not just reshaping the world we live in, but to how we effect change at both a professional and social level. It is all really quite remarkable.

Tapping into this machine, though, can be a challenge for many, and this is where community engagement and Developer Relations play such critical roles. This isn't as simple as marketing and broadcasting targeted messages to developers. Great Developer Relations requires someone to not just understand the needs and goals of a developer, but to fundamentally understand the fabric of their culture—where it differs and where it is the same. Doing this well can result in remarkable results; doing it poorly can manifest in developers awkwardly shoeing away such Developer Relations like an unwanted fly at a picnic.

Before you delve into Mary's book, I will leave you with one piece of advice. Developer Relations, and more broadly how we engage and build culture in communities, is a remarkably nuanced, complex, and context-specific discipline. There simply isn't a one-shot recipe that works well for everyone. You can benefit from others, such as the insight from Mary in this book, but consider this guidance as a starting point to get you going and then evolve your own approach.

I tell this to my clients all the time: every community is different, and although many methods and techniques work well and reliably, the most critical skill of all to learn is

FOREWORD

observing what is happening in your community and being able to react and optimize it. Just like with software engineering: learn, optimize—and then true mastery of your own can be accomplished.

This is an exciting journey. Good luck!

Jono Bacon
Jono Bacon Consulting
www.jonobacon.com | jono@jonobacon.com
May 2018

Introduction

This book is for anyone who's trying to figure out what a technical community is and why it's essential for the success of certain companies. It's for Developer Relations and Community Manager professionals, as well as directors, VPs, and C-suite folks who are trying to manage those teams within an ever-changing landscape of company goals and priorities. It's for those who are discouraged by the misconceptions of community building as well as those who are encouraged by the influx of companies trying to hire someone who understands the developer audience. It's for those who have yet to discover the true value of Developer Relations, as well as those who know that it has inherent value but struggle to put it into words.

But first, what is Developer Relations? At its foundation, the purpose of Developer Relations (or DevRel) is to *build relationships with the developer community*. DevRel professionals act as a liaison between their company and the developer audience, who are typically the end users of the product. Whereas most professionals have the best interests of the business at their front of their minds driving their day-to-day decisions, DevRel professionals have the best interests of the community as their driving factor. They, of course, care about the success of the business as well—it is, after all, what pays their bills—but they understand that if the community is happy and successful as a result of using the product, the business is far more likely to succeed as well.

In order for the DevRel team to succeed, however, they must be fully supported by the company. From having a clear set of business goals and expectations to having the right tools for the job, they need to know that their work is seen as valuable and is therefore not only allowed but actively encouraged by the stakeholders in their company.¹

Many companies are realizing that it's not enough to simply have a Developer Relations team. The stakeholders must understand the true value such a team provides, and the team must be set up for success.

¹You can read more on the items that every company needs to have a successful DevRel team in Anil Dash's post "A Developer Relations Bill of Rights": <https://medium.com/glitch/a-developer-relations-bill-of-rights-21381920e273>

INTRODUCTION

As such, this book is divided into two distinct sections. These sections, while applicable for both groups I've mentioned, are geared toward different audiences. Part I (Chapters 1–5) is directed toward the decision makers in the company who are trying to understand the value of a technical community. Their questions likely revolve around whether or not they need to foster their community, what the appropriate metrics are, what the difference is between Developer Relations and community building, and how to create a successful team.

In Chapter 1, I address the key questions you want to ask before investing in a technical community: why do you want one, what do you hope to accomplish, and who actually makes up your community?

Chapters 2–4 speak to creating a business case for building a technical community. How do you convince the business owners and decision makers that connecting with the developer community is an important investment, and how do you agree on metrics to ensure alignment and success?

Once we've discussed how to build this business case for including Developer Relations in your success strategy, it's time to figure out who's going to help fill that role. Chapter 5 walks through who your first hire should be, what the differences are between a community manager, Developer Advocate, and technical evangelist, and where your newly founded Developer Relations team should be placed within your organization.

Part II (Chapters 6–10) is geared toward the Developer Relations practitioners—those who are involved in community building on a daily basis. Their questions focus more on the day-to-day tactics, including finding the right audience, walking the tightrope between representing the company and building a personal brand (which also benefits the company), and doing in-person events.

Chapters 6–8 dive into what you can accomplish now that you actually have a team in place, from finding your particular segment of developers to interacting with them, both online and in person.

Lastly, I wrap things up by talking about some common issues that can pop up and how to prevent them (Chapter 9), as well as what it means to build your personal brand alongside the company and community (Chapter 10).

Throughout the chapters, I'm joined by other Developer Relations professionals who share their stories and give specific, real-world examples so that you can see a practical application of the principles I put forth. It should be noted that there is no “magic bullet” to building a community or being successful with a particular developer audience, and following the example of another company just because it was successful for them isn't

always the best solution.² But by using the principles I've laid out to walk through your company goals and discover where Developer Relations fits within that framework, you can formulate a plan tailored to your specific community.

For some, it will make sense to read straight through, following the process from start to finish and building out your documents, resolutions, and goals as you go. Others should feel free to jump around to solve any problems you're having right now. I often refer back to topics addressed in a previous chapter, so you'll know whether there's relevant material you should reference.

If you're reading this in print format, don't worry about trying to type in all the URLs. I've created an online resource to make your life a little easier—just navigate to marythengvall.com/devrelbook.

I embarked on this writing adventure largely because the essence of the word *community* has impacted me greatly. From launching my career and enabling adventures to facilitating lifelong friendships and making my daily job something that I love, I am deeply committed to communities both personally and professionally: building, fostering, engaging, and loving them. As such, my viewpoint may be a little unique and different from the typical “put the company first” mentality. But I truly believe that if you put the community first, the company will succeed. On the other hand, if we choose to value the product over the community that uses it, we harm both ourselves and the company as a whole, setting us up for potential failure at worst, or at best, keeping the company from reaching its full potential. It's my hope that this book will not only prove the intrinsic value that exists within developer communities, but it will encourage you to place relationship- and community-building at the core of your business.

Let's begin, shall we?

²<https://twitter.com/matthewrevell/status/1003477945707462656>

PART I

What Is the Value of a Technical Community?

CHAPTER 1

An Introduction to Community

If you run a Google search for “community” you’ll find a wide variety of results, from the hit TV sitcom of that name to three different online dictionary definitions, to a handful of developer community websites for particular brands, to a company that sells loudspeakers. With almost five billion search results, it’s no wonder people in the technology industry have such a hard time agreeing on what a community actually is, who it’s comprised of, and how to work with them.

BusinessDictionary.com¹ says community is defined as a “Self-organized network of people with common agenda, cause, or interest, who collaborate by sharing ideas, information, and other resources”. Merriam-Webster says² it’s everything from a unified body of individuals to society as a whole, or even a social state of condition. Dictionary.com³ defines community as “a social, religious, occupational, or other group sharing common characteristics or interests and perceived or perceiving itself as distinct in some respect from the larger society within which it exists”. But even this implies that everyone within the group holds largely the same beliefs and comes from a similar background.

Nothing could be further from the truth when you’re talking about technical communities. The only thing that by definition should bring us together is a common use of or interest in a particular technology, role, tool, or programming language. Yet in my experience, although the starting place may be the use of a particular piece of software, that’s not what makes technical communities special. What starts out as a simple response to a question can lead to a mentorship opportunity, a collaborative project, or the overwhelming success of a product.

¹<http://www.businessdictionary.com/definition/community.html>

²<https://www.merriam-webster.com/dictionary/community>

³<http://www.dictionary.com/browse/community>

Leveraging these definitions and my own professional experience, the definition that I propose for community as it relates to a technical audience is the following:

A group of people who not only share common principles, but also develop and share practices that help individuals in the group thrive

It's important to keep this definition in mind as you work your way through the topics in this book—it will come into play in how you define your departmental goals, what direction you decide to go in with your team, and how you interact with your community.

Establish the (Flexible) Boundaries

Generally speaking, you'll want your community to be inclusive—not limiting yourself to free or paid customers, or even, for that matter, to only those who are actively using your products. You'll want it to be inclusive in other ways as well: accepting of all (respectful) opinions and insights, no matter who they come from or how they challenge the current standards. This isn't always easy! Creating a welcoming, safe environment where everyone feels included is a difficult task that requires a lot of empathy, forethought, and humility to admit when you're wrong. But it's been proven that a more diverse workplace results in better products as well as a healthier work environment,⁴ and this is true in communities as well.

We'll talk later about the importance of getting feedback from a wide variety of customers—paid as well as free accounts, end users as well as buyers, and an assortment of company sizes, as well as gender and geographic demographics. Without a diverse group of customers, you'll wind up with biased feedback, which at best can result in a product that doesn't meet the needs of particular customers and at worst can actively offend the community you're trying to interact with.

That said, though being inclusive is important, you also need to define whom you want to actively engage with as a part of your community, and where the role of Developer Relations (DevRel) comes into play. Are you looking to engage new customers who may not have fully bought into the product yet? Is your priority to work

⁴<https://www.mckinsey.com/business-functions/organization/our-insights/why-diversity-matters>
<https://hbr.org/2018/03/5-things-we-learned-about-creating-a-successful-workplace-diversity-program>

with customers who are already actively engaged, gaining feedback and working with the product and engineering teams to improve your product based on said feedback? Perhaps you want to build up a community of external advocates who will in turn help increase your community support and awareness as you move into new geographic locations. Or maybe you're trying to find folks within the larger developer community who may not be using your product but who have the potential to do so.

Each of these questions ties into a concrete business value that aligns with a company goal: reducing churn, having a better product roadmap, reducing customer acquisition cost, recruiting employees, and more. The real question isn't whether DevRel is capable of contributing value, but which particular value it should focus on at this point in time.

Establishing your overarching goals for community engagement helps you define what your community looks like and what areas you should focus on. Giving yourself an all-inclusive view of community is great, but it can cause problems when your team is suddenly tasked with reaching "anyone and everyone who could possibly use or is already using our product." That's too broad of a task and has too many stakeholders in the company, resulting in an overwhelmed DevRel team with too many tasks on their plates, struggling to figure out what their priorities are. Suddenly, no matter what department DevRel falls under, they've got marketing, product, engineering, customer support, and sales clamoring for their attention. By understanding the specific value this team brings to the company and which segment of the community they should focus on, you can prioritize and delegate these overwhelming and sometimes conflicting requests.

So, first things first: it's essential that you understand your company goals. Once those are clear, think about whether or how fostering an active community can help your company achieve those goals. Here are some questions you'll want to ask:

- Why do you want a community?
- What do you hope to accomplish with a community?
- Who makes up your community?
- Which segment of that community do you want to focus on first?

And perhaps most relevant to our conversations throughout this book:

- Do you actually need a Developer Relations team to accomplish those goals? (Hint: the answer isn't always yes!)